User Guide

AWS Systems Manager

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Systems Manager: User Guide

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

..... xii
Was ist AWS Systems Manager? 1
Welche Vorteile bietet Systems Manager meinen Vorgängen? 2
Wer sollte Systems Manager verwenden? 2
Was sind die Hauptfeatures von Systems Manager? 3
Unterstützt AWS-Regionen 4
Zugriff auf Systems Manager 4
Systems Manager Service-Namengeschichte 6
Unterstützte Betriebssysteme und Maschinentypen 6
Unterstützte Betriebssysteme für Systems Manager 6
Unterstützte Maschinentypen in Hybrid- und Multi-Cloud-Umgebungen 11
Was ist die vereinheitlichte Konsole? 12
AWS Systems ManagerChange ManagerÄnderung der Verfügbarkeit 17
Bewertung alternativer Dienste 17
Einrichten von verwalteten Knoten für AWS Systems Manager 18
Verwalten von EC2-Instances mit Systems Manager 19
Konfigurieren von erforderlichen Instance-Berechtigungen für Systems Manager 19
Verbessern Sie die Sicherheit von EC2 Instanzen mithilfe von VPC-Endpunkten für Systems Manager 31
Verwalten von Knoten in Hybrid- und Multi-Cloud-Umgebungen mit Systems Manager 37
Erstellen Sie die für Systems Manager in Hybrid- und Multi-Cloud-Umgebungen erforderliche IAM-Servicerolle 39
Eine Hybridaktivierung erstellen, um Knoten bei Systems Manager zu registrieren 49
Installation von SSM Agent auf hybriden Linux-Knoten 55
Installation von SSM Agent auf hybriden Windows Server-Knoten 62
Verwalten von Edge-Geräten mit Systems Manager 67
Erstellen einer IAM-Servicerolle für Edge-Geräte 68
Konfigurieren Sie Ihre Edge-Geräte für AWS IoT Greengrass 75
Aktualisieren Sie die AWS IoT Greengrass Token-Exchange-Rolle und installieren Sie sie SSM Agent auf Ihren Edge-Geräten 75
Einen AWS Organizations delegierten Administrator für Systems Manager erstellen 76
Verwenden eines delegierten Administrators mit Change Manager 77
Verwenden eines delegierten Administrators mit Explorer 77
Verwenden eines delegierten Administrators mit OpsCenter 78

Verwenden eines delegierten Administrators mit Quick Setup ... 78
Allgemeine Einrichtung ... 78
Melden Sie sich an für ein AWS-Konto ... 79
Erstellen eines Benutzers mit Administratorzugriff ... 79
Einrichten von AWS Systems Manager ... 81
Einrichten des Systems-Manager-Konsolenzugriffs ... 81
Systems-Manager-Onboarding-Richtlinie ... 81
AWS Systems Manager Richtlinie für Konsolenbetreiber ... 87
AWS Systems Manager Richtlinie „Nur Lesen“ für Konsolenbetreiber ... 91
Einrichten der einheitlichen Systems-Manager-Konsole für eine Organisation ... 94
Voraussetzungen ... 94
Einheitliche Konsolenressourcen ... 95
Einrichten der vereinheitlichten Konsole ... 97
Einrichtung einer einheitlichen Systems-Manager-Konsole für ein einziges Konto und eine Region ... 100
Einheitliche Konsolenressourcen ... 100
Einrichten der vereinheitlichten Konsole ... 102
Deaktivieren der einheitlichen Systems-Manager-Konsole ... 104
Ausführen von Knotenverwaltungsaufgaben ... 106
Überprüfen von Erkenntnissen ... 106
Hinzufügen oder Entfernen von Widgets ... 109
Neuanordnen von Widgets ... 110
Erkunden von Knoten ... 111
Erkunden von Knoten mithilfe von Konsolenfiltern ... 112
Erkunden von Knoten mithilfe von Text-Prompts in Amazon Q ... 117
Details zu einzelnen Knoten anzeigen und Maßnahmen für einen Knoten ergreifen ... 123
Bericht über verwaltete Knoten herunterladen oder exportieren ... 125
Inhalt und Erscheinungsbild von Knotenberichten verwalten ... 129
Just-in-Time-Knotenzugriff mit Systems Manager ... 130
just-in-timeZugriff mit Systems Manager einrichten ... 133
Starten Sie eine just-in-time Knotenzugriffssitzung ... 183
Verwaltung von just-in-time Zugriffsanfragen ... 185
Übergang zum just-in-time Knotenzugriff von Session Manager ... 186
just-in-timeZugriff mit Systems Manager deaktivieren ... 191
Häufig gestellte Fragen ... 192
Diagnose und Abhilfemaßnahmen ... 194

Diagnose und Behebung fehlgeschlagener Bereitstellungen ... 195
Diagnose und Behebung von Konfigurationsabweichungen ... 196
Diagnose und Behebung nicht verwalteter Amazon-EC2-Instances ... 197
Arten von Runbook-Aktionen mit Auswirkungen auf die Behebung ... 205
Details zum Ausführungsverlauf von Behebungen anzeigen ... 208
Systems-Manager-Einstellungen anpassen ... 209
Kontoeinrichtungseinstellungen ... 209
Organisationseinstellungen ... 209
Funktionskonfigurationen ... 210
Einstellungen diagnostizieren und korrigieren ... 214
Arbeiten mit Amazon-S3-Buckets und Bucket-Richtlinien für Systems Manager ... 216
AWS Systems Manager Tools verwenden ... 226
Knoten-Tools ... 226
Tools für das Änderungsmanagement ... 229
Anwendungstools ... 230
Tools für den Betrieb ... 231
Knotenaufgabe mit Systems Manager ausführen ... 233
Voraussetzungen ... 233
Starten einer Instance mit einer AMI mit vorinstalliertem SSM Agent ... 233
Eine Verbindung zu Ihrer verwalteten Instance mithilfe von Systems Manager herstellen ... 235
Bereinigen Ihrer Instance ... 235
Knoten-Tools ... 235
Compliance ... 236
Distributor ... 256
Fleet Manager ... 314
Hybride Aktivierungen ... 407
Bestand ... 408
Patch Manager ... 510
Run Command ... 846
Session Manager ... 917
State Manager ... 1077
Veränderungsbeschleunigungs-Toolkit ... 1183
Automatisierung ... 1183
Change Calendar ... 1690
Change Manager ... 1711
-Documents ... 1810

Maintenance Windows 1959
Quick Setup 2097
Anwendungstools 230
AWS AppConfig 2151
Application Manager 2151
Parameter Store 2201
Tools für den Betrieb 2366
Incident Manager 2367
Explorer 2368
OpsCenter 2411
CloudWatch-Dashboards 2501
Arbeiten mit SSM Agent 2503
Technische Details über die SSM Agent erfahren 2504
Verhalten der Anmeldeinformationen von SSM Agent-Version 3.2.x.x 2504
Priorität der SSM Agent-Anmeldeinformationen 2504
Konfigurieren von SSM Agent für die Verwendung mit dem Federal Information Processing Standard (FIPS) 2507
Über das lokale ssm-user-Konto 2509
SSM Agent und die Instance Metadata Service (IMDS) 2509
Behalten SSM Agent up-to-date 2510
Sicherstellen, dass das SSM Agent-Installationsverzeichnis nicht geändert, verschoben oder gelöscht wird 2510
SSM Agentfortlaufende Updates von AWS-Regionen 2511
SSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets 2511
SSM Agent auf GitHub 2520
Grundlegendes zum SSM Agent-Ruhezustand 2521
Finden Sie AMIs mit dem vorinstallierten SSM Agent 2523
Überprüfen des Status des SSM Agent 2524
Arbeiten mit SSM Agent auf EC2-Instances für Linux 2529
Verifizieren der Signatur von SSM Agent 2529
Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux 2542
Konfigurieren Sie SSM Agent, um einen Proxy in Linux-Knoten zu verwenden 2586
Arbeiten mit SSM Agent EC2 On-Instances für macOS 2590
Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für macOS 2592
Arbeiten mit SSM Agent EC2 On-Instances für Windows Server 2594

Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Windows Server ... 2595
Konfigurieren des SSM Agent zur Nutzung eines Proxys für Windows Server-Instances ... 2598
Prüfen des SSM Agent-Status und Starten des Agenten ... 2602
Überprüfen der SSM Agent-Versionsnummer ... 2604
Anzeigen von SSM Agent-Protokollen ... 2608
Zulassen von SSM Agent-Debug-Protokollierung ... 2609
Einschränken des Zugriffs auf Befehle auf Stammebene durch SSM Agent ... 2612
Automatisieren von Updates für SSM Agent ... 2613
Automatische Aktualisierung von SSM Agent ... 2615
Abonnieren von SSM Agent-Benachrichtigungen ... 2616
Fehlerbehebung für SSM Agent ... 2617
SSM Agent ist veraltet ... 2618
Probleme mithilfe von SSM Agent-Protokolldateien beheben ... 2618
Agent-Protokolldateien werden nicht gedreht (Windows) ... 2619
Keine Verbindung mit SSM-Endpunkten möglich ... 2620
Ihre VPC-Konfiguration überprüfen ... 2620
Ihre VPC-DNS-bezogenen Attribute überprüfen ... 2622
Die Eingangsregeln für Endpunkt-Sicherheitsgruppen überprüfen ... 2623
Verwenden Sie `ssm-cli`, um die Verfügbarkeit von verwalteten Knoten zu überprüfen ... 2623
Sicherheit ... 2624
Datenschutz ... 2625
Datenverschlüsselung ... 2626
Richtlinie für den Datenverkehr zwischen Netzwerken ... 2629
Datenperimeter ... 2629
Identity and Access Management ... 2631
Zielgruppe ... 2631
Authentifizierung mit Identitäten ... 2632
Verwalten des Zugriffs mit Richtlinien ... 2633
Funktionsweise von AWS Systems Manager mit IAM ... 2637
Beispiele für identitätsbasierte Richtlinien ... 2647
AWS verwaltete Richtlinien ... 2660
Fehlersuche ... 2729
Verwenden von servicegebundenen Rollen ... 2731
Inventar und Explorer-Datenrollen ... 2733
OpsCenter- und Explorer-Rolle der Kontoerkennung ... 2736

OpsData und OpsItems Erstellungsrolle ... 2739
Rolle für die Erstellung operativer Einblicke ... 2744
Rolle der Zustandsprüfung bei der Quick Setup-Bereitstellung ... 2747
OpsData-Servicerolle exportieren ... 2750
Just-in-time Dienstrolle für den Knotenzugriff ... 2753
Just-in-time Dienstrolle für Benachrichtigungen über Knotenzugriffsanfragen ... 2757
Protokollierung und Überwachung ... 2761
Compliance-Validierung ... 2764
Ausfallsicherheit ... 2765
Sicherheit der Infrastruktur ... 2765
Konfigurations- und Schwachstellenanalyse ... 2766
Bewährte Methoden für die Gewährleistung der Sicherheit ... 2766
Bewährte Methoden für vorbeugende Systems Manager-Sicherheitsmaßnahmen ... 2766
Bewährte Methoden für die Installation von SSM Agent ... 2771
Bewährte Methoden zur Überwachung und Prüfung von Systems Manager ... 2771
Codebeispiele ... 2774
Grundlagen ... 2781
Hallo Systems Manager ... 2785
Kennenlernen der Grundlagen ... 2789
Aktionen ... 2841
Protokollierung und Überwachung ... 3183
Tools zur Überwachung ... 3184
Senden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent) ... 3184
Migrieren Sie die Erfassung von Windows Server-Knotenprotokollen auf den CloudWatch Agenten ... 3186
Speichern Sie die CloudWatch Agentenkonfigurationseinstellungen in Parameter Store ... 3197
Rollback zur Protokollerfassung mit SSM Agent ... 3198
SSM AgentProtokolle an CloudWatch Logs senden ... 3202
Überwachung der Ereignisse Ihrer Änderungsanfragen ... 3205
Überwachung Ihrer Automatisierungen ... 3208
Automation-Metriken ... 3209
Überwachen von Run Command-Metriken mit Amazon CloudWatch ... 3209
Systems Manager Run Command-Metriken und -Dimensionen ... 3210
Protokollierung von AWS Systems Manager-API-Aufrufen mit AWS CloudTrail ... 3211
Systems-Manager-Datenereignisse in CloudTrail ... 3213
Systems-Manager-Verwaltungsereignisse in CloudTrail ... 3215

Systems Manager Beispiele für Ereignisse ... 3215
Protokollierung der Automation-Aktionsausgabe mit CloudWatch Logs ... 3221
Konfigurieren von Amazon CloudWatch Logs für Run Command ... 3225
Angeben von CloudWatch Logs beim Senden von Befehlen ... 3226
Anzeigen von Befehlsausgaben in CloudWatch Logs ... 3227
Überwachung mit Amazon EventBridge ... 3228
Konfigurieren von EventBridge für Systems-Manager-Ereignisse ... 3230
EventBridge Amazon-Veranstaltungsbeispiele für Systems Manager ... 3233
Beispielszenarien: Systems Manager Manager-Ziele in EventBridge Amazon-Regeln ... 3250
Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen ... 3251
Konfigurieren von Amazon SNS-Benachrichtigungen für AWS Systems Manager ... 3252
Beispiele für Amazon SNS-Benachrichtigungen für AWS Systems Manager ... 3264
Verwenden von Run Command zum Senden eines Befehls, der Statusbenachrichtigungen zurückgibt ... 3266
Verwenden eines Wartungsfensters zum Senden eines Befehls, der Statusbenachrichtigungen zurückgibt ... 3270
Produkt- und Service-Integrationen ... 3276
Integration mit AWS-Services ... 3276
Datenverarbeitung ... 3276
Internet of Things (IoT) ... 3279
Speicher ... 3280
Entwicklertools ... 3281
Sicherheit, Identität und Compliance ... 3282
Kryptografie und PKI ... 3285
Verwaltung und Governance ... 3286
Netzwerk und Bereitstellung von Inhalten ... 3291
Analysen ... 3292
Anwendungsintegration ... 3294
AWS-Managementkonsole ... 3295
Ausführen von Skripts von Amazon S3 ... 3296
Referenzieren von AWS Secrets Manager-Geheimnissen über Parameter Store-Parameter ... 3301
AWS KMS Verschlüsselung für Parameter Store SecureString Parameter ... 3307
AWS Secrets Manager-Secrets in Amazon Elastic Kubernetes Service verwenden ... 3322
Verwenden von Parameter Store-Parametern in AWS Lambda -Funktionen ... 3342

Integration in andere Produkte und Services ... 3366
Ausführen von Skripts von GitHub ... 3370
Verwenden von Chef InSpec-Profilen mit Systems Manager Compliance ... 3379
Integration mit ServiceNow ... 3385
AWS Systems Manager Referenz ... 3386
Arbeitet mit AWS SDKs ... 3387
Amazon-S3-Buckets für Patch-Operationen ... 3388
Buckets, die SSM-Befehlsdokumente für Patch-Operationen enthalten (Linux und Windows Server) ... 3389
Buckets, die SSM-Befehlsdokumente für Patch-Operationen enthalten (macOS) ... 3392
Buckets mit verwalteten Patch-Baseline-Snapshots AWS ... 3395
EventBridge Ereignismuster und -typen für Systems Manager ... 3398
Ereignistyp: Automation ... 3399
Ereignistyp: Change Calendar ... 3400
Ereignistyp: Change Manager ... 3400
Ereignistyp: Configuration Compliance ... 3401
Ereignistyp: Inventory ... 3401
Ereignistyp: Wartungsfenster ... 3402
Ereignistyp: OpsCenter ... 3405
Ereignistyp: Parameter Store ... 3405
Ereignistyp: Run Command ... 3406
Ereignistyp: State Manager ... 3407
Cron- und Rate-Ausdrücke ... 3408
Allgemeine Informationen zu Cron- und Rate-Ausdrücken ... 3408
Cron- und Rate-Ausdrücke für Zuordnungen ... 3414
Cron- und Rate-Ausdrücke für Wartungsfenster ... 3417
ec2messages, ssmmessages und andere API-Operationen ... 3419
API-Vorgänge (ssmmessages- und ec2messages-Endpunkte) im Zusammenhang mit Agenten ... 3420
ssm:*-Namespace-Instance-bezogene API-Vorgänge ... 3422
ssm:* Namespace andere API-Vorgänge ... 3424
Formate von Datums- und Uhrzeitzeichenfolgen für Systems Manager ... 3424
Formatieren von Datums- und Uhrzeitzeichenfolgen für Systems Manager ... 3425
Erstellen benutzerdefinierter Datums- und Uhrzeitzeichenfolgen für Systems Manager ... 3425
Anwendungsfälle und bewährte Methoden ... 3428
Löschen von Systems Manager Ressourcen und Artefakten ... 3431

Auswahl zwischen State Manager und Maintenance Windows 3436
State Manager und Maintenance Windows: Hauptanwendungsfälle 3436
Ähnliche Informationen 3445
Dokumentverlauf 3447
Updates vor Juni 2018 3702
Dokumentkonventionen 3723

AWS Systems ManagerChange Managersteht neuen Kunden nicht mehr offen. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.

Was ist AWS Systems Manager?

AWS Systems Manager unterstützt Sie bei der zentralen Anzeige, Verwaltung und dem Betrieb skalierbarer Knoten in lokalen und AWS Multicloud-Umgebungen. Mit der Einführung der einheitlichen Konsolenoberfläche wurden unter dem Dach von Systems Manager mehrere Tools zusammengefasst, mit denen Sie allgemeine Knotenaufgaben in allen AWS-Konten und AWS-Regionen erledigen können.

Um Systems Manager verwenden zu können, müssen Knoten verwaltet werden, was bedeutet, dass SSM Agent auf der Maschine installiert sind und der Agent mit dem Systems-Manager-Service kommunizieren kann. Um herauszufinden, warum Knoten nicht als verwaltet gemeldet werden, bietet Systems Manager ein Runbook zur Diagnose und Behebung von Agentenproblemen mit nur einem Klick, das Sie so konfigurieren können, dass es automatisch gemäß einem von Ihnen definierten Zeitplan ausgeführt wird. Mit diesem Feature können Sie ermitteln, warum Knoten keine Verbindung zu Systems Manager herstellen können, einschließlich Netzwerkfehlkonfigurationen. Dieses Feature bietet auch empfohlene Runbooks zur Behebung von Netzwerkproblemen und anderen Problemen, die verhindern, dass Knoten als verwaltete Knoten konfiguriert werden.

Das einheitliche Konsolenerlebnis umfasst auch ein Dashboard, das einen anspruchsvollen Überblick über Ihre Knoten bietet. Sie können detailliertere Informationen zu Knoten abrufen, z. B. auf welchen Knoten veraltete Betriebssystemsoftware (OS) läuft. Sie können auch Filter für detaillierte Ansichten verwenden, die auf Instanz-Metadaten wie Betriebssystemversionen, AWS-Regionen AWS-Konten, OSs und Versionen basieren. SSM Agent Diese Filter helfen Ihnen dabei, relevante Informationen auf einer bestimmten Konto- oder Anwendungsebene in Ihrer gesamten Organisation abzurufen.

Themen

- Welche Vorteile bietet Systems Manager meinen Vorgängen?
- Wer sollte Systems Manager verwenden?
- Was sind die Hauptfeatures von Systems Manager?
- Unterstützt AWS-Regionen
- Zugriff auf Systems Manager
- Systems Manager Service-Namengeschichte
- Unterstützte Betriebssysteme und Maschinentypen
- Was ist die vereinheitlichte Konsole?

Welche Vorteile bietet Systems Manager meinen Vorgängen?

Systems Manager bietet folgende Vorteile:

- Transparenz Ihrer gesamten Infrastruktur verbessern

  Systems Manager bietet eine zentrale Ansicht der Knoten in den Konten und Regionen Ihres Unternehmens. Greifen Sie schnell auf Instance-Informationen wie ID, Name, Betriebssystemdetails und installierte Agenten zu. Verwenden Sie Amazon Q Developer, um Instance-Metadaten in natürlicher Sprache abzufragen, sodass Sie Probleme identifizieren und schneller Maßnahmen ergreifen können.

- Die betriebliche Effizienz durch Automatisierung steigern

  Automatisieren Sie allgemeine Betriebsaufgaben und reduzieren Sie den Zeit- und Arbeitsaufwand für die Wartung Ihrer Systeme. Systems Manager bietet eine sichere und zuverlässige Fernverwaltung Ihrer Knoten in großem Umfang, ohne sich bei Ihren Servern anmelden zu müssen. Sie müssen Bastion Hosts, SSH oder Remote nicht mehr verwenden. PowerShell Systems Manager bietet außerdem eine einfache Möglichkeit, gängige Verwaltungsaufgaben über Knotengruppen hinweg zu automatisieren, z. B. Registrierungsänderungen, Benutzerverwaltung sowie Software- und Patch-Installationen.

- Vereinfachen Sie die skalierbare Knotenverwaltung in jeder Umgebung

  Systems Manager unterstützt Sie bei der Verwaltung von Knoten in AWS lokalen und Multicloud-Umgebungen. Planen Sie automatisierte Diagnosen, um SSM Agent-Probleme zu identifizieren und diese mit Runbooks mit einem Klick zu beheben. Nachdem Ihre Knoten als verwaltete Knoten konfiguriert wurden, können Sie wichtige betriebliche Aufgaben wie das Anwenden von Sicherheitspatches, das Initiieren protokollierter Sitzungen und das Ausführen von Befehlen aus der Ferne ausführen.

Wer sollte Systems Manager verwenden?

Systems Manager wird von IT-Betriebsmanagern und -betreibern, DevOps Ingenieuren, Sicherheits- und Compliance-Managern sowie IT-Direktoren verwendet und CIOs. Im Großen und Ganzen ist Systems Manager für Folgendes geeignet:

- Organisationen, die das Management und die Sicherheit ihrer Knoten in großem Umfang verbessern möchten.

- Organisationen, die die Transparenz und betriebliche Flexibilität bei der Verwaltung ihrer Infrastruktur erhöhen möchten.
- Organisationen, die ihre betriebliche Effizienz in großem Umfang steigern möchten.

Was sind die Hauptfeatures von Systems Manager?

Die Hauptfeatures von Systems Manager werden von der vereinheitlichten Konsole und den einzelnen Tools gemeinsam genutzt, mit denen Systems Manager Sie bei der massenweisen Verwaltung von Knoten unterstützt.

Vereinheitlichte Konsole

Die vereinheitlichte Konsole bietet eine zentrale Oberfläche zum Anzeigen und Verwalten Ihrer Knoten. Die Konsole vereint mehrere Systems-Manager-Tools und andere, um Ihnen Folgendes zu bieten:

- Zentralisierte Ansichten Ihrer Knoten
- Detailinformationen zu Knoten
- Automatisierte Diagnose und Behebung häufiger Knotenprobleme

Weitere Informationen zur einheitlichen Konsole finden Sie unter Was ist die vereinheitlichte Konsole?.

Tools

Die Tools bestehen aus den einzelnen Funktionen von Systems Manager und deren Features wie Run Command, Session Manager, Automation und Parameter Store. Die Systems-Manager-Tools erlauben die folgenden Aktionen:

- Just-in-time Zugriff auf den Zugangsknoten
- Skaliertes Patchen von Knoten
- Sichere Verbindung zu Knoten, ohne eingehende Ports zu öffnen
- Befehle remote auf Knoten ausführen
- Daten speichern, auf die von Anwendungen verwiesen wird
- Allgemeine Systemverwaltungsaufgaben automatisieren

Weitere Informationen zu den Systems-Manager-Tools finden Sie unter AWS Systems Manager Tools verwenden.

Unterstützt AWS-Regionen

Eine Liste der Tools AWS-Regionen , die Systems Manager unterstützen, finden Sie unter Systems Manager Service Endpoints in der Allgemeine Amazon Web Services-Referenz.

Die vereinheitlichte Systems Manager Manager-Konsole, veröffentlicht am 21. November 2024, ist in den folgenden Versionen verfügbar AWS-Regionen:

- Region USA Ost (Nord-Virginia)
- Region USA Ost (Ohio)
- Region USA West (Nordkalifornien)
- Region USA West (Oregon)
- Region Kanada (Zentral)
- Region Südamerika (São Paulo)
- Region Asien-Pazifik (Mumbai)
- Region Asien-Pazifik (Tokio)
- Asia Pacific (Seoul) Region
- Region Asien-Pazifik (Singapur)
- Region Asien-Pazifik (Sydney)
- Region Europa (Frankfurt)
- Region Europa (Stockholm)
- Region Europa (Irland)
- Region Europa (London)
- Region Europa (Paris)

Zugriff auf Systems Manager

Sie können folgendermaßen mit Systems Manager arbeiten:

Systems Manager-Konsole

Die Systems-Manager-Konsole ist eine browserbasierte Schnittstelle für den Zugriff auf und die Verwendung von Systems Manager.

AWS IoT Greengrass V2 Konsole

Sie können Edge-Geräte, für die sie konfiguriert sind, AWS IoT Greengrass in der Greengrass-Konsole anzeigen und verwalten.

AWS Befehlszeilentools

Mithilfe der AWS Befehlszeilentools können Sie Befehle an der Befehlszeile Ihres Systems ausgeben, um Systems Manager und andere AWS Aufgaben auszuführen. Die Tools werden unter Linux, macOS und Windows unterstützt. Die AWS Command Line Interface (AWS CLI) kann schneller und bequemer zu verwenden sein als die Konsole. Die Befehlszeilen-Tools sind auch hilfreich, wenn Sie Skripts erstellen möchten, die AWS -Aufgaben ausführen.

AWS stellt zwei Gruppen von Befehlszeilentools bereit: das AWS Command Line Interfaceund das AWS Tools for Windows PowerShell. Informationen zur Installation und Verwendung von finden Sie im AWS Command Line Interface Benutzerhandbuch. AWS CLI Informationen zur Installation und Verwendung der Tools für Windows PowerShell finden Sie im AWS -Tools für PowerShell Benutzerhandbuch.

> **Note**
>
> Auf Ihren Windows Server-Instances wird Windows PowerShell 3.0 oder höher benötigt, um bestimmte SSM-Dokumente ausführen zu können (z. B. das Legacy–`AWS-ApplyPatchBaseline`Dokument). Überprüfen Sie, ob Ihre Windows Server-Instances auf Windows Management Framework 3.0 oder höher ausgeführt werden. Das Framework umfasst Windows PowerShell.

AWS SDKs

AWS stellt Softwareentwicklungskits (SDKs) bereit, die aus Bibliotheken und Beispielcode für verschiedene Programmiersprachen und Plattformen bestehen (z. B. Java, Python, Ruby, .NET, iOS und Android und andere). SDKsSie bieten eine bequeme Möglichkeit, programmatischen Zugriff auf Systems Manager zu gewähren. Informationen zu den AWS SDKs, einschließlich deren Download und Installation, finden Sie unter Tools für Amazon Web Services.

Systems Manager Service-Namengeschichte

AWS Systems Manager (Systems Manager) war früher bekannt als Amazon Simple Systems Manager (SSM) "und"Amazon EC2 Systems Manager (SSM)“. Der ursprüngliche abgekürzte Name des Dienstes, "SSM„, findet sich immer noch in verschiedenen AWS Ressourcen wieder, darunter auch in einigen anderen Servicekonsolen. Hier einige Beispiele:

- Systems Manager Agent: SSM Agent
- Systems Manager Parameter: SSM-Parameter
- Systems Manager-Service-Endpunkte: `ssm.region.amazonaws.com`
- AWS CloudFormation Ressourcentypen: `AWS::SSM::Document`
- AWS Config Regel-ID: `EC2_INSTANCE_MANAGED_BY_SSM`
- AWS Command Line Interface (AWS CLI) Befehle: `aws ssm describe-patch-baselines`
- AWS Identity and Access Management Namen der (IAM) verwalteten Richtlinien: `AmazonSSMReadOnlyAccess`
- Ressource für Systems Manager ARNs: `arn:aws:ssm:region:account-id:patchbaseline/pb-07d8884178EXAMPLE`

Unterstützte Betriebssysteme und Maschinentypen

Bevor Sie mit Systems Manager arbeiten, stellen Sie sicher, dass Ihr Betriebssystem (OS), Ihre Betriebssystemversion und Ihr Maschinentyp als verwaltete Knoten unterstützt werden.

Themen

- Unterstützte Betriebssysteme für Systems Manager
- Unterstützte Maschinentypen in Hybrid- und Multi-Cloud-Umgebungen

Unterstützte Betriebssysteme für Systems Manager

In den folgenden Abschnitten sind die von Systems Manager unterstützten Betriebssystemversionen OSs und Betriebssystemversionen aufgeführt.

> **Note**
>
> Wenn Sie planen, AWS IoT Greengrass Kerngeräte mithilfe von Systems Manager zu verwalten und zu konfigurieren, müssen diese Geräte die Anforderungen für erfüllen AWS IoT Greengrass. Weitere Informationen finden Sie im AWS IoT Greengrass Version 2 Entwicklerhandbuch unter Einrichten von AWS IoT Greengrass Kerngeräten.
> Wenn Sie Geräte verwalten AWS IoT und konfigurieren möchten, die keine AWS Edge-Geräte sind, müssen diese Geräte die hier aufgeführten Anforderungen erfüllen und als lokal verwaltete Knoten für Systems Manager konfiguriert sein. Weitere Informationen finden Sie unter Verwalten von Edge-Geräten mit Systems Manager.

> **Important**
>
> Wir empfehlen dringend, die Verwendung von Betriebssystemversionen zu vermeiden, die End-of-Life (EOL) erreicht haben. Betriebssystemanbieter, darunter auch, bieten AWS in der Regel keine Sicherheitspatches oder andere Updates für Versionen an, die EOL erreicht haben. Die weitere Verwendung eines EOL-Systems erhöht das Risiko, dass Upgrades, einschließlich Sicherheitsupdates, und andere Betriebsprobleme nicht durchgeführt werden können, erheblich. AWS testet die Systems Manager Manager-Funktionalität nicht auf Betriebssystemversionen, die EOL erreicht haben.
> Patch Manager, ein Tool in Systems Manager, unterstützt möglicherweise nicht alle in diesem Thema aufgeführten OS-Versionen. Eine Liste der von Patch Manager unterstützten Betriebssystemversionen finden Sie unter Patch Manager-Voraussetzungen.

Betriebssystemtypen

- Linux
- macOS(Nur EC2 Amazon-Instances)
- Windows Server

Wählen Sie eine Betriebssystemplattform aus, um die unterstützten Haupt- und Nebenversionen zu sehen.

Linux

AlmaLinux

Versionen	x86	x86_64	ARM64
8.3–8.10		✓	✓
9.x		✓	✓

Amazon Linux 2

Versionen	x86	x86_64	ARM64
2.0 und alle späteren Versionen		✓	✓

Amazon Linux 2023

Versionen	x86	x86_64	ARM64
2023.0.20230315.0 und alle späteren Versionen		✓	✓

Bottlerocket

Versionen	x86_64	ARM64
1.0.0 und alle späteren Versionen	✓	✓

CentOS Stream

Versionen	x86	x86_64	ARM64
9		✓	✓

Debian Server

Versionen	x86	x86_64	ARM64
Bullseye (11)		✓	✓
Bookworm(12)		✓	✓

Oracle Linux

Versionen	x86	x86_64	ARM64
7.5–7.8		✓	
8.x		✓	
9.x		✓	

Red Hat Enterprise Linux (RHEL)

Versionen	x86	x86_64	ARM64
7.0–7.5		✓	
7.6–8.x		✓	✓
9.x		✓	✓
10. x		✓	✓

Rocky Linux

Versionen	x86	x86_64	ARM64
8.x		✓	✓
9.x		✓	✓

SUSE Linux Enterprise Server (SLES)

Versionen	x86	x86_64	ARM64
15.3 und höher		✓	✓

Ubuntu Server

Versionen	x86	x86_64	ARM64
16.04 LTS und 18.04 LTS		✓	✓
20,04 L		✓	✓
22.04 LTS		✓	✓
24.04 LTS		✓	✓
25,04		✓	✓

macOS(Nur EC2 Amazon-Instances)

Version	x86	x86_64	Mac with Apple silicon
13.x (Ventura)		✓	✓
14.x (Sonoma)		✓	✓
15.x (Sequoia)		✓	✓

> **Note**
>
> macOSwird nicht in allen unterstützt AWS-Regionen. Weitere Informationen zum EC2 Amazon-Support für macOS finden Sie unter Amazon EC2 Mac-Instances im EC2 Amazon-Benutzerhandbuch.

Windows Server

SSM Agenterfordert Windows PowerShell 3.0 oder später die Ausführung bestimmter AWS Systems Manager Dokumente (SSM-Dokumente) auf Windows Server Instanzen (z. B. das ältere `AWS-ApplyPatchBaseline` Dokument). Überprüfen Sie, ob Ihre Windows Server-Instances auf Windows Management Framework 3.0 oder höher ausgeführt werden. Dieses Framework umfasst Windows PowerShell. Weitere Informationen finden Sie unter Windows Management Framework 3.0.

Version	x86	x86_64	ARM64
2012 und 2012 R2[2]		✓	
2016		✓	
2019		✓	
2022		✓	
2025		✓	

[1] Unterstützung für Windows Server 2012 und 2012 R2: Windows Server 2012 und 2012 R2 haben am 10. Oktober 2023 das Ende der Unterstützung erreicht. Für die Verwendung SSM Agent mit diesen Versionen empfehlen wir die Verwendung von Extended Security Updates (ESUs) von Microsoft. Weitere Informationen finden Sie auf der Microsoft-Website unter Windows Server 2012 und 2012 R2 erreichen das Ende des Supports.

Unterstützte Maschinentypen in Hybrid- und Multi-Cloud-Umgebungen

Systems Manager unterstützt eine Reihe von Maschinentypen als verwaltete Knoten. Ein verwalteter Knoten ist eine Maschine, die für die Arbeit mit Systems Manager konfiguriert ist.

In diesem Benutzerhandbuch werden die Begriffe Hybrid- und Multi-Cloud verwendet, um sich auf eine Umgebung zu beziehen, die eine beliebige Kombination der folgenden Maschinentypen enthält:

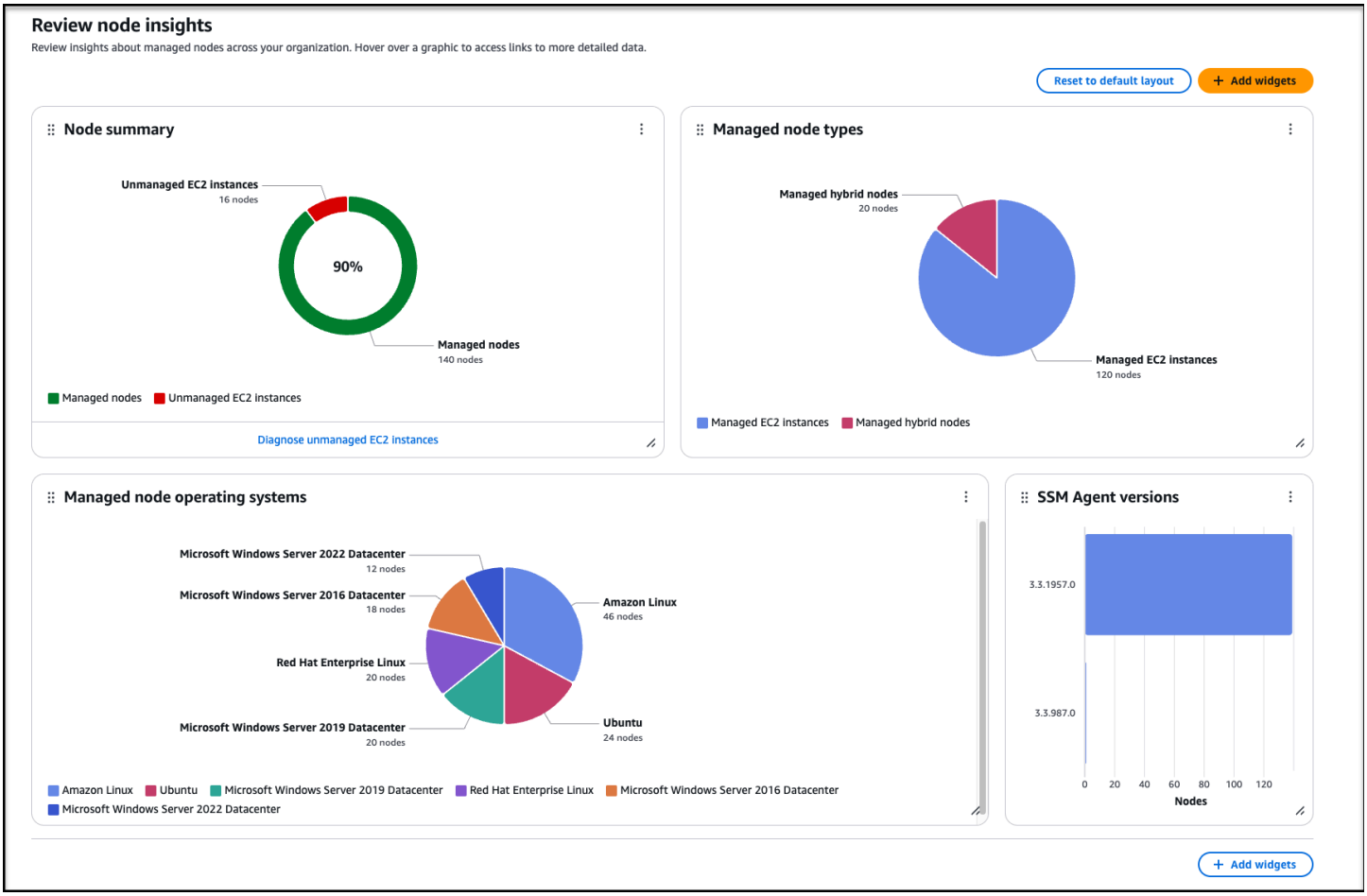
- Amazon Elastic Compute Cloud (Amazon EC2) -Instanzen
- Server in Ihren eigenen Räumlichkeiten (On-Premises-Server)
- AWS IoT Greengrass Kerngeräte
- AWS IoT und Geräte, die nicht zu den AWS Edge-Geräten gehören

- Virtuelle Maschinen (VMs), auch VMs in anderen Cloud-Umgebungen

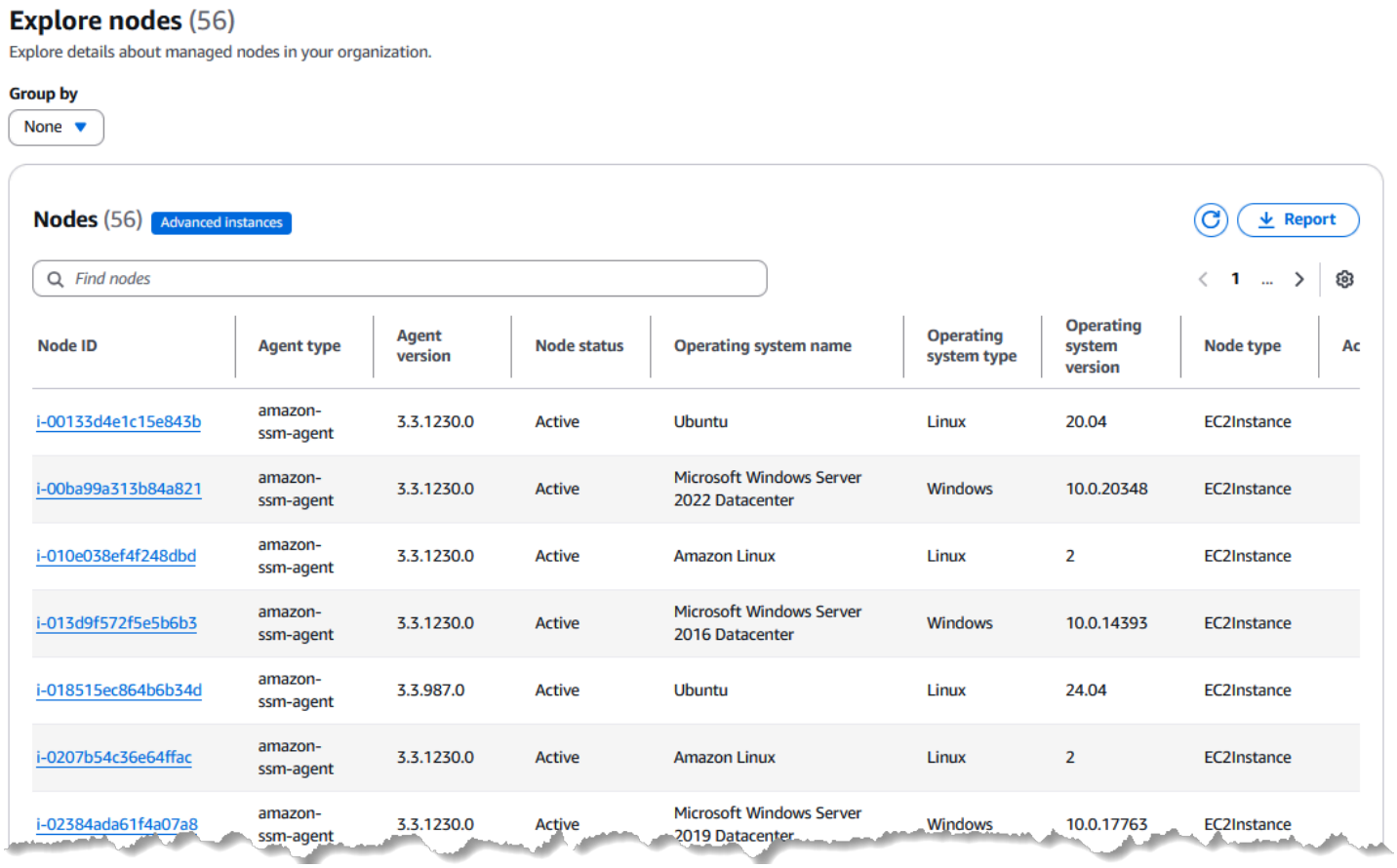
Informationen zur AWS Unterstützung von Hybrid- und Multicloud-Umgebungen finden Sie unter AWS Lösungen für Hybrid- und Multicloud-Umgebungen.

Was ist die vereinheitlichte Konsole?

Die einheitliche Systems Manager Manager-Konsole ist ein konsolidiertes Erlebnis, das verschiedene Tools kombiniert, um Ihnen zu helfen, gemeinsame Knotenaufgaben für mehrere Knoten AWS-Konten und AWS-Regionen innerhalb einer AWS Organizations Organisation oder für ein einzelnes Konto und eine Region zu erledigen. Knoten können EC2 Instanzen, Hybridserver oder Server sein, die in einer Multi-Cloud-Umgebung ausgeführt werden. Die vereinheitlichte Konsole liefert detaillierte Informationen zu Ihren Knoten. Sie können Berichte für Ihre Knoten erstellen und häufig auftretende Probleme diagnostizieren und beheben, die verhindern, dass Knoten als von Systems Manager verwaltet gemeldet werden, z. B. Verbindungsprobleme.



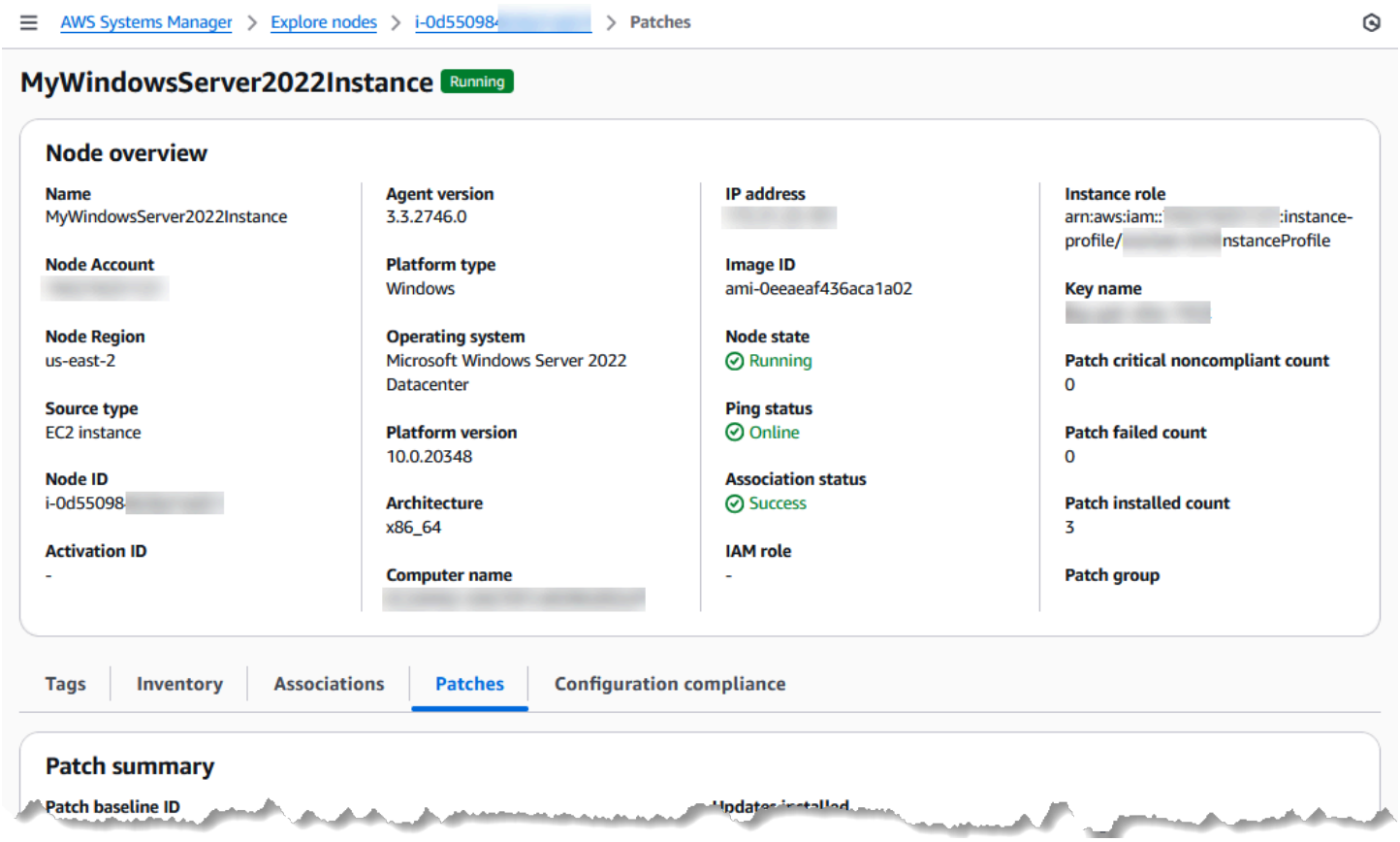
Zusätzlich zu den Zusammenfassungen über Knoten auf der Seite Knoten-Erkenntnisse überprüfen können Sie auf der Seite Knoten erkunden spezifische Details zu einem Knoten einsehen.



Node ID	Agent type	Agent version	Node status	Operating system name	Operating system type	Operating system version	Node type	Ac
i-00133d4e1c15e843b	amazon-ssm-agent	3.3.1230.0	Active	Ubuntu	Linux	20.04	EC2Instance	
i-00ba99a313b84a821	amazon-ssm-agent	3.3.1230.0	Active	Microsoft Windows Server 2022 Datacenter	Windows	10.0.20348	EC2Instance	
i-010e038ef4f248dbd	amazon-ssm-agent	3.3.1230.0	Active	Amazon Linux	Linux	2	EC2Instance	
i-013d9f572f5e5b6b3	amazon-ssm-agent	3.3.1230.0	Active	Microsoft Windows Server 2016 Datacenter	Windows	10.0.14393	EC2Instance	
i-018515ec864b6b34d	amazon-ssm-agent	3.3.987.0	Active	Ubuntu	Linux	24.04	EC2Instance	
i-0207b54c36e64ffac	amazon-ssm-agent	3.3.1230.0	Active	Amazon Linux	Linux	2	EC2Instance	
i-02384ada61f4a07a8	amazon-ssm-agent	3.3.1230.0	Active	Microsoft Windows Server 2019 Datacenter	Windows	10.0.17763	EC2Instance	

Registerkarten mit Knotendetails

Wenn Sie auf der Seite Knoten erkunden einen Knoten auswählen, erhalten Sie auf der Seite „Knotendetails“ einen Überblick über den Knoten. Eine Reihe von Registerkarten liefert zusätzliche Informationen:



Tags

(Optional) Mithilfe von Ressourcen-Tags können Sie den verwalteten Knoten nach anderen Ressourcen gruppieren und filtern. Tags bestehen aus einem groß-/kleinschreibungssensitiven Schlüssel-Wert-Paar und werden verwendet, um Ressourcen auf verschiedene Weise zu kategorisieren, beispielsweise nach Zweck, Eigentümer oder Umgebung.

Inventory

Zeigt Metadaten zum verwalteten Knoten an, die Sie nach über 10 verschiedenen Bestandstypen anzeigen können. Beispiel: Wenn Sie den Typ `AWS:Application` auswählen, zeigt der Bestandsfilter Details zu den auf dem Knoten installierten Anwendungen an, wie Name, Version, Architektur usw. Weitere Informationen zum Bestand finden Sie unter the section called “Bestand”.

Assoziationen

Eine Zuordnung ist ein Ressourcentyp in State Manager, der den Zielstatus für einen verwalteten Knoten definiert und dafür sorgt, dass alle verwalteten Knoten in Ihrem Konto in einem konsistenten Zustand bleiben. In einer Zuordnung können die Befehle, Skripts oder Richtlinien definiert sein, die für einzelne verwaltete Instances gelten sollten, und wie oft eine Zuordnung

ausgeführt werden soll, damit die Knoten der Knotenkonfiguration entsprechen. Eine Zuordnung vereinfacht zudem das Compliance-Reporting über den Status einzelner Ressourcen in Ihrem Konto. Weitere Informationen zu State Manager und Zuordnungen finden Sie unter the section called “State Manager”.

Patches

Zeigt Metadaten über den verwalteten Knoten an, z. B. welche Patch-Baseline dem Knoten zugewiesen ist, und die Gesamtzahl der Updates für Pakete, die erfolgreich aktualisiert wurden, fehlgeschlagen sind oder für die Installation erforderlich sind. Die Registerkarte Patches enthält auch Details zu den für den Knoten verfügbaren Patches basierend auf den Konfigurationsanforderungen in der Patch-Baseline. Dazu gehören der Name des Pakets, z. B. `libblockdev-crypto.x86_64`; Klassifizierung (z. B. `Bugfix` oder `Security`); Beschreibung (mit dem vollständigen Patch-Titel, z. B. `coreutils.x86_64:0:8.32-36.el9` und `java-11-amazon-corretto-headless-1:11.0.15+9-1.amzn2.x86_64`) und Patch-Status, z. B. `Installed`, `Installed_Pending_Reboot`, `Missing` und `Failed`.

> **Note**
>
> Der Patch-Status ist kein Hinweis darauf, ob ein verwalteter Knoten konform ist oder nicht. Patch-Compliance ist nicht von Natur aus an den Patch-Status gebunden und wird auch nicht von AWS Betriebssystemanbietern (OS) oder von Dritten wie Sicherheitsberatungsunternehmen definiert. Stattdessen definieren Sie in einer Patch-Baseline, was Patch-Compliance für verwaltete Knoten in Ihrer Organisation oder Ihrem Konto bedeutet. Weitere Informationen erhalten Sie unter the section called “Worin besteht Compliance in Patch Manager?” und the section called “Vordefinierte und benutzerdefinierte Patch-Baselines”.

Konfigurationskonformität

Informiert über die Patch-Konformität und Konfigurationsinkonsistenzen auf dem Knoten (unabhängig davon, ob der Status eines Pakets auf dem verwalteten Knoten `Compliant` oder `Non-compliant` gemäß der Definition von „konform“ lautet, wie entweder in einer Zuordnung in State Manager oder einer Patch-Baseline in Patch Manager definiert). Sie können die Konfigurationskonformitätsergebnisse nach Paket-ID, Konformitätsstatus, Konformitätstyp (`Association` oder `Patch`), `Severity` und verschiedenen Ausführungsdetails filtern. Zugehörige Informationen finden Sie unter the section called “Compliance”.

Wir empfehlen die Verwendung der einheitlichen Konsole, ganz gleich, ob Sie Knoten in mehreren Konten und Regionen in einer Organisation oder Knoten in einem einzigen Konto und einer Region haben. Weitere Informationen zu den Knotenaufgaben, die Sie jetzt mit der vereinheitlichten Konsole ausführen können, finden Sie unter Ausführen von Knotenverwaltungsaufgaben mit AWS Systems Manager.

Weitere Informationen zum Einrichten von Knoten für Systems Manager finden Sie unter Einrichten von verwalteten Knoten für AWS Systems Manager. Nachdem Sie Ihre Knoten eingerichtet haben, können Sie Systems Manager und die vereinheitlichte Konsole einrichten. Weitere Informationen zum Einrichten von Systems Manager finden Sie unter Einrichten von AWS Systems Manager.

AWS Systems ManagerChange ManagerÄnderung der Verfügbarkeit

Nach reiflicher Überlegung haben wir beschlossen, ab dem 7. November 2025 für Neukunden AWS Systems ManagerChange Manager zu schließen. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen.

Bewertung alternativer Dienste

Für Kunden, die als Ersatz eine Change-Management-Lösung der Enterprise-Klasse benötigen AWS Systems ManagerChange Manager, empfehlen wir den Umstieg auf eine der AWS Partner Network-Lösungen.

Im Leitfaden zur Cloud-Umgebung für Management und Governance wird empfohlen, dass Sie bei der Auswahl einer AWS Partnerlösung für das Servicemanagement die folgenden Fragen berücksichtigen:

- Bietet sie ITSM-Prozessunterstützung?
- Ermöglicht es Benutzern, ITSM-Prozesse in Beziehung zu setzen (d. h. Vorfälle mit Änderungsanfragen in Beziehung zu setzen)?
- Ermöglicht es Geschäftsabläufe und Genehmigungen?
- Ermöglicht es die Konfiguration, Anpassung und Integration in andere Systeme und Plattformen?
- Ermöglicht es die Erstellung von Berichten und Dashboards?
- Ermöglicht es Self-Service für Geschäfts- und Serviceanfragen?

Prüfen Sie sorgfältig Ihre spezifischen Anwendungsfälle und Anforderungen und überprüfen Sie anschließend die Funktionen der AWS Partner Network-Lösungen, um die für Ihre Bedürfnisse am besten geeignete Servicemanagement-Lösung auszuwählen.

Wenn Sie weitere Fragen haben, kontaktieren Sie uns bitte über das AWS Support Center oder lesen Sie unsere FAQs.

Einrichten von verwalteten Knoten für AWS Systems Manager

Führen Sie die Aufgaben in diesem Abschnitt aus, um Rollen, Benutzerkonten, Berechtigungen und erste Ressourcen für die Verwendung von AWS Systems Manager Tools einzurichten und zu konfigurieren. Die in diesem Abschnitt beschriebenen Aufgaben werden in der Regel von AWS-Konto Systemadministratoren ausgeführt. Nachdem Sie diese Schritte abgeschlossen sind, können Benutzer in Ihrer Organisation Systems Manager verwenden, um Ihre verwaltete Knoten zu konfigurieren, zu verwalten und darauf zuzugreifen. Ein verwalteter Knoten ist eine Maschine, die für die Verwendung mit Systems Manager in einer Hybrid- und Multi-Cloud-Umgebung konfiguriert ist.

> **Note**
>
> Wenn Sie EC2 Amazon-Instances und Ihre eigenen Rechenressourcen in einer Hybrid- und Multi-Cloud-Umgebung verwenden möchten, folgen Sie den Schritten unterVerwalten von EC2-Instances mit Systems Manager. In diesem Thema werden die Schritte in der richtigen Reihenfolge beschrieben, um das Systems Manager Manager-Setup für EC2 Instanzen und EC2 Nicht-Computer abzuschließen.

Wenn Sie bereits andere verwenden AWS-Services, haben Sie einige dieser Schritte abgeschlossen. Andere Schritte sind jedoch speziell auf den Systems Manager ausgelegt. Aus diesem Grund empfehlen wir, diesen gesamten Abschnitt zu lesen, um sicherzustellen, dass Sie mit der Funktionsweise aller Systems-Manager-Tools vertraut sind.

Themen

- Verwalten von EC2-Instances mit Systems Manager
- Verwalten von Knoten in Hybrid- und Multi-Cloud-Umgebungen mit Systems Manager
- Verwalten von Edge-Geräten mit Systems Manager
- Einen AWS Organizations delegierten Administrator für Systems Manager erstellen
- Allgemeine Einrichtung für AWS Systems Manager

Verwalten von EC2-Instances mit Systems Manager

Führen Sie die Aufgaben in diesem Abschnitt durch, um Rollen, Berechtigungen und ursprüngliche Ressourcen für AWS Systems Manager festzulegen und zu konfigurieren. Die in diesem Abschnitt beschriebenen Aufgaben werden in der Regel von AWS-Konto und Systemadministratoren ausgeführt. Nachdem Sie diese Schritte abgeschlossen sind, können Benutzer in Ihrer Organisation Systems Manager verwenden, um Amazon Elastic Compute Cloud (Amazon EC2)-Instances in Ihrem Konto zu konfigurieren, zu verwalten und darauf zuzugreifen.

> **Note**
>
> Wenn Sie mit Systems Manager On-Premises-Maschinen verwalten und konfigurieren möchten, befolgen Sie die Einrichtungsschritte in Verwalten von Knoten in Hybrid- und Multi-Cloud-Umgebungen mit Systems Manager. Wenn Sie planen, sowohl Amazon EC2 Instances als auch Nicht-EC2-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung zu verwenden, befolgen Sie zunächst die hier aufgeführten Schritte. Dieser Abschnitt beschreibt die empfohlene Reihenfolge der Schritte zur Konfiguration der Rollen, Benutzer, Berechtigungen und ersten Ressourcen für Ihre Systems-Manager.

Wenn Sie bereits andere AWS-Services verwenden, haben Sie einige dieser Schritte schon abgeschlossen. Andere Schritte sind jedoch speziell auf den Systems Manager ausgelegt. Aus diesem Grund empfehlen wir, diesen gesamten Abschnitt zu lesen, um sicherzustellen, dass Sie mit der Funktionsweise aller Systems Manager-Tools vertraut sind.

Inhalt

- Konfigurieren von erforderlichen Instance-Berechtigungen für Systems Manager
- Verbessern Sie die Sicherheit von EC2 Instanzen mithilfe von VPC-Endpunkten für Systems Manager

Konfigurieren von erforderlichen Instance-Berechtigungen für Systems Manager

Hat standardmäßig AWS Systems Manager keine Berechtigung, Aktionen auf Ihren Instances durchzuführen. Sie können Instanzberechtigungen auf Kontoebene mithilfe einer AWS Identity and Access Management (IAM-) Rolle oder auf Instanzebene mithilfe eines Instanzprofils gewähren.

Wenn Ihr Anwendungsfall dies zulässt, empfehlen wir, mithilfe der Standardkonfiguration für die Host-Verwaltung Zugriff auf Kontoebene zu gewähren.

> **Note**
>
> Sie können diesen Schritt überspringen und Systems Manager erlauben, bei der Einrichtung der vereinheitlichten Konsole die erforderlichen Berechtigungen auf Ihre Instances für Sie anzuwenden. Weitere Informationen finden Sie unter Einrichten von AWS Systems Manager.

Empfohlene Konfiguration für EC2 Instanzberechtigungen

Die Standard-Host-Management-Konfiguration ermöglicht es Systems Manager, Ihre EC2 Amazon-Instances automatisch zu verwalten. Nachdem Sie diese Einstellung aktiviert haben, werden alle Instances, die Instance Metadata Service Version 2 (IMDSv2) in der Version verwenden AWS-Region und AWS-Konto auf denen SSM Agent Version 3.2.582.0 oder höher installiert ist, automatisch zu verwalteten Instances. Die Standardkonfiguration für die Host-Verwaltung unterstützt die Instance-Metadaten-Service-Version 1 nicht. Informationen zur Umstellung auf IMDSv2 Version 2 finden Sie unter Umstellung auf die Nutzung von Instance Metadata Service Version 2 im EC2 Amazon-Benutzerhandbuch. Informationen zur Überprüfung der auf Ihrer Instance installierten Version des SSM Agent finden Sie unter Überprüfen der SSM Agent-Versionsnummer. Informationen zur Aktualisierung des SSM Agent finden Sie unter Automatische Aktualisierung von SSM Agent. Zu den Vorteilen verwalteter Instances gehören die folgenden:

- Stellen Sie mit Session Manager eine sichere Verbindung zu Ihren Instances her.
- Führen Sie automatisierte Patch-Scans mit Patch Manager durch.
- Zeigen Sie mit Systems Manager Inventory detaillierte Informationen zu Ihren Instances an.
- Verfolgen und verwalten Sie Instances mithilfe von Fleet Manager.
- Halten Sie den SSM Agent automatisch auf dem neuesten Stand.

Fleet Manager, Inventar Patch Manager und Session Manager sind Tools in AWS Systems Manager.

Die Standardkonfiguration für die Host-Verwaltung ermöglicht die Instance-Verwaltung ohne die Verwendung von Instance-Profilen und stellt sicher, dass Systems Manager über Berechtigungen zum Verwalten aller Instances in der Region und im Konto verfügt. Wenn die bereitgestellten Berechtigungen für Ihren Anwendungsfall nicht ausreichen, können Sie auch Richtlinien zur Standard-IAM-Rolle hinzufügen, die von der Standardkonfiguration für die Host-Verwaltung erstellt

wird. Wenn Sie keine Berechtigungen für alle Funktionen benötigen, die von der Standard-IAM-Rolle bereitgestellt werden, können Sie alternativ Ihre eigene benutzerdefinierte Rolle und Richtlinien erstellen. Alle Änderungen, die Sie an der IAM-Rolle vornehmen, die Sie für die Standard-Host-Management-Konfiguration auswählen, gelten für alle verwalteten EC2 Amazon-Instances in der Region und im Konto. Weitere Informationen zu der Richtlinie, die von der Standardkonfiguration für die Host-Verwaltung verwendet wird, finden Sie unter AWS verwaltete Richtlinie: Amazon SSMManaged EC2 InstanceDefaultPolicy. Weitere Informationen über die Standardkonfiguration für die Host-Verwaltung finden Sie unter Automatisches Verwalten von EC2 Instanzen mit der Standard-Host-Management-Konfiguration.

Important

Instances, die mit der Standardkonfiguration für die Host-Verwaltung registriert wurden, speichern Registrierungsinformationen lokal in den Verzeichnissen `/lib/amazon/ssm` oder `C:\ProgramData\Amazon`. Das Entfernen dieser Verzeichnisse oder der enthaltenen Dateien verhindert, dass die Instance die erforderlichen Anmeldeinformationen für die Verbindung mit Systems Manager über die Standardkonfiguration für die Host-Verwaltung erhält. In diesen Fällen müssen Sie ein Instance-Profil verwenden, um Ihrer Instance die erforderlichen Berechtigungen zu erteilen, oder die Instance neu erstellen.

Note

Dieses Verfahren sollte nur von Administratoren durchgeführt werden. Implementieren Sie den Zugriff mit den geringsten Berechtigungen, wenn Sie Einzelpersonen erlauben, die Standardkonfiguration für die Host-Verwaltung zu konfigurieren oder zu ändern. Sie müssen die Standard-Host-Management-Konfiguration für jede EC2 Instanz aktivieren, die AWS-Region Sie automatisch Amazon möchten.

So aktivieren Sie die Einstellung der Standardkonfiguration für die Host-Verwaltung

Sie können die Standardkonfiguration für die Host-Verwaltung über die Fleet Manager-Konsole aktivieren. Um dieses Verfahren mit dem AWS-Managementkonsole oder Ihrem bevorzugten Befehlszeilentool erfolgreich abschließen zu können, benötigen Sie Berechtigungen für die UpdateServiceSettingAPI-Operationen GetServiceSettingResetServiceSetting, und. Darüber hinaus müssen Sie über Berechtigungen für die `iam:PassRole`-Berechtigung für die

`AWSSystemsManagerDefaultEC2InstanceManagementRole`-IAM-Rolle verfügen. Es folgt eine Beispielrichtlinie . Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting",
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:servicesetting/ssm/
managed-instance/default-ec2-instance-management-role"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/service-role/
AWSSystemsManagerDefaultEC2InstanceManagementRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ssm.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

Bevor Sie beginnen: Wenn Sie Instance-Profile an Ihre EC2 Amazon-Instances angehängt haben, entfernen Sie alle Berechtigungen, die den `ssm:UpdateInstanceInformation` Vorgang zulassen. Der SSM Agent versucht, die Instance-Profilberechtigungen zu verwenden, bevor die Berechtigungen der Standardkonfiguration für die Host-Verwaltung verwendet werden. Wenn Sie die

`ssm:UpdateInstanceInformation`-Operation in Ihren Instance-Profilen zulassen, verwendet die Instance nicht die Berechtigungen der Standardkonfiguration für die Host-Verwaltung.

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie im Drop-Down-Menü Kontoverwaltung die Option Standardkonfiguration für die Host-Verwaltung konfigurieren aus.
4. Aktivieren Sie Standardkonfiguration für die Host-Verwaltung aktivieren.
5. Wählen Sie die IAM-Rolle aus, die zum Aktivieren von Systems-Manager-Tools für Ihre Instances verwendet wird. Wir empfehlen die Verwendung der Standardrolle, die in der Standardkonfiguration für die Host-Verwaltung bereitgestellt wird. Es enthält die Mindestberechtigungen, die für die Verwaltung Ihrer EC2 Amazon-Instances mit Systems Manager erforderlich sind. Wenn Sie es vorziehen, eine benutzerdefinierte Rolle zu verwenden, muss die Vertrauensrichtlinie der Rolle Systems Manager als vertrauenswürdige Entität zulassen.
6. Wählen Sie Konfigurieren, um die Einrichtung abzuschließen.

Nach dem Aktivieren der Standardkonfiguration für die Host-Verwaltung kann es 30 Minuten dauern, bis Ihre Instances die Anmeldeinformationen der von Ihnen ausgewählten Rolle verwenden. Sie müssen die Standard-Host-Management-Konfiguration in jeder Region aktivieren, in der Sie Ihre EC2 Amazon-Instances automatisch verwalten möchten.

Alternative Konfiguration für EC2 Instance-Berechtigungen

Sie können Zugriff auf der Ebene der einzelnen Instances gewähren, indem Sie ein AWS Identity and Access Management (IAM) Instance-Profil verwenden. Ein Instance-Profil ist ein Container, der beim Start IAM-Rolleninformationen an eine Amazon Elastic Compute Cloud (Amazon EC2) -Instance weitergibt. Sie können ein Instance-Profil für Systems Manager erstellen, indem Sie eine oder mehrere IAM-Richtlinien anfügen, die die erforderlichen Berechtigungen für eine neue Rolle oder eine bereits von ihnen erstellte Rolle definieren.

> **Note**
>
> Sie können ein Tool in verwenden Quick Setup AWS Systems Manager, um schnell ein Instance-Profil für alle Instances in Ihrer AWS-Konto zu konfigurieren. Quick Setuperstellt

außerdem eine IAM-Servicerolle (oder übernimmt die Rolle), sodass Systems Manager Befehle auf Ihren Instances in Ihrem Namen sicher ausführen kann. Mit Quick Setup können Sie diesen Schritt (Schritt 3) und Schritt 4 überspringen. Weitere Informationen finden Sie unter AWS Systems Manager Quick Setup.

Beachten Sie die folgenden Details zum Erstellen eines IAM-Instance-Profils:

- Wenn Sie EC2 Computer in einer Hybrid- und Multi-Cloud-Umgebung für Systems Manager konfigurieren, müssen Sie kein Instanzprofil für sie erstellen. Konfigurieren Sie stattdessen Ihre Server und verwenden VMs Sie eine IAM-Servicerolle. Weitere Informationen finden Sie unter Erstellen der für Systems Manager in Hybrid- und Multi-Cloud-Umgebungen erforderlichen IAM-Servicerolle.
- Wenn Sie das IAM-Instance-Profil ändern, kann es einige Zeit dauern, bis die Instance-Anmeldeinformationen aktualisiert werden. SSM Agent verarbeitet Anfragen erst, wenn dies erfolgt ist. Um den Aktualisierungsprozess zu beschleunigen, können Sie SSM Agent oder die Instance erneut starten.

Verwenden Sie eine der folgenden Vorgehensweisen, je nachdem, ob Sie eine neue Rolle für Ihr Instance-Profil erstellen oder die erforderlichen Berechtigungen zu einer vorhandenen Rolle hinzufügen.

Erstellen eines Instance-Profils für von Systems Manager verwaltete Instances (Konsole)

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen und dann Rolle erstellen.
3. Wählen Sie für Trusted entity type (Vertrauenswürdige Entität) die Option AWS-Service aus.
4. Wählen EC2Sie unmittelbar unter Anwendungsfall die Option und anschließend Weiter aus.
5. Gehen Sie auf der Seite Berechtigungen hinzufügen wie folgt vor:
   - Verwenden Sie das Suchfeld, um die SSMManagedInstanceCoreAmazon-Richtlinie zu finden. Aktivieren Sie das Kontrollkästchen neben dem Namen, wie in der folgenden Abbildung gezeigt.

Die Konsole behält Ihre Auswahl auch dann bei, wenn Sie nach anderen Richtlinien suchen.

- Wenn Sie im vorherigen Verfahren, (Optional) Erstellen einer benutzerdefinierten Richtlinie für den Zugriff auf einen S3-Bucket, eine benutzerdefinierte S3-Bucket-Richtlinie erstellt haben, suchen Sie danach und aktivieren Sie das Kontrollkästchen neben ihrem Namen.
- Wenn Sie Instances zu einem von verwalteten Active Directory hinzufügen möchten Directory Service, suchen Sie nach Amazon SSMDirectory ServiceAccess und aktivieren Sie das Kontrollkästchen neben dem Namen.
- Wenn Sie planen, Ihre Instance mithilfe von EventBridge oder CloudWatch Logs zu verwalten oder zu überwachen, suchen Sie nach dem entsprechenden Namen CloudWatchAgentServerPolicyund aktivieren Sie das Kontrollkästchen neben dem Namen.

6. Wählen Sie Weiter aus.
7. Geben Sie unter Role name (Rollenname) einen Namen für Ihr neues Instance-Profil ein, wie z. B. **`SSMInstanceProfile`**.

> **Note**
>
> Notieren Sie sich den Rollennamen. Sie wählen diese Rolle beim Erstellen neuer Instances, die Sie mit Systems Manager verwalten möchten.

8. (Optional) Aktualisieren Sie in Description (Beschreibung) die Beschreibung für dieses Instance-Profil ein.
9. (Optional) Fügen Sie für Tags ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Rolle zu organisieren, nachzuverfolgen oder zu steuern, und wählen Sie dann Create role (Rolle erstellen) aus. Das System leitet Sie zur Seite Rollen zurück.

So fügen Sie Instance-Profil-Berechtigungen für Systems Manager zu einer vorhandenen Rolle hinzu (Konsole)

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.

2. Klicken Sie im Navigationsbereich auf Rollen und wählen Sie die vorhandene Rolle aus, die Sie einem Instance-Profil für Systems Manager-Operationen zuordnen möchten.
3. Wählen Sie auf der Registerkarte Permissions (Berechtigungen) die Optionen Add permissions, Attach policies (Berechtigungen hinzufügen, Richtlinien anfügen) aus.
4. Führen Sie auf der Seite Attach Policy (Richtlinie anfügen) die folgenden Schritte aus:
   - Verwenden Sie das Suchfeld, um die SSMManagedInstanceCoreAmazon-Richtlinie zu finden. Aktivieren Sie das Kontrollkästchen neben dem Namen.
   - Wenn Sie eine benutzerdefinierte S3-Bucket-Richtlinie erstellt haben, suchen Sie danach und aktivieren Sie das Kontrollkästchen neben ihrem Namen. Weitere Informationen zu benutzerdefinierten S3-Bucket-Richtlinien für ein Instance-Profil finden Sie unter (Optional) Erstellen einer benutzerdefinierten Richtlinie für den Zugriff auf einen S3-Bucket.
   - Wenn Sie Instances zu einem von verwalteten Active Directory hinzufügen möchten Directory Service, suchen Sie nach Amazon SSMDirectory ServiceAccess und aktivieren Sie das Kontrollkästchen neben dem Namen.
   - Wenn Sie planen, Ihre Instance mithilfe von EventBridge oder CloudWatch Logs zu verwalten oder zu überwachen, suchen Sie nach dem entsprechenden Namen CloudWatchAgentServerPolicyund aktivieren Sie das Kontrollkästchen neben dem Namen.
5. Wählen Sie Richtlinien anfügen.

Informationen zum Aktualisieren einer Rolle mit einer vertrauenswürdigen juristischen Stelle oder zur weiteren Einschränkung des Zugriffs finden Sie unter Ändern einer Rolle im IAM-Benutzerhandbuch.

(Optional) Erstellen einer benutzerdefinierten Richtlinie für den Zugriff auf einen S3-Bucket

Es muss nur dann eine benutzerdefinierte Richtlinie für Amazon S3-Zugriff erstellt werden, wenn Sie einen VPC-Endpunkt oder einen eigenen S3 Bucket in Ihren Systems Manager-Operationen verwenden. Sie können diese Richtlinie an die Standard-IAM-Rolle anfügen, die Sie mit der Standardkonfiguration für die Host-Verwaltung erstellt haben, oder an ein Instance-Profil, das Sie mit dem vorherigen Verfahren erstellt haben.

Informationen zu den AWS verwalteten S3-Buckets, auf die Sie in der folgenden Richtlinie Zugriff gewähren, finden Sie unterSSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets.

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.

2. Wählen Sie im Navigationsbereich Richtlinien und dann Richtlinie erstellen.
3. Wählen Sie die Registerkarte JSON und ersetzen Sie den Standard-Text durch den folgenden.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": [
                "arn:aws:s3:::aws-ssm-us-east-2/*",
                "arn:aws:s3:::aws-windows-downloads-us-east-2/*",
                "arn:aws:s3:::amazon-ssm-us-east-2/*",
                "arn:aws:s3:::amazon-ssm-packages-us-east-2/*",
                "arn:aws:s3:::us-east-2-birdwatcher-prod/*",
                "arn:aws:s3:::aws-ssm-distributor-file-us-east-2/*",
                "arn:aws:s3:::aws-ssm-document-attachments-us-east-2/*",
                "arn:aws:s3:::patch-baseline-snapshot-us-east-2/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:GetEncryptionConfiguration"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*",
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        }
    ]
}
```

> **Note**
>
> Das erste `Statement`-Element ist nur erforderlich, wenn Sie einen VPC-Endpunkt verwenden.
> Das zweite `Statement`-Element ist nur erforderlich, wenn Sie einen S3 Bucket verwenden, den Sie zur Verwendung bei Ihren Systems Manager-Operationen erstellt haben.
> Die `PutObjectAcl`-ACL-Berechtigung ist nur erforderlich, wenn Sie vorhaben, kontoübergreifenden Zugriff auf S3-Buckets in anderen Konten zu unterstützen.
> Das `GetEncryptionConfiguration`-Element ist erforderlich, wenn Ihr S3-Bucket für die Verwendung der Verschlüsselung konfiguriert ist.
> Wenn Ihr S3 Bucket für die Verwendung der Verschlüsselung konfiguriert ist, muss das S3-Bucket-Stammverzeichnis (z. B. `arn:aws:s3:::amzn-s3-demo-bucket`) im Abschnitt Ressource aufgeführt werden. Ihr Benutzer, Ihre Gruppe oder Ihre Rolle muss mit Zugriff auf den Stamm-Bucket konfiguriert sein.

4. Wenn Sie einen VPC-Endpunkt in Ihren Operationen verwenden, gehen Sie wie folgt vor:

   Ersetzen Sie im ersten `Statement` Element jeden *`region`* Platzhalter durch den Bezeichner der Richtlinie, in der AWS-Region diese Richtlinie verwendet werden soll. Verwenden Sie beispielsweise `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *`region`* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

   > **Important**
   >
   > Wir empfehlen, dass Sie keine Platzhalterzeichen (*) für bestimmte Regionen in dieser Richtlinie verwenden. Verwenden Sie beispielsweise `arn:aws:s3:::aws-ssm-us-east-2/*` und nicht `arn:aws:s3:::aws-ssm-*/*`. Bei der Verwendung von Platzhaltern könnte Zugriff auf S3-Buckets erteilt werden, für die Sie keinen Zugriff gewähren möchten. Wenn Sie das Instance-Profil für mehr als eine Region verwenden möchten, empfehlen wir, das erste `Statement`-Element für jede Region zu wiederholen.

   –oder–

Wenn Sie in Ihren Operationen keinen VPC-Endpunkt verwenden, können Sie das erste `Statement`-Element löschen.

5. Wenn Sie einen eigenen S3-Bucket in Ihren Systems Manager-Operationen verwenden, gehen Sie wie folgt vor:

   Ersetzen Sie das zweite `Statement` Element *`amzn-s3-demo-bucket`* durch den Namen eines S3-Buckets in Ihrem Konto. Dieser Bucket wird nun für Ihre Systems Manager-Operationen verwendet. Er bietet die Berechtigung für Objekte im Bucket, wobei `"arn:aws:s3:::my-bucket-name/*"` als Ressource verwendet wird. Weitere Informationen zur Bereitstellung von Berechtigungen für Buckets oder Objekte in Buckets finden Sie im Thema Amazon S3 S3-Aktionen im Amazon Simple Storage Service-Benutzerhandbuch und im AWS Blogbeitrag IAM-Richtlinien und Bucket-Richtlinien und! ACLs Oh, My! (Steuern des Zugriffs auf S3-Ressourcen).

   > **Note**
   >
   > Wenn Sie mehr als einen Bucket verwenden, geben Sie den ARN für jeden davon an. Im folgenden Beispiel finden Sie Berechtigungen für Buckets.
   >
   > ```
   > "Resource": [
   > "arn:aws:s3:::amzn-s3-demo-bucket1/*",
   > "arn:aws:s3:::amzn-s3-demo-bucket2/*"
   >              ]
   > ```

   –oder–

   Wenn Sie bei Ihren Systems Manager-Operationen keinen eigenen S3 Bucket verwenden, können Sie das zweite `Statement`-Element löschen.

6. Wählen Sie Weiter: Tags aus.
7. (Optional) Fügen Sie Tags hinzu, indem Sie Tag hinzufügen auswählen und die bevorzugten Tags für die Richtlinie eingeben.
8. Wählen Sie Weiter: Prüfen aus.
9. Geben Sie unter Name einen Namen für diese Richtlinie, z. B. **`SSMInstanceProfileS3Policy`**, ein.
10. Wählen Sie Richtlinie erstellen aus.

Zusätzliche Richtlinienüberlegungen für verwaltete Instances

In diesem Abschnitt werden einige der Richtlinien beschrieben, die Sie der Standard-IAM-Rolle hinzufügen können, die von der Standardkonfiguration für die Host-Verwaltung oder Ihren Instance-Profilen für AWS Systems Manager erstellt wurde. Um Berechtigungen für die Kommunikation zwischen Instances und der Systems-Manager-API zu erteilen, empfehlen wir, benutzerdefinierte Richtlinien zu erstellen, die Ihre System- und Sicherheitsanforderungen berücksichtigen. Abhängig von Ihrem Betriebsplan benötigen Sie möglicherweise Berechtigungen, die in einer oder mehreren der anderen Richtlinien dargestellt werden.

Richtlinie: **`AmazonSSMDirectoryServiceAccess`**

Nur erforderlich, wenn Sie EC2 Amazon-Instances einem Microsoft AD-Verzeichnis hinzufügen möchten. Windows Server

Diese AWS verwaltete Richtlinie ermöglicht SSM Agent den Zugriff in Ihrem Namen AWS Directory Service auf Anfragen der verwalteten Instance, der Domain beizutreten. Weitere Informationen finden Sie unter Nahtloses Beitreten zu einer EC2 Windows-Instanz im AWS Directory Service Administratorhandbuch.

Richtlinie: **`CloudWatchAgentServerPolicy`**

Nur erforderlich, wenn Sie planen, den CloudWatch Agenten auf Ihren Instances zu installieren und auszuführen, um Metrik- und Protokolldaten auf einer Instance zu lesen und in Amazon zu schreiben CloudWatch. Diese helfen Ihnen dabei, Probleme oder Änderungen an Ihren AWS Ressourcen zu überwachen, zu analysieren und schnell darauf zu reagieren.

Ihre standardmäßige IAM-Rolle, die mit der Standard-Host-Management-Konfiguration oder dem Instance-Profil erstellt wurde, benötigt diese Richtlinie nur, wenn Sie Funktionen wie Amazon EventBridge oder Amazon CloudWatch Logs verwenden. (Sie können auch eine restriktivere Richtlinie erstellen, die beispielsweise den Schreibzugriff auf einen bestimmten CloudWatch Logs-Log-Stream einschränkt.)

> **Note**
>
> Die Verwendung der Funktionen EventBridge und CloudWatch Protokollierung ist optional. Wenn Sie sich jedoch hierzu entschlossen haben, sollte diese zu Beginn des Systems-Manager-Konfigurationsprozesses eingerichtet werden. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch und im Amazon CloudWatch Logs-Benutzerhandbuch.

Informationen zum Erstellen von IAM-Richtlinien mit Berechtigungen für zusätzliche Systems-Manager-Tools finden Sie in den folgenden Ressourcen:

- Einschränken des Zugriffs auf Parameter Store-Parameter mithilfe von IAM-Richtlinien
- Einrichten der Automatisierung
- Schritt 2: Überprüfen oder Hinzufügen von Instance-Berechtigungen für Session Manager

Anfügen des Systems-Manager-Instance-Profils an eine Instance (Konsole)

Das folgende Verfahren beschreibt, wie Sie mithilfe der EC2 Amazon-Konsole ein IAM-Instance-Profil an eine EC2 Amazon-Instance anhängen.

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich unter Instances die Option Instances.
3. Navigieren Sie zu Ihrer EC2 Instance und wählen Sie sie aus der Liste aus.
4. Wählen Sie im Menü Actions (Aktionen) die Option Security (Sicherheit), Modify IAM role (IAM-Rolle ändern).
5. Wählen Sie für die IAM-Rolle das Instance-Profil aus, das Sie mithilfe des Verfahrens in Alternative Konfiguration für EC2 Instance-Berechtigungen erstellt haben.
6. Wählen Sie Aktualisieren der IAM-Rolle.

Für weitere Informationen zum Anhängen von IAM-Rollen an Instances, wählen Sie, je nach Ihrem ausgewählten Betriebssystem, einen der folgenden Schritte aus:

- Fügen Sie einer Instance im EC2 Amazon-Benutzerhandbuch eine IAM-Rolle hinzu
- Fügen Sie einer Instance im EC2 Amazon-Benutzerhandbuch eine IAM-Rolle hinzu

Fahren Sie fort mit Verbessern Sie die Sicherheit von EC2 Instanzen mithilfe von VPC-Endpunkten für Systems Manager.

Verbessern Sie die Sicherheit von EC2 Instanzen mithilfe von VPC-Endpunkten für Systems Manager

Sie können die Sicherheitslage Ihrer verwalteten Knoten (einschließlich EC2 Nicht-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung) verbessern, indem Sie die Verwendung eines VPC-

Schnittstellen-Endpunkts in Amazon Virtual Private Cloud (Amazon VPC) konfigurieren AWS Systems Manager . Mithilfe eines Schnittstellen-VPC-Endpunkts (Schnittstellenendpunkt) können Sie eine Verbindung zu Diensten herstellen, die von bereitgestellt werden AWS PrivateLink. AWS PrivateLink ist eine Technologie, mit der Sie privat auf Amazon Elastic Compute Cloud (Amazon EC2) und Systems Manager zugreifen können, APIs indem Sie private IP-Adressen verwenden.

AWS PrivateLink schränkt den gesamten Netzwerkverkehr zwischen Ihren verwalteten Instances, Systems Manager und Amazon EC2 auf das Amazon-Netzwerk ein. Dies bedeutet, dass Ihre verwalteten Instances keinen Zugriff auf das Internet haben. Wenn Sie verwenden AWS PrivateLink, benötigen Sie kein Internet-Gateway, kein NAT-Gerät oder ein virtuelles privates Gateway.

Eine Konfiguration ist nicht erforderlich AWS PrivateLink, wird aber empfohlen. Weitere Informationen zu AWS PrivateLink VPC-Endpunkten finden Sie unter AWS PrivateLink und VPC-Endpoints.

> **Note**
>
> Die Alternative zur Verwendung eines VPC-Endpunkts ist das Aktivieren von ausgehendem Internetzugriff auf Ihren verwalteten Instances. In diesem Fall müssen die verwalteten Instances auch ausgehenden HTTPS-Datenverkehr (Port 443) zu den folgenden Endpunkten zulassen:
>
> - `ssm.region.amazonaws.com`
> - `ssmmessages.region.amazonaws.com`
> - `ec2messages.region.amazonaws.com`
>
> SSM Agent initiiert alle Verbindungen zum Systems Manager-Service in der Cloud. Aus diesem Grund müssen Sie Ihre Firewall nicht so konfigurieren, dass eingehender Datenverkehr zu Ihren Instances für Systems Manager zugelassen wird. Weitere Informationen über Anrufe an diese Endpunkte finden Sie unter Referenz: ec2messages, ssmmessages und andere API-Operationen.
> Wenn Sie Systems Manager in einer Umgebung verwenden, die nur unterstützt IPv6, müssen Sie auch ausgehenden Datenverkehr zu den folgenden Endpunkten zulassen:
>
> - `ssm.region.api.aws`
> - `ssmmessages.region.api.aws`
> - `ec2messages.region.api.aws`

Weitere Informationen zu Dual-Stack-Dienstendpunkten finden Sie unter Dual-Stack-Endpunkte im Allgemeinen Referenzhandbuch.AWS
Sie müssen außerdem sicherstellen, dass die Patch-Operations-Buckets von Ihren Knoten aus erreichbar sind, wie unter Referenz: Amazon S3 S3-Buckets für Patch-Operationen beschrieben.

Über Amazon VPC

Sie können Amazon Virtual Private Cloud (Amazon VPC) verwenden, um ein virtuelles Netzwerk in Ihrem eigenen logisch isolierten Bereich innerhalb der AWS Cloud sogenannten Virtual Private Cloud (VPC) zu definieren. Sie können Ihre AWS -Ressourcen, z. B. Instances, in Ihrer VPC launchen. Eine VPC ist einem herkömmlichen Netzwerk in einem eigenen Rechenzentrum sehr ähnlich, bietet jedoch die Vorteile durch die Nutzung der skalierbaren Infrastruktur von AWS. Sie können Ihre VPC konfigurieren. Hierzu können Sie den IP-Adressbereich auswählen, Subnetze erstellen sowie Routing-Tabellen, Netzwerk-Gateways und Sicherheitseinstellungen konfigurieren. Sie können jetzt Instances in der VPC mit dem Internet verbinden. Sie können Ihre VPC mit Ihrem eigenen Unternehmensrechenzentrum verbinden und sie so zu AWS Cloud einer Erweiterung Ihres Rechenzentrums machen. Um die Ressourcen in den einzelnen Subnetzen zu schützen, können Sie mehrere Sicherheitsebenen verwenden, darunter Sicherheitsgruppen und Netzwerk-Zugriffskontrolllisten. Weitere Informationen finden Sie im Amazon-VPC-Benutzerhandbuch.

Themen

- Beschränkungen und Einschränkungen bei VPC-Endpunkten
- Erstellen von VPC-Endpunkten für Systems Manager
- Erstellen einer Schnittstellen-VPC-Endpunkt-Richtlinie

Beschränkungen und Einschränkungen bei VPC-Endpunkten

Seien Sie sich der folgenden Einschränkungen bewusst, bevor Sie VPC-Endpunkte für Systems Manager konfigurieren.

VPC-Peering-Verbindungen

Der Zugriff auf VPC-Schnittstellenendpunkte erfolgt sowohl über intraregionale als auch interregionale VPC-Peering-Verbindungen. Weitere Informationen zu VPC-Peering-

Verbindungsanforderungen für VPC-Schnittstellenendpunkte finden Sie unter VPC-Peering-Verbindungen (Kontingente) im Benutzerhandbuch zu Amazon Virtual Private Cloud.

VPC-Gateway-Endpunktverbindungen können nicht auf einen Bereich außerhalb einer VPC erweitert werden. Ressourcen am anderen Ende einer VPC-Peering-Verbindung in Ihrer VPC können nicht über den Gateway-Endpunkt mit Ressourcen im Gateway-Endpunktservice kommunizieren. Weitere Informationen zu VPC-Peering-Verbindungsanforderungen für VPC-Gateway-Endpunkte finden Sie unter VPC-Endpunkte (Kontingente) im Benutzerhandbuch zu Amazon Virtual Private Cloud

Eingehende Verbindungen

Die Sicherheitsgruppe für den VPC-Endpunkt müssen eingehende Verbindungen auf Port 443 aus dem privaten Subnetz der verwalteten Instance zulassen. Wenn eingehende Verbindungen nicht zulässig sind, kann die verwaltete Instanz keine Verbindung zum SSM und EC2 zu den Endpunkten herstellen.

DNS-Auflösung

Wenn Sie einen benutzerdefinierten DNS-Server verwenden, müssen Sie dem Amazon-DNS-Server für Ihre VPC einen bedingten Weiterleiter für alle Abfragen an die `amazonaws.com`-Domain hinzufügen.

S3-Buckets

Ihre VPC-Endpunktrichtlinie muss mindestens Zugriff auf die folgenden Amazon-S3-Buckets in SSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets gewähren.

> **Note**
>
> Wenn Sie eine On-Premises-Firewall verwenden und Patch Manager nutzen möchten, muss diese Firewall auch den Zugriff auf den geeigneten Patch-Baseline-Endpunkt zulassen.

CloudWatch Amazon-Protokolle

Wenn Sie Ihren Instances nicht erlauben, auf das Internet zuzugreifen, erstellen Sie einen VPC-Endpunkt für CloudWatch Logs, um Funktionen zu verwenden, die Logs an CloudWatch Logs senden. Weitere Informationen zum Erstellen eines Endpunkts für CloudWatch Logs finden Sie unter Creating a VPC Endpoint for CloudWatch Logs im Amazon CloudWatch Logs-Benutzerhandbuch.

DNS in einer Hybrid- und Multi-Cloud-Umgebung

Informationen zur Konfiguration von DNS für die Verwendung mit AWS PrivateLink Endpunkten in Hybrid- und Multicloud-Umgebungen finden Sie unter Private DNS für Schnittstellenendpunkte im Amazon VPC-Benutzerhandbuch. Wenn Sie Ihre eigene DNS verwenden möchten, können Sie Route 53-Resolver nutzen. Weitere Informationen finden Sie unter Auflösen von DNS-Abfragen zwischen VPCs und Ihrem Netzwerk im Amazon Route 53-Entwicklerhandbuch.

Erstellen von VPC-Endpunkten für Systems Manager

Verwenden Sie die folgenden Informationen, um VPC-Schnittstellenendpunkte für zu erstellen. AWS Systems Manager Dieses Thema verweist auf Verfahren im Amazon VPC User Guide.

> **Note**
>
> *region*stellt den Bezeichner für eine Region dar AWS Systems Manager, die von AWS-Region unterstützt wird, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

Befolgen Sie die Schritte unter Erstellen eines Schnittstellen-Endpunkts zum Erstellen der folgenden Schnittstellen-Endpunkte:

- **com.amazonaws.*region*.ssm** – Der Endpunkt für den Systems-Manager-Service.
- **com.amazonaws.*region*.ec2messages** – Systems Manager verwendet diesen Endpunkt, um Anrufe vom SSM Agent an den Systems-Manager-Service zu tätigen. Ab Version 3.3.40.0 von SSM Agent begann Systems Manager, den `ssmmessages:*`-Endpunkt (Amazon Message Gateway Service), wann immer verfügbar, anstelle des `ec2messages:*`-Endpunkts (Amazon Message Delivery Service) zu verwenden.
- **com.amazonaws.*region*.ec2**— Wenn Sie Systems Manager verwenden, um VSS-fähige Snapshots zu erstellen, müssen Sie sicherstellen, dass Sie über einen Endpunkt für den Service verfügen. EC2 Wenn der EC2 Endpunkt nicht definiert ist, schlägt ein Aufruf zur Aufzählung angehängter Amazon EBS-Volumes fehl, was dazu führt, dass der Systems Manager Manager-Befehl fehlschlägt.
- **com.amazonaws.*region*.s3** – Systems Manager verwendet diesen Endpunkt für die AktualisierungSSM Agent. Systems Manager verwendet diesen Endpunkt auch, wenn Sie optional Skripts oder andere in Buckets gespeicherte Dateien abrufen oder Ausgabeprotokolle in einen Bucket hochladen möchten. Wenn die mit Ihren Instances verknüpfte Sicherheitsgruppe den

ausgehenden Datenverkehr einschränkt, müssen Sie eine Regel hinzufügen, um Datenverkehr zur Präfixliste für Amazon S3 zuzulassen. Weitere Informationen finden Sie unter Modify your security group im AWS PrivateLink -Guide.

- **`com.amazonaws.region.ssmmessages`** – Dieser Endpunkt wird benötigt, damit SSM Agent mit dem Systems-Manager-Service kommunizieren kann, für Run Command und wenn Sie über einen sicheren Datenkanal eine Verbindung zu Ihren Instances mit Session Manager herstellen. Weitere Informationen erhalten Sie unter AWS Systems Manager Session Manager und Referenz: ec2messages, ssmmessages und andere API-Operationen.
- (Optional) **`com.amazonaws.region.kms`**— Erstellen Sie diesen Endpunkt, wenn Sie die Verschlüsselung AWS Key Management Service (AWS KMS) für Session Manager oder Parameter verwenden möchten. Parameter Store
- (Optional) **`com.amazonaws.region.logs`**— Erstellen Sie diesen Endpunkt, wenn Sie Amazon CloudWatch Logs (CloudWatch Logs) fürSession Manager,Run Command, oder SSM Agent Logs verwenden möchten.

Informationen zu den AWS verwalteten S3-Buckets, auf die zugegriffen werden SSM Agent muss, finden Sie unterSSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets. Wenn Sie in Ihren Systems Manager-Vorgängen einen VPC-Endpunkt (Virtual Private Cloud) verwenden, müssen Sie in einem EC2 Instanzprofil für Systems Manager oder in einer Servicerolle für nicht EC2 verwaltete Knoten in einer Hybrid- und Multi-Cloud-Umgebung ausdrückliche Berechtigungen erteilen.

Erstellen einer Schnittstellen-VPC-Endpunkt-Richtlinie

Sie können Richtlinien für VPC-Schnittstellen-Endpoints erstellen, für AWS Systems Manager die Sie Folgendes angeben können:

- Der Prinzipal, der die Aktionen ausführen kann
- Aktionen, die ausgeführt werden können
- Ressourcen, für die Aktionen ausgeführt werden können

Weitere Informationen finden Sie unter Steuerung des Zugriffs auf Services mit VPC-Endpunkten im Benutzerhandbuch zu Amazon VPC.

Verwalten von Knoten in Hybrid- und Multi-Cloud-Umgebungen mit Systems Manager

Sie können AWS Systems Manager damit sowohl Amazon Elastic Compute Cloud (EC2) -Instances als auch eine Reihe von EC2 Maschinentypen verwalten. In diesem Abschnitt werden die Einrichtungsaufgaben beschrieben, die Konto- und Systemadministratoren ausführen, um EC2 Computer mit Systems Manager in einer Hybrid- und Multi-Cloud-Umgebung zu verwalten. Nachdem diese Schritte abgeschlossen sind, können Benutzer, denen der AWS-Konto Administrator Berechtigungen erteilt hat, Systems Manager verwenden, um die EC2 Nicht-Computer ihrer Organisation zu konfigurieren und zu verwalten.

Jede Maschine, die für die Verwendung mit Systems Manager konfiguriert wurde, wird als verwalteter Knoten bezeichnet.

> **Note**
>
> - Sie können Edge-Geräte als verwaltete Knoten registrieren, indem Sie dieselben Schritte zur Hybrid-Aktivierung verwenden, die auch für andere Geräte verwendet werden, die keine Computer sind. EC2 Zu diesen Arten von Edge-Geräten gehören sowohl Geräte als auch AWS IoT Geräte, bei denen es sich nicht AWS IoT um Geräte handelt. Verwenden Sie den in diesem Abschnitt beschriebenen Prozess, um diese Arten von Edge-Geräten einzurichten.
>
>   Systems Manager unterstützt auch Edge-Geräte, die AWS IoT Greengrass Core-Software verwenden. Der Einrichtungsprozess und die Anforderungen für AWS IoT Greengrass Core-Geräte unterscheiden sich von denen für Edge-Geräte AWS IoT und Edge-Geräte, bei denen es sich nicht um AWS Edge-Geräte handelt. Informationen zur Registrierung von AWS IoT Greengrass Geräten für die Verwendung mit Systems Manager finden Sie unterVerwalten von Edge-Geräten mit Systems Manager.
> - EC2 macOSNicht-Maschinen werden für Hybrid- und Multicloud-Umgebungen von Systems Manager nicht unterstützt.

Wenn Sie planen, Systems Manager zur Verwaltung von Amazon Elastic Compute Cloud (Amazon EC2) -Instances zu verwenden oder sowohl EC2 Amazon-Instances als auch EC2 Nicht-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung zu verwenden, folgen Sie Verwalten von EC2-Instances mit Systems Manager zunächst den Schritten unter.

Nachdem Sie Ihre Hybrid- und Multi-Cloud-Umgebung für Systems Manager konfiguriert haben, können Sie Folgendes tun:

- Erstellen Sie eine konsistente und sichere Möglichkeit, Hybrid- und Multi-Cloud-Workloads mit denselben Tools oder Skripts von einem Remote-Standort aus zu verwalten.
- Zentralisieren Sie die Zugriffskontrolle für Aktionen, die mithilfe von AWS Identity and Access Management (IAM) auf Ihren Computern ausgeführt werden können.
- Zentralisieren Sie die Überwachung der auf Ihren Maschinen durchgeführten Vorgänge, indem Sie die in AWS CloudTrail aufgezeichnete API-Aktivität anzeigen.

  Informationen CloudTrail zur Überwachung von Systems Manager Manager-Aktionen finden Sie unterProtokollierung von AWS Systems Manager-API-Aufrufen mit AWS CloudTrail.
- Zentralisieren Sie die Überwachung, indem Sie Amazon EventBridge und Amazon Simple Notification Service (Amazon SNS) so konfigurieren, dass Benachrichtigungen über die erfolgreiche Ausführung des Dienstes gesendet werden.

  Informationen EventBridge zur Überwachung von Systems Manager Manager-Ereignissen finden Sie unterÜberwachung von Systems Manager-Ereignissen mit Amazon EventBridge.

Informationen zu verwalteten Knoten

Nachdem Sie die Konfiguration Ihrer EC2 Nicht-Computer für Systems Manager wie in diesem Abschnitt beschrieben abgeschlossen haben, werden Ihre hybridaktivierten Maschinen in den Knoten aufgeführt AWS-Managementkonsole und als verwaltete Knoten beschrieben. In der Konsole werden Ihre hybridaktivierten verwalteten Knoten mit dem Präfix „mi-“ von EC2 Amazon-Instances unterschieden. IDs EC2Amazon-Instances IDs verwenden das Präfix „i-“.

Eine verwalteter Knoten ist jede für Systems Manager konfigurierte Maschine. Bisher wurden alle verwalteten Knoten als verwaltete Instances bezeichnet. Der Begriff Instance bezieht sich jetzt nur noch auf EC2 Instances. Der deregister-managed-instanceBefehl wurde vor dieser Terminologieänderung benannt.

Weitere Informationen finden Sie unter Arbeiten mit verwalteten Knoten.

> **Important**
>
> Es wird dringend empfohlen, die Verwendung von Betriebssystemversionen zu vermeiden, die End-of-Life (EOL) erreicht haben. Betriebssystemanbieter, darunter auch, bieten AWS in

der Regel keine Sicherheitspatches oder andere Updates für Versionen an, die EOL erreicht haben. Die weitere Verwendung eines EOL-Systems erhöht das Risiko, dass Upgrades, einschließlich Sicherheitsupdates, und andere Betriebsprobleme nicht durchgeführt werden können, erheblich. AWS testet die Systems Manager Manager-Funktionalität nicht auf Betriebssystemversionen, die EOL erreicht haben.

Informationen zum Instance-Kontingent

Systems Manager bietet eine Stufe mit Standardinstanzen und eine Stufe mit erweiterten Instanzen für nicht EC2 verwaltete Knoten in Ihrer Hybrid- und Multi-Cloud-Umgebung. Mit dem Standard-Instances-Kontingent erlaubt Ihnen maximal 1 000 hybridaktivierte Maschinen pro AWS-Konto pro AWS-Region zu registrieren. Wenn Sie mehr als 1.000 EC2 Nicht-Computer in einem einzigen Konto und einer Region registrieren müssen, verwenden Sie die Stufe „Advanced-Instances“. Mit Advanced-Instances können Sie auch eine Verbindung zu Ihren EC2 Nicht-Rechnern herstellen, indem Sie AWS Systems Manager Session Manager Session Managerbietet interaktiven Shell-Zugriff auf Ihre verwalteten Knoten.

Weitere Informationen finden Sie unter Konfigurieren von Instance-Kontingenten.

Themen

- Erstellen Sie die für Systems Manager in Hybrid- und Multi-Cloud-Umgebungen erforderliche IAM-Servicerolle
- Eine Hybridaktivierung erstellen, um Knoten bei Systems Manager zu registrieren
- Installation von SSM Agent auf hybriden Linux-Knoten
- Installation von SSM Agent auf hybriden Windows Server-Knoten

Erstellen Sie die für Systems Manager in Hybrid- und Multi-Cloud-Umgebungen erforderliche IAM-Servicerolle

Maschinen, die nicht EC2 (Amazon Elastic Compute Cloud) in einer Hybrid- und Multi-Cloud-Umgebung sind, benötigen eine AWS Identity and Access Management (IAM) -Servicerolle, um mit dem AWS Systems Manager Service zu kommunizieren. Die Rolle gewährt AWS Security Token Service (AWS STS) `AssumeRole` Zugriff auf den Systems Manager-Dienst. Sie müssen nur einmal eine Servicerolle für eine Hybrid- und Multi-Cloud-Umgebung erstellen, und zwar für jedes AWS-Konto. Sie können jedoch mehrere Servicerollen für verschiedene Hybrid- und Multi-Cloud-

Aktivierungen erstellen, wenn Maschinen in Ihrer Hybrid-Umgebung unterschiedliche Berechtigungen benötigen.

Im Folgenden wird beschrieben, wie Sie die erforderliche Servicerolle mit der Systems-Manager-Konsole oder Ihrem bevorzugten Befehlszeilen-Tool erstellen.

Verwenden von AWS-Managementkonsole , um eine IAM-Servicerolle für Systems-Manager-Hybrid-Aktivierungen zu erstellen

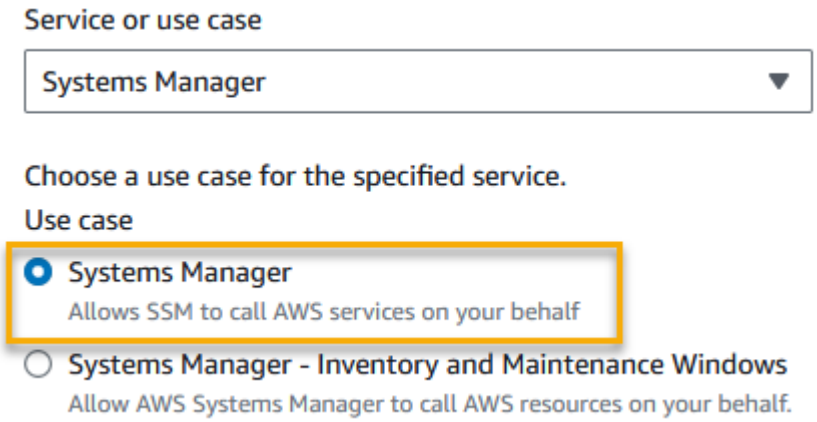
Verwenden Sie das folgende Verfahren zum Erstellen einer Servicerolle für eine Hybrid-Aktivierung. Dieses Verfahren verwendet die `AmazonSSMManagedInstanceCore`-Richtlinie für die Kernfunktionalität von Systems Manager. Je nach Anwendungsfall müssen Sie Ihrer Servicerolle möglicherweise zusätzliche Richtlinien hinzufügen, damit Ihre On-Premises-Maschinen auf andere Systems-Manager-Tools oder AWS-Services zugreifen können. Ohne Zugriff auf die erforderlichen AWS verwalteten Amazon Simple Storage Service (Amazon S3) -Buckets schlagen beispielsweise Patch Manager Patch-Vorgänge fehl.

So erstellen Sie eine -Servicerolle (Konsole)

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen und dann Rolle erstellen.
3. Wählen Sie für Select trusted entity (Vertrauenswürdige Entität auswählen) die folgenden Optionen:

    1. Wählen Sie für Vertrauenswürdige Entität die Option AWS-Service aus.
    2. Wählen Sie Systems Manager für Anwendungsfälle für andere AWS-Services.
    3. Wählen Sie Systems Manager aus.

        In der folgenden Abbildung wird die Position der Systems-Manager-Option hervorgehoben.



4. Wählen Sie Weiter aus.
5. Gehen Sie auf der Seite Berechtigungen hinzufügen wie folgt vor:

   - Verwenden Sie das Suchfeld, um die SSMManagedInstanceCoreAmazon-Richtlinie zu finden. Aktivieren Sie das Kontrollkästchen neben dem Namen, wie in der folgenden Abbildung gezeigt.

> **Note**
>
> Die Konsole behält Ihre Auswahl auch dann bei, wenn Sie nach anderen Richtlinien suchen.

   - Wenn Sie im Verfahren (Optional) Erstellen einer benutzerdefinierten Richtlinie für den Zugriff auf einen S3-Bucket, eine benutzerdefinierte S3-Bucket-Richtlinie erstellt haben, suchen Sie danach und aktivieren Sie das Kontrollkästchen neben ihrem Namen.
   - Wenn Sie beabsichtigen, einem Active Directory, das von verwaltet wird Directory Service, keine EC2 Maschinen hinzuzufügen, suchen Sie nach Amazon SSMDirectory ServiceAccess und aktivieren Sie das Kontrollkästchen neben dem Namen.

- Wenn Sie planen, Ihren verwalteten Knoten mithilfe von EventBridge oder CloudWatch Logs zu verwalten oder zu überwachen, suchen Sie nach dem entsprechenden Knoten CloudWatchAgentServerPolicyund aktivieren Sie das Kontrollkästchen neben dem Namen.

6. Wählen Sie Weiter aus.
7. Geben Sie unter Rollenname einen Namen für Ihre neue IAM-Serverrolle ein, z. B. **`SSMServerRole`**.

   > **Note**
   >
   > Notieren Sie sich den Rollennamen. Sie wählen diese Rolle, wenn Sie neue Maschinen registrieren, die Sie mit Systems Manager verwalten möchten.

8. (Optional) Aktualisieren Sie für Beschreibung die Beschreibung für diese IAM-Serverrolle.
9. (Optional) Fügen Sie für Tags ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Rolle zu organisieren, nachzuverfolgen oder zu steuern.
10. Wählen Sie Create role (Rolle erstellen) aus. Das System leitet Sie zur Seite Rollen zurück.

Verwenden der AWS CLI , um eine IAM-Dienstrolle für Systems Manager Manager-Hybridaktivierungen zu erstellen

Verwenden Sie das folgende Verfahren zum Erstellen einer Servicerolle für eine Hybrid-Aktivierung. Dieses Verfahren verwendet die `AmazonSSMManagedInstanceCore`-Richtlinie für die Kernfunktionalität von Systems Manager. Abhängig von Ihrem Anwendungsfall müssen Sie Ihrer Servicerolle möglicherweise zusätzliche Richtlinien für Ihre EC2 Nicht-Computer in einer Hybrid- und Multi-Cloud-Umgebung hinzufügen, um auf andere Tools zugreifen zu können oder. AWS-Services

S3-Bucket-Richtlinienanforderung

Wenn einer der folgenden Fälle zutrifft, müssen Sie eine benutzerdefinierte IAM-Berechtigungsrichtlinie für Amazon Simple Storage Service (Amazon S3)-Buckets erstellen, bevor Sie dieses Verfahren durchführen:

- Fall 1 — Sie verwenden einen VPC-Endpunkt, um Ihre VPC privat mit unterstützten AWS-Services und VPC-Endpunktdiensten zu verbinden, die von unterstützt werden. AWS PrivateLink
- Fall 2: Sie beabsichtigen, einen Amazon-äS3-Bucket zu verwenden, den Sie als Teil Ihrer Systems-Manager-Operationen erstellen, z. B. zum Speichern der Ausgabe für Run Command-Befehle

oder Session Manager-Sitzungen in einem S3-Bucket. Bevor Sie fortfahren, befolgen Sie die Schritte unter Eine benutzerdefinierte S3-Bucket-Richtlinie für ein Instance-Profil erstellen. Die Informationen über S3-Bucket-Richtlinien in diesem Thema gelten auch für Ihre Service-Rolle.

AWS CLI

So erstellen Sie eine IAM-Servicerolle für eine Hybrid- und Multi-Cloud-Umgebung (AWS CLI)

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Erstellen Sie eine Textdatei mit einem Namen wie z. B. `SSMService-Trust.json` mit der folgenden Vertrauensrichtlinie auf Ihrer lokalen Maschine. Stellen Sie sicher, dass Sie die Datei mit der Erweiterung `.json` speichern. Stellen Sie sicher, dass Sie Ihre AWS-Konto und die AWS-Region in der ARN angeben, in der Sie Ihre Hybrid-Aktivierung erstellt haben. Ersetzen Sie *`placeholder values`* die Konto-ID und die Region durch Ihre eigenen Informationen.

   JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"",
         "Effect":"Allow",
         "Principal":{
            "Service":"ssm.amazonaws.com"
         },
         "Action":"sts:AssumeRole",
         "Condition":{
            "StringEquals":{
               "aws:SourceAccount":"123456789012"
            },
            "ArnEquals":{
               "aws:SourceArn":"arn:aws:ssm:us-east-1:111122223333:*"
            }
         }
```

```
        }
    ]
}
```

3. Öffnen Sie die und führen Sie in dem Verzeichnis AWS CLI, in dem Sie die JSON-Datei erstellt haben, den Befehl create-role aus, um die Servicerolle zu erstellen. In diesem Beispiel wird eine Rolle mit dem Namen `SSMServiceRole` erstellt. Sie können auf Wunsch einen anderen Namen wählen.

   Linux & macOS

   ```
   aws iam create-role \
       --role-name SSMServiceRole \
       --assume-role-policy-document file://SSMService-Trust.json
   ```

   Windows

   ```
   aws iam create-role ^
       --role-name SSMServiceRole ^
       --assume-role-policy-document file://SSMService-Trust.json
   ```

4. Führen Sie den attach-role-policyBefehl wie folgt aus, damit die Servicerolle, die Sie gerade erstellt haben, ein Sitzungstoken erstellen kann. Das Sitzungstoken erteilt Ihrem verwalteten Knoten die Berechtigung, Befehle mit Systems Manager auszuführen.

   > **Note**
   >
   > Die Richtlinien, die Sie für ein Serviceprofil für verwaltete Knoten in einer Hybrid- und Multi-Cloud-Umgebung hinzufügen, sind dieselben Richtlinien, die zum Erstellen eines Instance-Profils für Amazon Elastic Compute Cloud (Amazon EC2) -Instances verwendet werden. Weitere Informationen zu den AWS -Richtlinien finden Sie unter Erforderliche Instance-Berechtigungen für Systems Manager konfigurieren.

   (Erforderlich) Führen Sie den folgenden Befehl aus, damit ein verwalteter Knoten die Kernfunktionen des AWS Systems Manager Service nutzen kann.

   Linux & macOS

   ```
   aws iam attach-role-policy \
   ```

```
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
```

Windows

```
aws iam attach-role-policy ^
    --role-name SSMServiceRole ^
    --policy-arn arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
```

Wenn Sie eine benutzerdefinierte S3-Bucket-Richtlinie für Ihre Servicerolle erstellt haben, führen Sie den folgenden Befehl aus, damit AWS Systems Manager Agent (SSM Agent) auf die Buckets zugreifen kann, die Sie in der Richtlinie angegeben haben. Ersetzen Sie *account-id* und *amzn-s3-demo-bucket* durch Ihre AWS-Konto ID und Ihren Bucket-Namen.

Linux & macOS

```
aws iam attach-role-policy \
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::account-id:policy/amzn-s3-demo-bucket
```

Windows

```
aws iam attach-role-policy ^
    --role-name SSMServiceRole ^
    --policy-arn arn:aws:iam::account-id:policy/amzn-s3-demo-bucket
```

(Optional) Führen Sie den folgenden Befehl aus, SSM Agent um in Ihrem Namen Directory Service auf Anfragen zum Beitritt zur Domäne durch den verwalteten Knoten zugreifen zu können. Ihre Servicerolle benötigt diese Richtlinie nur, wenn Sie Ihre Knoten mit einem Microsoft-AD-Verzeichnis verbinden.

Linux & macOS

```
aws iam attach-role-policy \
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::aws:policy/AmazonSSMDirectoryServiceAccess
```

Windows

```
aws iam attach-role-policy ^
    --role-name SSMServiceRole ^
    --policy-arn arn:aws:iam::aws:policy/AmazonSSMDirectoryServiceAccess
```

(Optional) Führen Sie den folgenden Befehl aus, damit der CloudWatch Agent auf Ihren verwalteten Knoten ausgeführt werden kann. Dieser Befehl ermöglicht es, Informationen auf einem Knoten zu lesen und darauf zu schreiben CloudWatch. Ihr Serviceprofil benötigt diese Richtlinie nur, wenn Sie Dienste wie Amazon EventBridge oder Amazon CloudWatch Logs verwenden.

```
aws iam attach-role-policy \
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

Tools for PowerShell

So erstellen Sie eine IAM-Servicerolle für eine Hybrid- und Multi-Cloud-Umgebung (AWS Tools for Windows PowerShell)

1. Installieren und konfigurieren Sie die AWS -Tools für PowerShell (Tools für Windows PowerShell), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren des AWS -Tools für PowerShell.

2. Erstellen Sie eine Textdatei mit einem Namen wie z. B. `SSMService-Trust.json` mit der folgenden Vertrauensrichtlinie auf Ihrer lokalen Maschine. Stellen Sie sicher, dass Sie die Datei mit der Erweiterung `.json` speichern. Stellen Sie sicher, dass Sie Ihre AWS-Konto und die AWS-Region in der ARN angeben, in der Sie Ihre Hybrid-Aktivierung erstellt haben.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
```

```
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:ssm:us-
east-1:123456789012:*"
                }
            }
        }
    ]
}
```

3. Öffnen Sie PowerShell im Administratormodus und führen Sie in dem Verzeichnis, in dem Sie die JSON-Datei erstellt haben, New- IAMRole wie folgt aus, um eine Servicerolle zu erstellen. In diesem Beispiel wird eine Rolle mit dem Namen `SSMServiceRole` erstellt. Sie können auf Wunsch einen anderen Namen wählen.

```
New-IAMRole `
    -RoleName SSMServiceRole `
    -AssumeRolePolicyDocument (Get-Content -raw SSMService-Trust.json)
```

4. Verwenden Sie Register — IAMRole Policy wie folgt, damit die von Ihnen erstellte Servicerolle ein Sitzungstoken erstellen kann. Das Sitzungstoken erteilt Ihrem verwalteten Knoten die Berechtigung, Befehle mit Systems Manager auszuführen.

> **Note**
>
> Die Richtlinien, die Sie für ein Dienstprofil für verwaltete Knoten in einer Hybrid- und Multi-Cloud-Umgebung hinzufügen, sind dieselben Richtlinien, die zum Erstellen eines Instanzprofils für EC2 Instances verwendet werden. Weitere Informationen zu den in den folgenden Befehlen verwendeten AWS Richtlinien finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

(Erforderlich) Führen Sie den folgenden Befehl aus, damit ein verwalteter Knoten die Kernfunktionen des AWS Systems Manager Service nutzen kann.

```
Register-IAMRolePolicy `
    -RoleName SSMServiceRole `
    -PolicyArn arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
```

Wenn Sie eine benutzerdefinierte S3-Bucket-Richtlinie für Ihre Servicerolle erstellt haben, führen Sie den folgenden Befehl aus, um SSM Agent den Zugriff auf die Buckets zu ermöglichen, die Sie in der Richtlinie angegeben haben. Ersetzen Sie *account-id* und *my-bucket-policy-name* durch Ihre AWS-Konto ID und Ihren Bucket-Namen.

```
Register-IAMRolePolicy `
    -RoleName SSMServiceRole `
    -PolicyArn arn:aws:iam::account-id:policy/my-bucket-policy-name
```

(Optional) Führen Sie den folgenden Befehl aus, SSM Agent um in Ihrem Namen Directory Service auf Anfragen zum Beitritt zur Domäne durch den verwalteten Knoten zugreifen zu können. Ihre Servicerolle benötigt diese Richtlinie nur, wenn Sie Ihre Knoten mit einem Microsoft-AD-Verzeichnis verbinden.

```
Register-IAMRolePolicy `
    -RoleName SSMServiceRole `
    -PolicyArn arn:aws:iam::aws:policy/AmazonSSMDirectoryServiceAccess
```

(Optional) Führen Sie den folgenden Befehl aus, damit der CloudWatch Agent auf Ihren verwalteten Knoten ausgeführt werden kann. Dieser Befehl ermöglicht es, Informationen auf einem Knoten zu lesen und darauf zu schreiben CloudWatch. Ihr Serviceprofil benötigt diese Richtlinie nur, wenn Sie Dienste wie Amazon EventBridge oder Amazon CloudWatch Logs verwenden.

```
Register-IAMRolePolicy `
    -RoleName SSMServiceRole `
    -PolicyArn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

Fahren Sie fort mit Eine Hybridaktivierung erstellen, um Knoten bei Systems Manager zu registrieren.

Eine Hybridaktivierung erstellen, um Knoten bei Systems Manager zu registrieren

Um andere Maschinen als Amazon Elastic Compute Cloud (EC2) -Instances als verwaltete Knoten für eine Hybrid- und Multi-Cloud-Umgebung einzurichten, müssen Sie eine Hybrid-Aktivierung erstellen und anwenden. Nachdem Sie die Aktivierung erfolgreich abschließen, erhalten Sie sofort einen Aktivierungscode und eine Aktivierungs-ID oben auf der Konsolenseite. Sie geben diese Kombination aus Code und ID bei der Installation AWS Systems Manager SSM Agent auf EC2 Computern für Ihre Hybrid- und Multi-Cloud-Umgebung an. Der Code und die ID bieten einen sicheren Zugriff auf den Systems-Manager-Service von Ihren verwalteten Knoten aus.

> **Important**
>
> Systems Manager übergibt den Aktivierungscode und die ID sofort an die Konsole oder das Befehlsfenster, je nachdem, wie Sie die Aktivierung erstellt haben. Kopieren Sie diese Informationen und speichern Sie sie an einem sicheren Ort. Wenn Sie die Konsole verlassen oder das Befehlsfenster schließen, können diese Informationen verloren gehen. Wenn Sie die Informationen verlieren, müssen Sie eine neue Aktivierung erstellen.

Informationen zu den Aktivierungabläufen

Ein Aktivierungsablauf ist ein Zeitfenster, in dem Sie On-Premises-Maschinen mit Systems Manager registrieren können. Eine abgelaufene Aktivierung hat keine Auswirkungen auf Ihre Server oder VMs darauf, dass Sie sich zuvor bei Systems Manager registriert haben. Wenn eine Aktivierung abläuft, können Sie VMs mit dieser speziellen Aktivierung keine weiteren Server oder bei Systems Manager registrieren. Sie müssen eine neue erstellen.

Jeder On-Premises-Server und jede VM, die Sie zuvor registriert haben, bleibt als verwalteter Systems-Manager-Knoten registriert, bis Sie die Registrierung explizit abmelden. Sie können einen nicht EC2 verwalteten Knoten auf folgende Weise deregistrieren:

- Über die Registerkarte Verwaltete Knoten in Fleet Manager in der Systems-Manager-Konsole
- Verwenden Sie den Befehl AWS CLI deregister-managed-instance
- Mithilfe der API-Aktion DeregisterManagedInstance

Weitere Informationen finden Sie in den folgenden Themen:

- Abmelden und Neuregistrieren eines verwalteten Knotens (Linux)
- Abmelden und Neuregistrieren eines verwalteten Knotens (Windows Server)

Informationen zu verwalteten Knoten

Ein verwalteter Knoten ist ein beliebiger Computer, für den er konfiguriert ist AWS Systems Manager. AWS Systems Manager unterstützt Amazon Elastic Compute Cloud (Amazon EC2) -Instances, Edge-Geräte und lokale Server oder VMs auch VMs in anderen Cloud-Umgebungen. Bisher wurden alle verwalteten Knoten als verwaltete Instances bezeichnet. Der Begriff Instance bezieht sich jetzt nur noch auf EC2 Instances. Der deregister-managed-instanceBefehl wurde vor dieser Terminologieänderung benannt.

Informationen zu Aktivierungs-Tags

Wenn Sie eine Aktivierung entweder mit AWS Command Line Interface (AWS CLI) oder erstellen AWS Tools for Windows PowerShell, können Sie Tags angeben. Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Hier ist ein AWS CLI Beispielbefehl, der in der Region USA Ost (Ohio) auf einem lokalen Linux-Computer ausgeführt werden kann und optionale Tags enthält.

```
aws ssm create-activation \
  --default-instance-name MyWebServers \
  --description "Activation for Finance department webservers" \
  --iam-role service-role/AmazonEC2RunCommandRoleForManagedInstances \
  --registration-limit 10 \
  --region us-east-2 \
  --tags "Key=Department,Value=Finance"
```

Wenn Sie beim Erstellen einer Aktivierung Tags angeben, werden diese Tags automatisch Ihren verwalteten Knoten zugewiesen, wenn Sie diese aktivieren.

Es ist nicht möglich, Tags zu einer vorhandenen Aktivierung hinzuzufügen oder daraus zu löschen. Wenn Sie Ihren lokalen Servern nicht automatisch und VMs mithilfe einer Aktivierung Tags zuweisen möchten, können Sie ihnen später Tags hinzufügen. Insbesondere können Sie Ihre lokalen Server taggen und VMs nachdem sie sich zum ersten Mal mit Systems Manager verbunden haben. Nachdem diese eine Verbindung hergestellt haben, wird ihnen eine verwaltete Knoten-ID zugewiesen und sie werden in der Systems-Manager-Konsole mit einer ID mit dem Präfix „mi-“ aufgeführt.

> **Note**
>
> Sie können einer Aktivierung keine Tags zuweisen, wenn Sie sie mithilfe der Systems Manager-Konsole erstellen. Sie müssen es entweder mit den Tools AWS CLI oder mit den Tools für Windows PowerShell erstellen.

Wenn Sie einen On-Premises-Server oder eine virtuelle Maschine (VM) nicht mehr mithilfe von Systems Manager verwalten möchten, können Sie die Registrierung aufheben. Weitere Informationen finden Sie unter Aufheben der Registrierung von verwalteten Knoten in einer Hybrid- und Multi-Cloud-Umgebung.

Themen

- Verwenden von AWS-Managementkonsole , um eine Aktivierung für die Registrierung verwalteter Knoten bei Systems Manager zu erstellen
- Verwenden Sie die Befehlszeile zum Erstellen einer Aktivierung für die Registrierung verwalteter Knoten bei Systems Manager

Verwenden von AWS-Managementkonsole , um eine Aktivierung für die Registrierung verwalteter Knoten bei Systems Manager zu erstellen

So erstellen Sie eine Aktivierung für einen verwalteten Knoten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Hybrid Activations.
3. Wählen Sie Create activation aus.

   –oder–

   Wenn Sie in der aktuellen Version zum ersten Mal auf Hybrid-Aktivierungen zugreifen AWS-Region, wählen Sie Create an Activations.
4. (Optional) Geben Sie für Aktivierungs-Beschreibung eine Beschreibung für diese Aktivierung ein. Wir empfehlen, eine Beschreibung einzugeben, wenn Sie eine große Anzahl von Servern aktivieren möchten und VMs.

5. Geben Sie unter Instanzlimit die Gesamtzahl der Knoten an, bei denen Sie sich im AWS Rahmen dieser Aktivierung registrieren möchten. Der Standardwert ist 1 Instance.
6. Wählen Sie für die IAM-Rolle eine Servicerollenoption aus, die es Ihren Servern ermöglicht, mit ihnen AWS Systems Manager in der Cloud VMs zu kommunizieren:

   - Option 1: Wählen Sie Verwenden Sie die vom System erstellte Standardrolle, um eine Rolle und verwaltete Richtlinie zu verwenden, die von AWS bereitgestellt wird.
   - Option 2: Wählen Sie Select an existing custom IAM role that has the required permissions (Vorhandene benutzerdefinierte IAM Rolle auswählen) aus, um die optionale benutzerdefinierte Rolle zu verwenden, die Sie zuvor erstellt haben. Diese Rolle muss über eine Vertrauensbeziehungsrichtlinie verfügen, die `"Service": "ssm.amazonaws.com"` angibt. Wenn Ihre IAM-Rolle dieses Prinzip in einer Vertrauensbeziehungsrichtlinie nicht angibt, wird die folgende Fehlermeldung angezeigt:

     ```
     An error occurred (ValidationException) when calling the CreateActivation
                                        operation: Not existing role:
      arn:aws:iam::<accountid>:role/SSMRole
     ```

     Weitere Informationen zum Erstellen dieser Rolle finden Sie unter Erstellen Sie die für Systems Manager in Hybrid- und Multi-Cloud-Umgebungen erforderliche IAM-Servicerolle.
7. Geben Sie im Feld Activation expiry date (Aktivierungsablaufdatum) ein Ablaufdatum für die Aktivierung an. Das Verfallsdatum muss in der Zukunft liegen - jedoch nicht mehr als 30 Tage. Der Standardwert beträgt 24 Stunden.

   > **Note**
   >
   > Wenn Sie nach dem Ablaufdatum weitere verwaltete Knoten registrieren möchten, müssen Sie eine neue Aktivierung erstellen. Das Verfallsdatum wirkt sich nicht auf registrierte und ausgeführte Knoten aus.
8. (Optional) Geben Sie für das Feld Default instance name (Standard-Instance-Name) einen identifizierenden Namenswert an, der für alle verwalteten Knoten angezeigt werden soll, die dieser Aktivierung zugeordnet sind.
9. Wählen Sie Create activation aus. Der Systems Manager gibt den Aktivierungscode und die ID sofort an die Konsole zurück.

Verwenden Sie die Befehlszeile zum Erstellen einer Aktivierung für die Registrierung verwalteter Knoten bei Systems Manager

Das folgende Verfahren beschreibt, wie Sie AWS Command Line Interface (AWS CLI) (unter Linux oderWindows Server) verwenden oder eine verwaltete Knotenaktivierung erstellen. AWS -Tools für PowerShell

So erstellen Sie eine Aktivierung

1. Installieren und konfigurieren Sie das AWS CLI oder das AWS -Tools für PowerShell, falls Sie das noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.

2. Führen Sie den folgenden Befehl aus, um eine Aktivierung zu erstellen.

   > **Note**
   >
   > - Ersetzen Sie den Befehl im folgenden Befehl *`region`* durch Ihre eigenen Informationen. Eine Liste der unterstützten *`region`* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.
   > - Die Rolle, die Sie für den *`iam-role`* Parameter angeben, muss über eine Vertrauensstellungsrichtlinie verfügen, die Folgendes festlegt`"Service": "ssm.amazonaws.com"`. Wenn Ihre AWS Identity and Access Management (IAM-) Rolle dieses Prinzip nicht in einer Vertrauensstellungsrichtlinie spezifiziert, erhalten Sie die folgende Fehlermeldung:
   >
   >   ```
   >   An error occurred (ValidationException) when calling the CreateActivation
   >                                           operation: Not existing role:
   >    arn:aws:iam::<accountid>:role/SSMRole
   >   ```
   >
   >   Weitere Informationen zum Erstellen dieser Rolle finden Sie unter Erstellen Sie die für Systems Manager in Hybrid- und Multi-Cloud-Umgebungen erforderliche IAM-Servicerolle.
   > - Geben Sie für `--expiration-date` ein Datum im Zeitstempel-Format an, z. B. `"2021-07-07T00:00:00"`, wenn der Aktivierungscode abläuft. Sie können ein

Datum bis zu 30 Tage im Voraus angeben. Wenn Sie kein Ablaufdatum angeben, läuft der Aktivierungscode innerhalb von 24 Stunden ab.

Linux & macOS

```
aws ssm create-activation \
    --default-instance-name name \
    --iam-role iam-service-role-name \
    --registration-limit number-of-managed-instances \
    --region region \
    --expiration-date "timestamp" \\
    --tags "Key=key-name-1,Value=key-value-1" "Key=key-name-2,Value=key-value-2"
```

Windows

```
aws ssm create-activation ^
    --default-instance-name name ^
    --iam-role iam-service-role-name ^
    --registration-limit number-of-managed-instances ^
    --region region ^
    --expiration-date "timestamp" ^
    --tags "Key=key-name-1,Value=key-value-1" "Key=key-name-2,Value=key-value-2"
```

PowerShell

```
New-SSMActivation -DefaultInstanceName name `
    -IamRole iam-service-role-name `
    -RegistrationLimit number-of-managed-instances `
    -Region region `
    -ExpirationDate "timestamp" `
    -Tag @{"Key"="key-name-1";"Value"="key-value-1"},@{"Key"="key-
name-2";"Value"="key-value-2"}
```

Ein Beispiel.

Linux & macOS

```
aws ssm create-activation \
    --default-instance-name MyWebServers \
```

```
    --iam-role service-role/AmazonEC2RunCommandRoleForManagedInstances \
    --registration-limit 10 \
    --region us-east-2 \
    --expiration-date "2021-07-07T00:00:00" \
    --tags "Key=Environment,Value=Production" "Key=Department,Value=Finance"
```

Windows

```
aws ssm create-activation ^
    --default-instance-name MyWebServers ^
    --iam-role service-role/AmazonEC2RunCommandRoleForManagedInstances ^
    --registration-limit 10 ^
    --region us-east-2 ^
    --expiration-date "2021-07-07T00:00:00" ^
    --tags "Key=Environment,Value=Production" "Key=Department,Value=Finance"
```

PowerShell

```
New-SSMActivation -DefaultInstanceName MyWebServers `
    -IamRole service-role/AmazonEC2RunCommandRoleForManagedInstances `
    -RegistrationLimit 10 `
    -Region us-east-2 `
    -ExpirationDate "2021-07-07T00:00:00" `
    -Tag
 @{"Key"="Environment";"Value"="Production"},@{"Key"="Department";"Value"="Finance"}
```

Wenn die Aktivierung erfolgreich erstellt wurde, gibt das System sofort einen Aktivierungscode und eine Aktivierungs-ID zurück.

Installation von SSM Agent auf hybriden Linux-Knoten

In diesem Thema wird beschrieben, wie die Installation AWS Systems Manager SSM Agent auf Linux-Computern, die nicht EC2 (Amazon Elastic Compute Cloud) sind, in einer Hybrid- und Multi-Cloud-Umgebung durchgeführt wird. Informationen zur Installation SSM Agent auf EC2 Instances für Linux finden Sie unterManuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux.

Bevor Sie beginnen, suchen Sie den Aktivierungscode und die Aktivierungs-ID, die Sie während der Hybrid-Aktivierung generiert haben, wie unter Eine Hybridaktivierung erstellen, um Knoten bei

Systems Manager zu registrieren beschrieben. Sie geben den Code und die ID in den folgenden Schritten an.

Zur Installation SSM Agent auf EC2 Nicht-Computern in einer Hybrid- und Multi-Cloud-Umgebung

1. Melden Sie sich bei einem Server oder einer VM in Ihrer Hybrid- und Multi-Cloud-Umgebung an.
2. Wenn Sie einen HTTP- oder HTTPS-Proxy verwenden, müssen Sie die `http_proxy` oder `https_proxy`-Umgebungsvariablen in der aktuellen Shell-Sitzung einstellen. Wenn Sie keinen Proxy verwenden, können Sie diesen Schritt überspringen.

   Geben Sie für einen HTTP-Proxy-Server die folgenden Befehle in der Befehlszeile ein:

   ```
   export http_proxy=http://hostname:port
   export https_proxy=http://hostname:port
   ```

   Geben Sie für einen HTTPS-Proxy-Server die folgenden Befehle in der Befehlszeile ein:

   ```
   export http_proxy=http://hostname:port
   export https_proxy=https://hostname:port
   ```

3. Kopieren Sie einen der folgenden Befehlsblöcke und fügen Sie ihn in SSH ein. Ersetzen Sie die Platzhalterwerte durch den Aktivierungscode und die Aktivierungs-ID, die während des Hybrid-Aktivierungsprozesses generiert wurden, sowie durch die ID der Datei, von der AWS-Region Sie herunterladen SSM Agent möchten, und drücken Sie dann auf. `Enter`

   > **Important**
   >
   > Beachten Sie die folgenden wichtigen Details:
   >
   > - Die Verwendung `ssm-setup-cli` für EC2 Nichtinstallationen maximiert die Sicherheit Ihrer Systems Manager Manager-Installation und -Konfiguration.
   > - `sudo` ist nicht erforderlich, wenn Sie ein Stammbenutzer sind.
   > - Laden Sie es `ssm-setup-cli` von dem Ort herunter AWS-Region , an dem Ihre Hybrid-Aktivierung erstellt wurde.
   > - `ssm-setup-cli` unterstützt eine `manifest-url`-Option, die die Quelle bestimmt, von der der Agent heruntergeladen wird. Geben Sie für diese Option keinen Wert an, es sei denn, Ihre Organisation verlangt dies.

- Verwenden Sie bei der Registrierung von Instances nur den bereitgestellten Download-Link für `ssm-setup-cli`. `ssm-setup-cli`sollte nicht separat für die zukünftige Verwendung aufbewahrt werden.
- Sie können das hier bereitgestellte Skript verwenden, um die Signatur von `ssm-setup-cli` zu überprüfen.

*`region`*steht für die Kennung einer Region, die von AWS-Region unterstützt wird AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *`region`* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

`ssm-setup-cli` enthält zusätzlich die folgenden Optionen:

- `version` – Gültige Werte sind `latest` und `stable`.
- `downgrade` – Erlaubt, dass SSM Agent auf eine ältere Version zurückgesetzt wird. Geben Sie `true` an, um eine frühere Version des Agenten zu installieren.
- `skip-signature-validation` – Überspringt die Signaturvalidierung während des Herunterladens und der Installation des Agenten.

Amazon Linux 2, RHEL 7.x und Oracle Linux

```
mkdir /tmp/ssm
curl https://amazon-ssm-region.s3.region.amazonaws.com/latest/linux_amd64/ssm-setup-cli
 -o /tmp/ssm/ssm-setup-cli
sudo chmod +x /tmp/ssm/ssm-setup-cli
sudo /tmp/ssm/ssm-setup-cli -register -activation-code "activation-code" -activation-id
 "activation-id" -region "region"
```

RHEL 8.x

```
mkdir /tmp/ssm
curl https://amazon-ssm-region.s3.region.amazonaws.com/latest/linux_amd64/ssm-setup-cli
 -o /tmp/ssm/ssm-setup-cli
sudo chmod +x /tmp/ssm/ssm-setup-cli
sudo /tmp/ssm/ssm-setup-cli -register -activation-code "activation-code" -activation-id
 "activation-id" -region "region"
```

Debian Server

```
mkdir /tmp/ssm
curl https://amazon-ssm-region.s3.region.amazonaws.com/latest/debian_amd64/ssm-setup-
cli -o /tmp/ssm/ssm-setup-cli
sudo chmod +x /tmp/ssm/ssm-setup-cli
sudo /tmp/ssm/ssm-setup-cli -register -activation-code "activation-code" -activation-id
 "activation-id" -region "region"
```

Ubuntu Server

- Verwenden von .deb-Paketen

  ```
  mkdir /tmp/ssm
  curl https://amazon-ssm-region.s3.region.amazonaws.com/latest/debian_amd64/ssm-setup-
  cli -o /tmp/ssm/ssm-setup-cli
  sudo chmod +x /tmp/ssm/ssm-setup-cli
  sudo /tmp/ssm/ssm-setup-cli -register -activation-code "activation-code" -activation-
  id "activation-id" -region "region"
  ```

- Verwenden von Snap-Paketen

  Sie müssen keine URL für den Download angeben, da der `snap`-Befehl den Agenten automatisch aus dem Snap App Store unter https://snapcraft.io herunterlädt.

  Unter Ubuntu Server 20.04, 18.04 und 16.04 LTS werden SSM Agent-Installationsdateien, einschließlich Agentenbinär- und Konfigurationsdateien, im folgenden Verzeichnis gespeichert: `/snap/amazon-ssm-agent/current/`. Wenn Sie Änderungen an einer Konfigurationsdatei in diesem Verzeichnis vornehmen, müssen Sie dies Datei aus dem Verzeichnis `/snap` in das Verzeichnis `/etc/amazon/ssm/` kopieren. Protokoll- und Bibliotheksdateien wurden nicht geändert (`/var/lib/amazon/ssm`, `/var/log/amazon/ssm`).

  ```
  sudo snap install amazon-ssm-agent --classic
  sudo systemctl stop snap.amazon-ssm-agent.amazon-ssm-agent.service
  sudo /snap/amazon-ssm-agent/current/amazon-ssm-agent -register -code "activation-
  code" -id "activation-id" -region "region"
  sudo systemctl start snap.amazon-ssm-agent.amazon-ssm-agent.service
  ```

> **Important**
>
> Der Kandidat-Kanal im Snap Store enthält die neueste Version von SSM Agent; nicht den stabilen Kanal. Wenn Sie SSM Agent-Versionsinformationen auf dem Kandidatenkanal nachverfolgen möchten, führen Sie den folgenden Befehl auf Ihren von Ubuntu Server verwalteten 18.04- und 16.04-LTS-64-Bit-Knoten aus.
>
> ```
> sudo snap switch --channel=candidate amazon-ssm-agent
> ```

Der Befehl lädt SSM Agent herunter und installiert es auf der hybrid-aktivierten Maschine in Ihrer Hybrid- und Multi-Cloud-Umgebung. Der Befehl stoppt den SSM Agent und registriert die Maschine beim Systems Manager-Service. Die Maschine ist nun ein verwalteter Knoten. EC2 Amazon-Instances, die für Systems Manager konfiguriert sind, sind ebenfalls verwaltete Knoten. In der Systems Manager Manager-Konsole werden Ihre hybridaktivierten Knoten jedoch mit dem Präfix „mi-“ von EC2 Amazon-Instances unterschieden.

Fahren Sie fort mit Installation von SSM Agent auf hybriden Windows Server-Knoten.

Automatische Drehung des privaten Schlüssels einrichten

Um Ihre Sicherheitslage zu stärken, können Sie AWS Systems Manager Agent (SSM Agent) so konfigurieren, dass der private Schlüssel für Ihre Hybrid- und Multi-Cloud-Umgebung automatisch rotiert wird. Sie können auf dieses Feature zugreifen, indem Sie SSM Agent-Version 3.0.1031.0 oder höher verwenden. Aktivieren Sie dieses Feature wie folgt.

Um SSM Agent zu konfigurieren, um den privaten Schlüssel für eine Hybrid- und Multi-Cloud-Umgebung zu rotieren

1. Navigieren Sie zu `/etc/amazon/ssm/` auf einem Linux-Computer oder zu `C:\Program Files\Amazon\SSM` für einen Windows-Computer.
2. Kopieren Sie den Inhalt von `amazon-ssm-agent.json.template` in eine neue Datei namens `amazon-ssm-agent.json`. Speichern Sie `amazon-ssm-agent.json` in demselben Verzeichnis, in dem sich `amazon-ssm-agent.json.template` befindet.
3. Suchen Sie `Profile`, `KeyAutoRotateDays`. Geben Sie die gewünschte Anzahl der Tage zwischen den automatischen Drehungen des privaten Schlüssels eingeben.
4. Starten Sie SSM Agent neu.

Jedes Mal, wenn Sie die Konfiguration ändern, starten Sie SSM Agent neu.

Sie können andere Features von SSM Agent mit dem gleichen Verfahren personalisieren. Eine up-to-date Liste der verfügbaren Konfigurationseigenschaften und ihrer Standardwerte finden Sie unter Definitionen von Konfigurationseigenschaften.

Abmelden und Neuregistrieren eines verwalteten Knotens (Linux)

Sie können die Registrierung eines hybridaktivierten verwalteten Knotens aufheben, indem Sie den DeregisterManagedInstanceAPI-Vorgang entweder über Tools für Windows AWS CLI oder über Tools for Windows aufrufen. PowerShell Hier sehen Sie ein Beispiel für einen CLI-Befehl:

```
aws ssm deregister-managed-instance --instance-id "mi-1234567890"
```

Um die verbleibenden Registrierungsinformationen für den Agenten zu entfernen, entfernen Sie den `IdentityConsumptionOrder`-Schlüssel aus der `amazon-ssm-agent.json`-Datei. Dann führen Sie je nach Installationstyp einen der folgenden Befehle aus.

Auf Ubuntu Server-Knoten, auf denen SSM Agent mithilfe von Snap-Paketen installiert wurde:

```
sudo /snap/amazon-ssm-agent/current/amazon-ssm-agent -register -clear
```

In allen anderen Linux-Installationen:

```
amazon-ssm-agent -register -clear
```

> **Note**
>
> Sie können einen lokalen Server, ein Edge-Gerät oder eine VM mit demselben Aktivierungscode und derselben ID erneut registrieren, solange Sie das Instance-Limit für den angegebenen Aktivierungscode und die angegebene ID nicht erreicht haben. Sie können das Instance-Limit für einen Aktivierungscode und eine Aktivierungs-ID überprüfen, indem Sie in der AWS CLI die describe-activations API aufrufen. Nachdem Sie den Befehl ausgeführt haben, stellen Sie sicher, dass der Wert für `RegistrationCount` nicht über `RegistrationLimit` liegt. Wenn der Wert über dem Limit liegt, müssen Sie einen anderen Aktivierungscode und eine andere Aktivierungs-ID verwenden.

Um einen verwalteten Knoten auf einem Computer, der nicht zu Linux gehört, erneut zu registrieren EC2

1. Verbinden Sie sich mit Ihrer Maschine.
2. Führen Sie den folgenden Befehl aus. Stellen Sie sicher, dass Sie die Platzhalterwerte durch den Aktivierungscode und die Aktivierungs-ID ersetzen, die generiert wurden, als Sie eine Aktivierung für einen verwalteten Knoten erstellt haben, sowie durch die Kennung der Region, aus der Sie den SSM Agent herunterladen möchten.

```
echo "yes" | sudo /tmp/ssm/ssm-setup-cli -register -activation-code "activation-code" -activation-id "activation-id" -region "region
```

Problembehandlung bei der SSM Agent Installation auf Nicht-Linux-Computern EC2

Nutzen Sie die folgenden Informationen, um Probleme bei der Installation von SSM Agent auf hybrid-aktivierten Linux-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung zu beheben.

Sie erhalten eine DeliveryTimedOut Fehlermeldung

Problem: Wenn Sie eine Maschine in einer Maschine AWS-Konto als verwalteten Knoten für eine separate Maschine konfigurieren AWS-Konto, erhalten Sie `DeliveryTimedOut` nach der Ausführung die Befehle zur Installation SSM Agent auf dem Zielcomputer.

Lösung: `DeliveryTimedOut` ist der erwartete Antwortcode für dieses Szenario. Der Befehl zum Installieren von SSM Agent auf dem Zielknoten ändert die Knoten-ID des Quellknotens. Da sich die Knoten-ID geändert hat, kann der Quellknoten dem Zielknoten nicht antworten, dass der Befehl bei der Ausführung fehlgeschlagen, abgeschlossen oder abgelaufen ist.

Knotenzuordnungen können nicht geladen werden

Problem: Nach dem Ausführen der Installationsbefehle wird der folgende Fehler im SSM Agent in den Fehlerprotokollen angezeigt:

```
Unable to load instance associations, unable to retrieve
associations unable to retrieve associations error occurred in
RequestManagedInstanceRoleToken: MachineFingerprintDoesNotMatch:
Fingerprint doesn't match
```

Sie sehen diesen Fehler, wenn die Computer-ID bei einem Neustart nicht beibehalten wird.

Lösung: Um dieses Problem zu lösen, führen Sie den folgenden Befehl aus. Dieser Befehl zwingt, dass die Computer-ID bei einem Neustart beibehalten wird.

```
umount /etc/machine-id
systemd-machine-id-setup
```

Installation von SSM Agent auf hybriden Windows Server-Knoten

In diesem Thema wird beschrieben, wie die Installation AWS Systems Manager SSM Agent auf Windows Server Maschinen in einer Hybrid- und Multicloud-Umgebung durchgeführt wird. Informationen zur Installation SSM Agent auf EC2 Instanzen für finden Sie Windows Server unterManuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Windows Server.

Bevor Sie beginnen, suchen Sie den Aktivierungscode und die Aktivierungs-ID, die Sie während der Hybrid-Aktivierung generiert haben, wie unter Eine Hybridaktivierung erstellen, um Knoten bei Systems Manager zu registrieren beschrieben. Sie geben den Code und die ID in den folgenden Schritten an.

Zur Installation SSM Agent auf EC2 Windows Server Nicht-Computern in einer Hybrid- und Multi-Cloud-Umgebung

1. Melden Sie sich bei einem Server oder einer VM in Ihrer Hybrid- und Multi-Cloud-Umgebung an.
2. Wenn Sie einen HTTP- oder HTTPS-Proxy verwenden, müssen Sie die `http_proxy` oder `https_proxy`-Umgebungsvariablen in der aktuellen Shell-Sitzung einstellen. Wenn Sie keinen Proxy verwenden, können Sie diesen Schritt überspringen.

   Legen Sie für einen HTTP-Proxyserver folgende Variable fest:

   ```
   http_proxy=http://hostname:port
   https_proxy=http://hostname:port
   ```

   Legen Sie für einen HTTPS-Proxyserver folgende Variable fest:

   ```
   http_proxy=http://hostname:port
   https_proxy=https://hostname:port
   ```

   Konfigurieren Sie für PowerShell die INet Win-Proxyeinstellungen:

   ```
   [System.Net.WebRequest]::DefaultWebProxy
   ```

```
$proxyServer = "http://hostname:port"
$proxyBypass = "169.254.169.254"
$WebProxy = New-Object System.Net.WebProxy($proxyServer,$true,$proxyBypass)

[System.Net.WebRequest]::DefaultWebProxy = $WebProxy
```

> **Note**
>
> Für den PowerShell Betrieb ist eine INet Win-Proxykonfiguration erforderlich. Weitere Informationen finden Sie unter the section called “SSM Agent-Proxy-Einstellungen und Systems Manager-Services”.

3. Öffnen Sie Windows PowerShell im erweiterten (Verwaltungs-)Modus.
4. Kopieren Sie den folgenden Befehlsblock in Windows PowerShell. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen. Zum Beispiel der Aktivierungscode und die Aktivierungs-ID, die generiert wurden, wenn Sie eine Hybrid-Aktivierung erstellen, zusammen mit der ID der Datei, SSM Agent von der AWS-Region Sie herunterladen möchten.

> **Important**
>
> Beachten Sie die folgenden wichtigen Details:
>
> - Die Verwendung `ssm-setup-cli` für EC2 Nichtinstallationen maximiert die Sicherheit Ihrer Systems Manager Manager-Installation und -Konfiguration.
> - `ssm-setup-cli` unterstützt eine `manifest-url`-Option, die die Quelle bestimmt, von der der Agent heruntergeladen wird. Geben Sie für diese Option keinen Wert an, es sei denn, Ihre Organisation verlangt dies.
> - Sie können das hier bereitgestellte Skript verwenden, um die Signatur von `ssm-setup-cli` zu überprüfen.
> - Verwenden Sie bei der Registrierung von Instances nur den bereitgestellten Download-Link für `ssm-setup-cli`. `ssm-setup-cli`sollte nicht separat für die zukünftige Verwendung aufbewahrt werden.

*region*steht für den Bezeichner einer Region AWS Systems Manager, die von AWS-Region unterstützt wird, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

`ssm-setup-cli` enthält zusätzlich die folgenden Optionen:

- `version` – Gültige Werte sind `latest` und `stable`.
- `downgrade` – Setzt den Agenten auf eine frühere Version zurück.
- `skip-signature-validation` – Überspringt die Signaturvalidierung während des Herunterladens und der Installation des Agenten.

64-bit

```
[System.Net.ServicePointManager]::SecurityProtocol = 'TLS12'
$code = "activation-code"
$id = "activation-id"
$region = "us-east-1"
$dir = $env:TEMP + "\ssm"
New-Item -ItemType directory -Path $dir -Force
cd $dir
(New-Object System.Net.WebClient).DownloadFile("https://amazon-ssm-$region.s3.
$region.amazonaws.com/latest/windows_amd64/ssm-setup-cli.exe", $dir + "\ssm-
setup-cli.exe")
./ssm-setup-cli.exe -register -activation-code="$code" -activation-id="$id" -
region="$region"
Get-Content ($env:ProgramData + "\Amazon\SSM\InstanceData\registration")
Get-Service -Name "AmazonSSMAgent"
```

5. Drücken Sie `Enter`.

> **Note**
>
> Wenn der Befehl fehlschlägt, überprüfen Sie, ob Sie die neueste Version von AWS -Tools für PowerShell verwenden.

Der Befehl hat folgende Auswirkungen:

- Lädt SSM Agent herunter und installiert es auf dem Computer.
- Registriert den Computer beim Systems Manager Service.
- Gibt eine Antwort auf die Anfrage zurück, die der folgenden ähnlich ist:

```
    Directory: C:\Users\ADMINI~1\AppData\Local\Temp\2


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----       07/07/2018   8:07 PM                ssm
{"ManagedInstanceID":"mi-008d36be46EXAMPLE","Region":"us-east-2"}

Status      : Running
Name        : AmazonSSMAgent
DisplayName : Amazon SSM Agent
```

Die Maschine ist nun ein verwalteter Knoten. Diese verwalteten Knoten werden jetzt mit dem Präfix „mi-“ gekennzeichnet. Sie können verwaltete Knoten auf der Seite Verwaltete Knoten in anzeigenFleet Manager, indem Sie den AWS CLI Befehl describe-instance-informationoder den API-Befehl DescribeInstanceInformationverwenden.

Automatische Drehung des privaten Schlüssels einrichten

Um Ihre Sicherheitslage zu stärken, können Sie AWS Systems Manager Agent (SSM Agent) so konfigurieren, dass der private Schlüssel für eine Hybrid- und Multi-Cloud-Umgebung automatisch rotiert wird. Sie können auf dieses Feature zugreifen, indem Sie SSM Agent-Version 3.0.1031.0 oder höher verwenden. Aktivieren Sie dieses Feature wie folgt.

Um SSM Agent zu konfigurieren, um den privaten Schlüssel für eine Hybrid- und Multi-Cloud-Umgebung zu rotieren

1. Navigieren Sie zu `/etc/amazon/ssm/` auf einem Linux-Computer oder zu `C:\Program Files\Amazon\SSM` für einen Windows Server-Computer.
2. Kopieren Sie den Inhalt von `amazon-ssm-agent.json.template` in eine neue Datei namens `amazon-ssm-agent.json`. Speichern Sie `amazon-ssm-agent.json` in demselben Verzeichnis, in dem sich `amazon-ssm-agent.json.template` befindet.

3. Suchen Sie `Profile`, `KeyAutoRotateDays`. Geben Sie die gewünschte Anzahl der Tage zwischen den automatischen Drehungen des privaten Schlüssels eingeben.
4. Starten Sie SSM Agent neu.

Jedes Mal, wenn Sie die Konfiguration ändern, starten Sie SSM Agent neu.

Sie können andere Features von SSM Agent mit dem gleichen Verfahren personalisieren. Eine up-to-date Liste der verfügbaren Konfigurationseigenschaften und ihrer Standardwerte finden Sie unter Definitionen von Konfigurationseigenschaften.

Abmelden und Neuregistrieren eines verwalteten Knotens (Windows Server)

Sie können die Registrierung eines verwalteten Knotens aufheben, indem Sie den DeregisterManagedInstanceAPI-Vorgang entweder in den Tools für Windows AWS CLI oder in den Tools für Windows aufrufen. PowerShell Hier sehen Sie ein Beispiel für einen CLI-Befehl:

```
aws ssm deregister-managed-instance --instance-id "mi-1234567890"
```

Um die verbleibenden Registrierungsinformationen für den Agenten zu entfernen, entfernen Sie den `IdentityConsumptionOrder`-Schlüssel aus der `amazon-ssm-agent.json`-Datei. Führen Sie anschließend den folgenden Befehl aus:

```
amazon-ssm-agent -register -clear
```

> **Note**
>
> Sie können einen lokalen Server, ein Edge-Gerät oder eine VM mit demselben Aktivierungscode und derselben ID erneut registrieren, solange Sie das Instance-Limit für den angegebenen Aktivierungscode und die angegebene ID nicht erreicht haben. Sie können das Instance-Limit für einen Aktivierungscode und eine Aktivierungs-ID überprüfen, indem Sie in der AWS CLI die describe-activations API aufrufen. Nachdem Sie den Befehl ausgeführt haben, stellen Sie sicher, dass der Wert für `RegistrationCount` nicht über `RegistrationLimit` liegt. Wenn der Wert über dem Limit liegt, müssen Sie einen anderen Aktivierungscode und eine andere Aktivierungs-ID verwenden.

So registrieren Sie einen verwalteten Knoten auf einem Windows Server-Hybridcomputer neu

1. Verbinden Sie sich mit Ihrer Maschine.

2. Führen Sie den folgenden Befehl aus. Achten Sie darauf, die Platzhalterwerte durch den Aktivierungscode und die Aktivierungs-ID zu ersetzen, die bei der Erstellung einer Hybrid-Aktivierung generiert wurden, sowie durch die Kennung der Region, aus der Sie den SSM Agent herunterladen möchten.

```
$dir = $env:TEMP + "\ssm"
cd $dir
Start-Process ./ssm-setup-cli.exe -ArgumentList @(
    "-register",
    "-activation-code=$code",
    "-activation-id=$id",
    "-region=$region"
) -Wait -NoNewWindow
```

Verwalten von Edge-Geräten mit Systems Manager

In diesem Abschnitt werden die Einrichtungsaufgaben beschrieben, die Konto- und Systemadministratoren ausführen, um die Konfiguration und Verwaltung von AWS IoT Greengrass Kerngeräten zu ermöglichen. Nachdem Sie diese Aufgaben abgeschlossen haben, können Benutzer, denen der AWS-Konto Administrator Berechtigungen erteilt hat, sie AWS Systems Manager zur Konfiguration und Verwaltung der AWS IoT Greengrass Kerngeräte ihrer Organisation verwenden.

> **Note**
>
> - SSM Agentfür wird unter macOS Windows 10 AWS IoT Greengrass nicht unterstützt. Sie können keine Systems-Manager-Tools verwenden, um Edge-Geräte zu verwalten und zu konfigurieren, die diese Betriebssysteme verwenden.
> - Systems Manager unterstützt auch Edge-Geräte, die nicht als AWS IoT Greengrass Core-Geräte konfiguriert sind. Um Systems Manager zur Verwaltung von AWS IoT Core-Geräten und AWS Nicht-Edge-Geräten zu verwenden, müssen Sie sie mithilfe einer Hybridaktivierung konfigurieren. Weitere Informationen finden Sie unter Verwalten von Knoten in Hybrid- und Multi-Cloud-Umgebungen mit Systems Manager.
> - Um Session Manager und Microsoft-Anwendungs-Patching mit Ihren Edge-Geräten zu verwenden, müssen Sie das Advanced-Instances-Kontingent aktivieren. Weitere Informationen finden Sie unter Aktivieren des Kontingents für erweiterte Instances.

Bevor Sie beginnen

Stellen Sie sicher, dass Ihre Edge-Geräte die folgenden Anforderungen erfüllen.

- Ihre Edge-Geräte müssen die Anforderungen erfüllen, um als AWS IoT Greengrass Core-Geräte konfiguriert zu werden. Weitere Informationen finden Sie unter Einrichten von AWS IoT Greengrass Kerngeräten im AWS IoT Greengrass Version 2 Entwicklerhandbuch.
- Ihre Edge-Geräte müssen mit AWS Systems Manager Agent (SSM Agent) kompatibel sein. Weitere Informationen finden Sie unter Unterstützte Betriebssysteme für Systems Manager.
- Ihre Edge-Geräte müssen mit dem Systems-Manager-Service in der Cloud kommunizieren können. Systems Manager unterstützt keine getrennten Edge-Geräte.

Informationen über das Einrichten von Edge-Geräten

Das Einrichten von AWS IoT Greengrass Geräten für Systems Manager umfasst die folgenden Prozesse.

> **Note**
>
> Informationen zur Deinstallation SSM Agent von einem Edge-Gerät aus finden Sie unter Deinstallieren des AWS Systems Manager Agenten im AWS IoT Greengrass Version 2 Entwicklerhandbuch.

Erstellen einer IAM-Servicerolle für Edge-Geräte

AWS IoT Greengrass Für die Kommunikation mit Kerngeräten ist eine AWS Identity and Access Management (IAM) -Servicerolle erforderlich. AWS Systems Manager Die Rolle gewährt dem Systems Manager Manager-Dienst AssumeRoleVertrauen AWS Security Token Service (AWS STS). Sie müssen die Servicerolle nur einmal für jedes AWS-Konto erstellen. Sie geben diese Rolle für den `RegistrationRole` Parameter an, wenn Sie die SSM Agent Komponente konfigurieren und auf Ihren AWS IoT Greengrass Geräten bereitstellen. Wenn Sie diese Rolle bereits bei der Einrichtung von EC2 Nicht-Knoten für eine Hybrid- und Multicloud-Umgebung erstellt haben, können Sie diesen Schritt überspringen.

> **Note**
>
> Benutzer in Ihrem Unternehmen oder Ihrer Organisation, die Systems Manager auf Ihren Edge-Geräten verwenden, müssen in IAM die Berechtigung für den Aufruf der Systems-Manager-API erhalten.

S3-Bucket-Richtlinienanforderung

Wenn einer der folgenden Fälle zutrifft, müssen Sie eine benutzerdefinierte IAM-Berechtigungsrichtlinie für Amazon Simple Storage Service (Amazon S3)-Buckets erstellen, bevor Sie dieses Verfahren durchführen:

- Fall 1: Sie verwenden einen VPC-Endpunkt, um Ihre VPC privat mit unterstützten AWS-Services und VPC-Endpunktdiensten zu verbinden, die von unterstützt werden. AWS PrivateLink
- Fall 2: Sie beabsichtigen, einen S3-Bucket zu verwenden, den Sie im Rahmen Ihrer Systems Manager-Operationen erstellen, z. B. zum Speichern von Ausgaben für Run Command-Befehle oder Session Manager-Sitzungen in einem S3-Bucket. Bevor Sie fortfahren, befolgen Sie die Schritte unter Eine benutzerdefinierte S3-Bucket-Richtlinie für ein Instance-Profil erstellen. Die Informationen über S3-Bucket-Richtlinien in diesem Thema gelten auch für Ihre Service-Rolle.

  > **Note**
  >
  > Wenn Ihre Geräte durch eine Firewall geschützt sind und Sie planen, Patch Managerzu verwenden, muss die Firewall den Zugriff auf den Patch-Baseline-Endpunkt `arn:aws:s3:::patch-baseline-snapshot-`*`region`*`/*` erlauben. *`region`*steht für die Kennung einer Region, die von AWS-Region unterstützt wird AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *`region`* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

AWS CLI

So erstellen Sie eine IAM-Dienstrolle für eine AWS IoT Greengrass Umgebung ()AWS CLI

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Erstellen Sie eine Textdatei mit einem Namen wie z. B. `SSMService-Trust.json` mit der folgenden Vertrauensrichtlinie auf Ihrer lokalen Maschine. Stellen Sie sicher, dass Sie die Datei mit der Erweiterung `.json` speichern.

> **Note**
>
> Notieren Sie den Namen. Sie geben es an, wenn Sie es SSM Agent auf Ihren AWS IoT Greengrass Kerngeräten bereitstellen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "ssm.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
}
```

3. Öffnen Sie die und führen Sie in dem Verzeichnis AWS CLI, in dem Sie die JSON-Datei erstellt haben, den Befehl create-role aus, um die Servicerolle zu erstellen. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen.

Linux und macOS

```
aws iam create-role \
    --role-name SSMServiceRole \
    --assume-role-policy-document file://SSMService-Trust.json
```

Windows

```
aws iam create-role ^
```

```
    --role-name SSMServiceRole ^
    --assume-role-policy-document file://SSMService-Trust.json
```

4. Führen Sie den attach-role-policyBefehl wie folgt aus, damit die Servicerolle, die Sie gerade erstellt haben, ein Sitzungstoken erstellen kann. Das Sitzungs-Token gewährt Ihren Edge-Geräten die Berechtigung zum Ausführen von Befehlen mit Systems Manager.

> **Note**
>
> Die Richtlinien, die Sie für ein Serviceprofil für Edge-Geräte hinzufügen, sind dieselben Richtlinien, die zum Erstellen eines Instanzprofils für Amazon Elastic Compute Cloud (Amazon EC2) -Instances verwendet wurden. Weitere Informationen zu IAM-Richtlinien finden Sie unter Erforderliche Instance-Berechtigungen für Systems Manager konfigurieren.

(Erforderlich) Führen Sie den folgenden Befehl aus, damit ein Edge-Gerät die Kernfunktionen des AWS Systems Manager Service nutzen kann.

Linux und macOS

```
aws iam attach-role-policy \
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
```

Windows

```
aws iam attach-role-policy ^
    --role-name SSMServiceRole ^
    --policy-arn arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
```

Wenn Sie eine benutzerdefinierte S3-Bucket-Richtlinie für Ihre Servicerolle erstellt haben, führen Sie den folgenden Befehl aus, damit AWS Systems Manager Agent (SSM Agent) auf die Buckets zugreifen kann, die Sie in der Richtlinie angegeben haben. Ersetzen Sie *account_ID* und *my_bucket_policy_name* durch Ihre AWS-Konto ID und Ihren Bucket-Namen.

Linux und macOS

```
aws iam attach-role-policy \
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::account_ID:policy/my_bucket_policy_name
```

Windows

```
aws iam attach-role-policy ^
    --role-name SSMServiceRole ^
    --policy-arn arn:aws:iam::account_id:policy/my_bucket_policy_name
```

(Optional) Führen Sie den folgenden Befehl aus, um SSM Agent in Ihrem Namen den Zugriff auf Directory Service für Anforderungen zum Beitritt zur Domain von Edge-Geräten zu erlauben. Die Servicerolle benötigt diese Richtlinie nur, wenn Sie Ihre Edge-Geräte einem Microsoft-AD-Verzeichnis zuteilen.

Linux und macOS

```
aws iam attach-role-policy \
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::aws:policy/AmazonSSMDirectoryServiceAccess
```

Windows

```
aws iam attach-role-policy ^
    --role-name SSMServiceRole ^
    --policy-arn arn:aws:iam::aws:policy/AmazonSSMDirectoryServiceAccess
```

(Optional) Führen Sie den folgenden Befehl aus, damit der CloudWatch Agent auf Ihren Edge-Geräten ausgeführt werden kann. Dieser Befehl ermöglicht es, Informationen auf einem Gerät zu lesen und darauf zu schreiben CloudWatch. Für Ihre Servicerolle ist diese Richtlinie nur erforderlich, wenn Sie Dienste wie Amazon EventBridge oder Amazon CloudWatch Logs verwenden.

```
aws iam attach-role-policy \
    --role-name SSMServiceRole \
    --policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

Tools for PowerShell

Um eine IAM-Servicerolle für eine AWS IoT Greengrass Umgebung zu erstellen ()AWS Tools for Windows PowerShell

1. Installieren und konfigurieren Sie die AWS -Tools für PowerShell (Tools für Windows PowerShell), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren des AWS -Tools für PowerShell.

2. Erstellen Sie eine Textdatei mit einem Namen wie z. B. `SSMService-Trust.json` mit der folgenden Vertrauensrichtlinie auf Ihrer lokalen Maschine. Stellen Sie sicher, dass Sie die Datei mit der Erweiterung `.json` speichern.

   > **Note**
   >
   > Notieren Sie den Namen. Sie geben es bei der Bereitstellung SSM Agent auf Ihren AWS IoT Greengrass Kerngeräten an.

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": {
           "Effect": "Allow",
           "Principal": {
               "Service": "ssm.amazonaws.com"
           },
           "Action": "sts:AssumeRole"
       }
   }
   ```

3. Öffnen Sie PowerShell im Administratormodus und führen Sie in dem Verzeichnis, in dem Sie die JSON-Datei erstellt haben, New- IAMRole wie folgt aus, um eine Servicerolle zu erstellen.

   ```
   New-IAMRole `
       -RoleName SSMServiceRole `
       -AssumeRolePolicyDocument (Get-Content -raw SSMService-Trust.json)
   ```

4. Verwenden Sie Register — IAMRole Policy wie folgt, damit die von Ihnen erstellte Servicerolle ein Sitzungstoken erstellen kann. Das Sitzungs-Token gewährt Ihren Edge-Geräten die Berechtigung zum Ausführen von Befehlen mit Systems Manager.

> **Note**
>
> Bei den Richtlinien, die Sie für eine Servicerolle für Edge-Geräte in einer AWS IoT Greengrass Umgebung hinzufügen, handelt es sich um dieselben Richtlinien, die zum Erstellen eines Instanzprofils für EC2 Instanzen verwendet werden. Weitere Informationen zu den in den folgenden Befehlen verwendeten AWS Richtlinien finden Sie unter Konfigurieren der für Systems Manager erforderlichen Instanzberechtigungen.

(Erforderlich) Führen Sie den folgenden Befehl aus, damit ein Edge-Gerät die Kernfunktionen des AWS Systems Manager Service nutzen kann.

```
Register-IAMRolePolicy `
    -RoleName SSMServiceRole `
    -PolicyArn arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
```

Wenn Sie eine benutzerdefinierte S3-Bucket-Richtlinie für Ihre Servicerolle erstellt haben, führen Sie den folgenden Befehl aus, um SSM Agent den Zugriff auf die Buckets zu ermöglichen, die Sie in der Richtlinie angegeben haben. Ersetzen Sie *account_ID* und *my_bucket_policy_name* durch Ihre AWS-Konto ID und Ihren Bucket-Namen.

```
Register-IAMRolePolicy `
    -RoleName SSMServiceRole `
    -PolicyArn arn:aws:iam::account_ID:policy/my_bucket_policy_name
```

(Optional) Führen Sie den folgenden Befehl aus, um SSM Agent in Ihrem Namen den Zugriff auf Directory Service für Anforderungen zum Beitritt zur Domain von Edge-Geräten zu erlauben. Die Servicerolle benötigt diese Richtlinie nur, wenn Sie Ihre Edge-Geräte einem Microsoft-AD-Verzeichnis zuteilen.

```
Register-IAMRolePolicy `
    -RoleName SSMServiceRole `
    -PolicyArn arn:aws:iam::aws:policy/AmazonSSMDirectoryServiceAccess
```

(Optional) Führen Sie den folgenden Befehl aus, damit der CloudWatch Agent auf Ihren Edge-Geräten ausgeführt werden kann. Dieser Befehl ermöglicht es, Informationen auf einem Gerät zu lesen und darauf zu schreiben CloudWatch. Für Ihre Servicerolle ist diese Richtlinie nur erforderlich, wenn Sie Dienste wie Amazon EventBridge oder Amazon CloudWatch Logs verwenden.

```
Register-IAMRolePolicy `
    -RoleName SSMServiceRole `
    -PolicyArn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

Konfigurieren Sie Ihre Edge-Geräte für AWS IoT Greengrass

Richten Sie Ihre Edge-Geräte als AWS IoT Greengrass Kerngeräte ein. Der Einrichtungsprozess umfasst die Überprüfung der unterstützten Betriebssysteme und Systemanforderungen sowie die Installation und Konfiguration der AWS IoT Greengrass Core-Software auf Ihren Geräten. Weitere Informationen finden Sie unter Einrichten von AWS IoT Greengrass -Core-Geräten im AWS IoT Greengrass Version 2 -Entwicklerhandbuch.

Aktualisieren Sie die AWS IoT Greengrass Token-Exchange-Rolle und installieren Sie sie SSM Agent auf Ihren Edge-Geräten

Im letzten Schritt zur Einrichtung und Konfiguration Ihrer AWS IoT Greengrass Kerngeräte für Systems Manager müssen Sie die Gerätedienst-Rolle AWS IoT Greengrass AWS Identity and Access Management (IAM), auch Token-Austauschrolle genannt, aktualisieren und AWS Systems Manager Agent (SSM Agent) auf Ihren AWS IoT Greengrass Geräten bereitstellen. Informationen zu diesen Prozessen finden Sie unter Installation des AWS Systems Manager -Agenten im AWS IoT Greengrass Version 2 -Entwicklerhandbuch.

Nach der Bereitstellung SSM Agent auf Ihren Geräten werden Ihre Geräte AWS IoT Greengrass automatisch bei Systems Manager registriert. Es ist keine weitere Registrierung erforderlich. Sie können damit beginnen, die Systems Manager Manager-Tools für den Zugriff, die Verwaltung und Konfiguration Ihrer AWS IoT Greengrass Geräte zu verwenden.

> **Note**
>
> Ihre Edge-Geräte müssen mit dem Systems-Manager-Service in der Cloud kommunizieren können. Systems Manager unterstützt keine getrennten Edge-Geräte.

Einen AWS Organizations delegierten Administrator für Systems Manager erstellen

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestehende Kunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Wenn Sie eine Organisation in einrichten AWS Organizations, weisen Sie ein Verwaltungskonto zu, mit dem alle Verwaltungsaufgaben für alle ausgeführt AWS-Services werden. Der Benutzer des Verwaltungskontos kann nur Systems Manager ein delegiertes Administratorkonto zuweisen, um Verwaltungsaufgaben für Change ManagerExplorer, und OpsCenter auszuführen. AWS Organizations ist ein Kontoverwaltungsdienst, den Sie verwenden können, um eine Organisation zu erstellen und diese Konten zentral AWS-Konten zu verwalten. Weitere Informationen AWS Organizations dazu finden Sie AWS Organizationsim AWS Organizations Benutzerhandbuch.

Change ManagerExplorer, undOpsCenter, Tools in AWS Systems Manager, arbeiten mit, AWS Organizations um Aufgaben für alle Mitgliedskonten Ihrer Organisation auszuführen. Sie können immer nur einen delegierten Administrator für alle Systems-Manager-Tools zuweisen. Das delegierte Administratorkonto muss ein Mitglied der Organisationseinheit sein, der es zugewiesen ist.

Themen

- Verwenden eines delegierten Administrators mit Change Manager
- Verwenden eines delegierten Administrators mit Explorer
- Verwenden eines delegierten Administrators mit OpsCenter
- Verwenden eines delegierten Administrators mit Quick Setup

Verwenden eines delegierten Administrators mit Change Manager

Change Manager ist ein Change-Management-Framework für Unternehmen zum Anfordern, Genehmigen, Implementieren und Melden von Betriebsänderungen an Ihrer Anwendungskonfiguration und Infrastruktur.

Wenn Sie Change Manager unternehmensweit verwenden, weisen Sie ein delegiertes Administratorkonto zu, um Änderungsvorlagen, Genehmigungen und Berichte für alle Mitgliedskonten zu verwalten. Mit Quick Setup können Sie Change Manager für die Verwendung mit einer Organisation einrichten und das delegierte Administratorkonto auswählen. Wenn Sie Change Manager mit einem einzigen Konto arbeiten AWS-Konto, ist das delegierte Administratorkonto nicht erforderlich.

Change Manager zeigt standardmäßig alle Aufgaben im Zusammenhang mit Änderungen im delegierten Administratorkonto an. Anweisungen zur Konfiguration eines delegierten Administrators bei der Einrichtung von Change Manager für eine Organisation finden Sie unter Einrichten von Change Manager für eine Organisation (Management-Konto).

> **Important**
>
> Wenn Sie den Change Manager in einer Organisation verwenden, empfehlen wir, Änderungen immer über das delegierte Administratorkonto vorzunehmen. Obwohl Sie Änderungen von anderen Konten in der Organisation vornehmen, werden diese Änderungen nicht im delegierten Administratorkonto gemeldet oder können nicht angezeigt werden.

Verwenden eines delegierten Administrators mit Explorer

Explorerist ein anpassbares Operations-Dashboard, das eine aggregierte Ansicht der Betriebsdaten (OpsData) für Ihren AWS-Konten Across bietet. AWS-Regionen

Sie können ein delegiertes Administratorkonto für Systems Manager konfigurieren, um Explorer Daten aus mehreren Regionen und Konten mithilfe der Ressourcendatensynchronisierung mit AWS Organizations zu aggregieren. Ein delegierter Administrator kann Explorer Daten mithilfe von, AWS Command Line Interface (AWS CLI) oder suchen AWS-Managementkonsole, filtern und aggregieren. AWS Tools for Windows PowerShell

Wenn Sie ein delegiertes Administratorkonto für Explorer verwenden, begrenzen Sie die Anzahl der Administratoren, die Daten für mehrere Konten und regionale Ressourcen erstellen oder löschen können, auf ein einzelnes AWS-Konto.

Sie können Betriebsdaten AWS-Konten in Ihrer gesamten Organisation synchronisieren, indem Sie Explorer Informationen zum Zuweisen eines delegierten Administrators von Explorer finden Sie unter Konfigurierung eines delegierten Administrators für Explorer.

Verwenden eines delegierten Administrators mit OpsCenter

OpsCenterbietet einen zentralen Ort, an dem Betriebsingenieure und IT-Experten betriebliche Arbeitselemente (OpsItems) im Zusammenhang mit AWS Ressourcen verwalten können. Wenn Sie OpsCenter verwenden möchten, um OpsItems zentral über Konten hinweg zu verwalten, müssen Sie die Organisation in AWS Organizations einrichten.

Mithilfe von Quick Setup für OpsCenter können Sie ein delegiertes Administratorkonto zuweisen und OpsCenter für die zentrale Verwaltung von OpsItems konfigurieren. Weitere Informationen finden Sie unter (Optional) Konfigurieren Sie OpsCenter für die kontenübergreifende Verwaltung von OpsItems mithilfe von Quick Setup.

Verwenden eines delegierten Administrators mit Quick Setup

Quick Setup ist ein Tool in Systems Manager, mit der Sie häufig verwendete AWS -Services und Funktionen mit empfohlenen bewährten Methoden schnell konfigurieren können. Sie können ein delegiertes Administratorkonto konfigurieren, Quick Setup um Sie bei der kontenübergreifenden und regionsübergreifenden Bereitstellung und Verwaltung von AWS Organizations Konfigurationen zu unterstützen. Ein delegierter Administrator für Quick Setup kann Configuration-Manager-Ressourcen in Ihrer Organisation erstellen, aktualisieren, anzeigen und löschen. Systems Manager registriert einen delegierten Administrator für Quick Setup als Teil des Einrichtungsprozesses für die integrierte Konsolenerfahrung. Weitere Informationen finden Sie unter Einrichten der einheitlichen Systems-Manager-Konsole für eine Organisation.

Allgemeine Einrichtung für AWS Systems Manager

Falls Sie dies noch nicht getan haben, registrieren Sie sich für einen AWS-Konto und erstellen Sie einen Administratorbenutzer.

Melden Sie sich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu https://aws.amazon.com/gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS-Managementkonsoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.
2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center Benutzerhandbuch.
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center Benutzerhandbuch.

Einrichten von AWS Systems Manager

In den folgenden Themen wird beschrieben, wie Sie die einheitliche AWS Systems Manager-Konsole für AWS Organizations-Organisationen und einzelne AWS-Konten einrichten.

Themen

- Einrichten des Systems-Manager-Konsolenzugriffs
- Einrichten der einheitlichen Systems-Manager-Konsole für eine Organisation
- Einrichtung einer einheitlichen Systems-Manager-Konsole für ein einziges Konto und eine Region
- Deaktivieren der einheitlichen Systems-Manager-Konsole

Einrichten des Systems-Manager-Konsolenzugriffs

AWS Systems Manager Um den verwenden zu können AWS-Managementkonsole, müssen Sie die richtigen Berechtigungen konfiguriert haben.

Weitere Informationen zum Erstellen von AWS Identity and Access Management Richtlinien und zum Anhängen dieser Richtlinien an IAM-Identitäten finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch

Systems-Manager-Onboarding-Richtlinie

Sie können eine IAM-Richtlinie wie die im folgenden Beispiel gezeigte erstellen und die Richtlinie an Ihre IAM-Identitäten anhängen. Diese Richtlinie gewährt uneingeschränkten Zugriff auf Systems Manager und dessen Konfiguration.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm-quicksetup` – Ermöglicht Prinzipalen Zugriff auf alle AWS Systems Manager Quick Setup -Aktionen.
- `ssm` – Ermöglicht Prinzipalen den Zugriff auf Systems Manager Automation und Resource Explorer.
- `organizations`— Ermöglicht es Prinzipalen, die Struktur einer Organisation zu lesen und delegierte Administratoren zu verwalten AWS Organizations, wenn sie als Organisation in Systems Manager einsteigen.
- `cloudformation` – Ermöglicht Prinzipalen das Verwalten ihrer Quick Setup-Stacks.

- `iam` – Ermöglicht Prinzipalen, IAM-Rollen und -Richtlinien zu verwalten, die für das Onboarding von Systems Manager erforderlich sind.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
     {
       "Sid": "QuickSetupActions",
       "Effect": "Allow",
       "Action": [
         "ssm-quicksetup:*"
       ],
       "Resource": "*"
     },

     {
       "Sid": "SsmReadOnly",
       "Effect": "Allow",
       "Action": [
         "ssm:DescribeAutomationExecutions",
         "ssm:GetAutomationExecution",
         "ssm:ListAssociations",
         "ssm:DescribeAssociation",
         "ssm:ListDocuments",
         "ssm:ListResourceDataSync",
         "ssm:DescribePatchBaselines",
         "ssm:GetPatchBaseline",
         "ssm:DescribeMaintenanceWindows",
         "ssm:DescribeMaintenanceWindowTasks"
       ],
       "Resource": "*"
     },
     {
       "Sid": "SsmDocument",
       "Effect": "Allow",
       "Action": [
         "ssm:GetDocument",
         "ssm:DescribeDocument"
       ],
       "Resource": [
```

```
        "arn:aws:ssm:*:*:document/AWSQuickSetupType-*",
        "arn:aws:ssm:*:*:document/AWS-EnableExplorer"
      ]
    },
    {
      "Sid": "SsmEnableExplorer",
      "Effect": "Allow",
      "Action": "ssm:StartAutomationExecution",
      "Resource": [
         "arn:aws:ssm:*:*:document/AWS-EnableExplorer",
         "arn:aws:ssm:*:*:automation-execution/*"
      ]
    },
    {
      "Sid": "SsmExplorerRds",
      "Effect": "Allow",
      "Action": [
        "ssm:GetOpsSummary",
        "ssm:CreateResourceDataSync",
        "ssm:UpdateResourceDataSync"
      ],
      "Resource": "arn:aws:ssm:*:*:resource-data-sync/AWS-QuickSetup-*"
    },
    {
      "Sid": "OrgsReadOnly",
      "Effect": "Allow",
      "Action": [
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization",
        "organizations:ListDelegatedAdministrators",
        "organizations:ListRoots",
        "organizations:ListParents",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:DescribeOrganizationalUnit",
        "organizations:ListAWSServiceAccessForOrganization"
      ],
      "Resource": "*"
    },
    {
      "Sid": "OrgsAdministration",
      "Effect": "Allow",
      "Action": [
        "organizations:EnableAWSServiceAccess",
        "organizations:RegisterDelegatedAdministrator",
```

```
        "organizations:DeregisterDelegatedAdministrator"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "organizations:ServicePrincipal": [
            "ssm.amazonaws.com",
            "ssm-quicksetup.amazonaws.com",
            "member.org.stacksets.cloudformation.amazonaws.com",
            "resource-explorer-2.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "CfnReadOnly",
      "Effect": "Allow",
      "Action": [
        "cloudformation:ListStacks",
        "cloudformation:DescribeStacks",
        "cloudformation:ListStackSets",
        "cloudformation:DescribeOrganizationsAccess"
      ],
      "Resource": "*"
    },
    {
      "Sid": "OrgCfnAccess",
      "Effect": "Allow",
      "Action": [
        "cloudformation:ActivateOrganizationsAccess"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CfnStackActions",
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation:DeleteStack",
        "cloudformation:DescribeStackResources",
        "cloudformation:DescribeStackEvents",
        "cloudformation:GetTemplate",
        "cloudformation:RollbackStack",
        "cloudformation:TagResource",
```

```
        "cloudformation:UntagResource",
        "cloudformation:UpdateStack"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stack/StackSet-AWS-QuickSetup-*",
        "arn:aws:cloudformation:*:*:stack/AWS-QuickSetup-*",
        "arn:aws:cloudformation:*:*:type/resource/*"
      ]
    },
    {
      "Sid": "CfnStackSetActions",
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStackInstances",
        "cloudformation:CreateStackSet",
        "cloudformation:DeleteStackInstances",
        "cloudformation:DeleteStackSet",
        "cloudformation:DescribeStackInstance",
        "cloudformation:DetectStackSetDrift",
        "cloudformation:ListStackInstanceResourceDrifts",
        "cloudformation:DescribeStackSet",
        "cloudformation:DescribeStackSetOperation",
        "cloudformation:ListStackInstances",
        "cloudformation:ListStackSetOperations",
        "cloudformation:ListStackSetOperationResults",
        "cloudformation:TagResource",
        "cloudformation:UntagResource",
        "cloudformation:UpdateStackSet"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stackset/AWS-QuickSetup-*",
        "arn:aws:cloudformation:*:*:type/resource/*",
        "arn:aws:cloudformation:*:*:stackset-target/AWS-QuickSetup-*:*"
      ]
    },
    {
      "Sid": "ValidationReadonlyActions",
      "Effect": "Allow",
      "Action": [
        "iam:ListRoles",
        "iam:GetRole"
      ],
      "Resource": "*"
    },
```

```
    {
      "Sid": "IamRolesMgmt",
      "Effect": "Allow",
      "Action": [
        "iam:CreateRole",
        "iam:DeleteRole",
        "iam:GetRole",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "iam:GetRolePolicy",
        "iam:ListRolePolicies"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWS-QuickSetup-*",
        "arn:aws:iam::*:role/service-role/AWS-QuickSetup-*"
      ]
    },
    {
      "Sid": "IamPassRole",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWS-QuickSetup-*",
        "arn:aws:iam::*:role/service-role/AWS-QuickSetup-*"
      ],
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "ssm.amazonaws.com",
            "ssm-quicksetup.amazonaws.com",
            "cloudformation.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "IamRolesPoliciesMgmt",
      "Effect": "Allow",
      "Action": [
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy"
      ],
```

```
      "Resource": [
        "arn:aws:iam::*:role/AWS-QuickSetup-*",
        "arn:aws:iam::*:role/service-role/AWS-QuickSetup-*"
      ],
      "Condition": {
        "ArnEquals": {
          "iam:PolicyARN": [
            "arn:aws:iam::aws:policy/
AWSSystemsManagerEnableExplorerExecutionPolicy",
            "arn:aws:iam::aws:policy/AWSQuickSetupSSMDeploymentRolePolicy"
          ]
        }
      }
    },
    {
      "Sid": "CfnStackSetsSLR",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/aws-service-role/
stacksets.cloudformation.amazonaws.com/
AWSServiceRoleForCloudFormationStackSetsOrgAdmin",
        "arn:aws:iam::*:role/aws-service-role/ssm.amazonaws.com/
AWSServiceRoleForAmazonSSM",
        "arn:aws:iam::*:role/aws-service-role/
accountdiscovery.ssm.amazonaws.com/AWSServiceRoleForAmazonSSM_AccountDiscovery",
        "arn:aws:iam::*:role/aws-service-role/ssm-quicksetup.amazonaws.com/
AWSServiceRoleForSSMQuickSetup",
        "arn:aws:iam::*:role/aws-service-role/resource-explorer-2.amazonaws.com/
AWSServiceRoleForResourceExplorer"
      ]
    }
  ]
}
```

AWS Systems Manager Richtlinie für Konsolenbetreiber

Sie können eine IAM-Richtlinie wie die im folgenden Beispiel gezeigte erstellen und die Richtlinie an Ihre IAM-Identitäten anhängen. Diese Richtlinie gewährt vollen Zugriff auf den Betrieb von

Systems Manager und ermöglicht es Systems Manager, Automation-Dokumente zur Diagnose und Problembehebung auszuführen.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm`— Ermöglicht Prinzipalen den Zugriff auf alle Systems Manager APIs.
- `ssm-quicksetup` – Ermöglicht Prinzipalen die Verwaltung ihrer Quick Setup-Konfigurationen.
- `ec2`— Ermöglicht Systems Manager, Ihren aktivierten Status AWS-Regionen und Ihren EC2 Amazon-Instance-Status zu ermitteln.
- `cloudformation` – Ermöglicht Prinzipalen das Lesen ihrer Quick Setup-Stacks.
- `organizations`— Ermöglicht es Prinzipalen, die Struktur einer Organisation zu lesen und delegierte Administratoren zu verwalten AWS Organizations, wenn sie als Organisation in Systems Manager einsteigen.
- `s3` – Ermöglicht Prinzipalen das Auflisten und Abrufen von Objekten in einem Amazon-S3-Bucket zur Diagnose, der während des Onboarding-Prozesses von Systems Manager erstellt wird.
- `iam:PassRole` – Ermöglicht Prinzipalen, Rollen, die sie übernehmen, an Systems Manager zu übergeben, wenn sie Automatisierungen zur Diagnose und Behebung von nicht verwalteten Knoten ausführen.
- `iam:GetRole` – Ermöglicht Prinzipalen, spezifische Rolleninformationen für Quick Setup-Rollen abzurufen, wenn sie in Systems Manager arbeiten.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:*",
        "ssm-quicksetup:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowEC2DescribeActions",
```

```
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstances",
        "ec2:DescribeRegions"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CfnAccess",
      "Effect": "Allow",
      "Action": [
        "cloudformation:ListStacks",
        "cloudformation:ListStackSets",
        "cloudformation:ListStackInstances",
        "cloudformation:ListStackSetOperations",
        "cloudformation:ListStackSetOperationResults",
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackSet",
        "cloudformation:DescribeStackSetOperation",
        "cloudformation:DescribeOrganizationsAccess",
        "cloudformation:DescribeStackInstance",
        "cloudformation:DetectStackSetDrift",
        "cloudformation:ListStackInstanceResourceDrifts"
      ],
      "Resource": "*"
    },
    {
      "Sid": "OrgsReadOnly",
      "Effect": "Allow",
      "Action": [
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization",
        "organizations:ListDelegatedAdministrators",
        "organizations:ListRoots",
        "organizations:ListParents",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:DescribeOrganizationalUnit",
        "organizations:ListAWSServiceAccessForOrganization"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowKMSOperations",
```

```
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "arn:aws:kms:*:*:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/SystemsManagerManaged": "true"
        },
        "ArnLike": {
          "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::do-not-delete-ssm-
diagnosis-*"
        },
        "StringLike": {
          "kms:ViaService": "s3.*.amazonaws.com"
        },
        "Bool": {
          "aws:ViaAWSService": "true"
        }
      }
    },
    {
      "Sid": "AllowReadS3BucketFromOrganization",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::do-not-delete-ssm-diagnosis*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceOrgId": "${aws:PrincipalOrgId}"
        }
      }
    },
    {
      "Sid": "AllowReadS3BucketFromSingleAccount",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::do-not-delete-ssm-diagnosis*",
```

```
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": [
        "arn:aws:iam::*:role/AWS-SSM-DiagnosisAdminRole*",
        "arn:aws:iam::*:role/AWS-SSM-DiagnosisExecutionRole*",
        "arn:aws:iam::*:role/AWS-SSM-RemediationAdminRole*",
        "arn:aws:iam::*:role/AWS-SSM-RemediationExecutionRole*"
      ],
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "ssm.amazonaws.com"
        }
      }
    },
    {
      "Sid": "IamReadOnly",
      "Effect": "Allow",
      "Action": "iam:GetRole",
      "Resource": [
        "arn:aws:iam::*:role/AWS-QuickSetup-*",
        "arn:aws:iam::*:role/service-role/AWS-QuickSetup-*"
      ]
    }
  ]
}
```

AWS Systems Manager Richtlinie „Nur Lesen“ für Konsolenbetreiber

Sie können eine IAM-Richtlinie wie die im folgenden Beispiel gezeigte erstellen und die Richtlinie an Ihre IAM-Identitäten anhängen. Diese Richtlinie gewährt schreibgeschützten Zugriff für die Verwendung von Systems Manager.

- `ssm`— Ermöglicht Prinzipalen den schreibgeschützten APIs Zugriff auf Systems Manager.
- `ssm-quicksetup` – Ermöglicht Prinzipalen das Lesen ihrer Quick Setup-Konfigurationen.

- `cloudformation` – Ermöglicht Prinzipalen das Lesen ihrer Quick Setup-Stacks.
- `iam:GetRole` – Ermöglicht Prinzipalen, spezifische Rolleninformationen für Quick Setup-Rollen abzurufen, wenn sie Systems Manager verwenden.
- `ec2:DescribeRegions` – Ermöglicht Systems Manager,Ihre aktivierten AWS-Regionen zu ermitteln.
- `organizations`— Ermöglicht es Prinzipalen, die Struktur einer Organisation zu lesen AWS Organizations , wenn sie als Organisation in Systems Manager einsteigen.
- `s3` – Ermöglicht Prinzipalen, Objekte n in einem Amazon-S3-Bucket aufzulisten und abzurufen, der während des Systems-Manager-Onboarding-Prozesses erstellt wird.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:Describe*",
        "ssm:Get*",
        "ssm:List*",
        "ssm-quicksetup:List*",
        "ssm-quicksetup:Get*",
        "cloudformation:Describe*",
        "cloudformation:Get*",
        "cloudformation:List*",
        "iam:GetRole",
        "ec2:DescribeRegions",
        "organizations:Describe*",
        "organizations:List*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowKMSOperations",
      "Effect": "Allow",
```

```
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:*:*:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/SystemsManagerManaged": "true"
        },
        "ArnLike": {
          "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::do-not-delete-ssm-
diagnosis-*"
        },
        "StringLike": {
          "kms:ViaService": "s3.*.amazonaws.com"
        },
        "Bool": {
          "aws:ViaAWSService": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::do-not-delete-ssm-diagnosis*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceOrgId": "${aws:PrincipalOrgId}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::do-not-delete-ssm-diagnosis*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
```

```
      }
    }
  ]
}
```

Einrichten der einheitlichen Systems-Manager-Konsole für eine Organisation

Der Einrichtungsvorgang für die einheitliche Systems-Manager-Konsolenerfahrung ist von AWS-Managementkonsole aus mit nur wenigen Klicks abgeschlossen. Um Systems Manager für eine AWS Organizations-Organisation einzurichten, benötigen Sie Zugriff auf das Verwaltungskonto Ihrer Organisation und ein anderes Konto in Ihrer Organisation, das Sie als delegierter Administrator verwenden können. Der Zugriff auf das Verwaltungskonto ist nur erforderlich, um Systems Manager zu aktivieren oder zu deaktivieren. Um Ihre Knoten zu verwalten, verwenden Sie das delegierte Administratorkonto.

Voraussetzungen

Bei der Verwaltung von Knoten in einem Unternehmen verwendet Systems Manager verschiedene abhängige Services, um die Funktionalität der vereinheitlichten Konsole einzurichten und zu erweitern. Daher muss Systems Manager vertrauenswürdigen Zugriff aktivieren und ein delegiertes Administratorkonto für die folgenden Services registrieren:

- AWS CloudFormation – Stellt die für Systems Manager erforderlichen Ressourcen für Ihre Konten bereit.
- AWS Resource Explorer – Suchen und Filtern von EC2-Instances in Ihren Konten.
- AWS Systems Manager Explorer – Überwachung und Problembehebung des Zustands der Ressourcen, die für Systems Manager in Ihren Konten bereitgestellt werden.
- AWS Systems Manager Quick Setup – Stellt für Ihre Konten die für Systems Manager erforderlichen Quick Setup-Konfigurationen bereit.

Bevor beginnen, stellen Sie sicher, dass Sie das Kontingent an delegierten Administratoren für keinen dieser abhängigen Services bereits überschritten haben. Andernfalls können Sie die delegierten Administratorkonten, die zur Aktivierung von Systems Manager erforderlich sind, nicht registrieren. Wenn Sie Systems Manager für eine Organisation aktivieren, ist jedes Konto in Ihrer Organisation enthalten. Derzeit gibt es keine Möglichkeit, Konten vom Einrichtungsprozess auszuschließen.

Wenn Sie Systems Manager aktivieren, können Sie auswählen, welche AWS-Regionen Sie einbeziehen möchten. Es können nur Regionen ausgewählt werden, in denen die vereinheitlichte Systems Manager-Konsole unterstützt wird. Weitere Informationen zu den Regionen, in denen das Konsolenerlebnis verfügbar ist, finden Sie unter Unterstützte AWS-Regionen.

Einheitliche Konsolenressourcen

Der Einrichtungsprozess für die vereinheitlichte Systems Manager-Konsole erledigt viele der erforderlichen Aufgaben für Sie. Abhängig von den Features, die Sie konfigurieren möchten, gehören dazu die Aktivierung der Standard-Host-Management-Konfiguration, um Ihren Knoten die erforderlichen IAM-Berechtigungen bereitzustellen, und vieles mehr. Im Folgenden finden Sie eine detaillierte Liste der Ressourcen, die von Systems Manager für die einheitliche Konsole erstellt wurden. Abhängig von den Features, die Sie konfigurieren möchten, werden einige Ressourcen möglicherweise nicht erstellt.

AWS Resource Explorer-verwaltete Ansichten

- `AWSManagedViewForSSM`: Ermöglicht Systems Manager den Zugriff auf Ressourceninformationen, die vom Ressourcen-Explorer für Ihre Organisation indexiert wurden. Diese verwalteten Ansichten können nur von Systems Manager aktualisiert oder gelöscht werden. Das heißt, wenn Sie die verwalteten Ansichten löschen oder Ressourcen-Explorer deaktivieren möchten, müssen Sie die vereinheitliche Konsole deaktivieren. Weitere Informationen zum Deaktivieren der vereinheitlichten Konsole finden Sie unter Deaktivieren der einheitlichen Systems-Manager-Konsole. Weitere Informationen zu verwalteten Ansichten finden Sie unter AWS Managed Views im Benutzerhandbuch zum Ressourcen-Explorer.

  > **Note**
  >
  > Wenn Sie einen Aggregatorindex für Resource Explorer in einer anderen Region als Ihrer Heimatregion erstellt haben, stuft Systems Manager den aktuellen Index herab. Anschließend bewirbt Systems Manager den lokalen Index in Ihrer Heimatregion als neuen Aggregatorindex. Während dieser Zeit werden nur Knoten für Ihre Heimatregion angezeigt. Dieser Vorgang kann bis zu 24 Stunden dauern.

IAM roles

- `RoleForOnboardingAutomation` – Ermöglicht Systems Manager, Ressourcen während des Einrichtungsvorgangs zu verwalten. Weitere Informationen zu der Richtlinie finden Sie unter AWSQuickSetupSSMManageResourcesExecutionPolicy.
- `RoleForLifecycleManagement` – Ermöglicht Lambda, den Lebenszyklus von Ressourcen zu verwalten, die durch den Einrichtungsprozess erstellt wurden. Weitere Informationen zu der Richtlinie finden Sie unter AWSQuickSetupSSMLifecycleManagementExecutionPolicy.
- `RoleForAutomation` – Eine Servicerolle, die Systems Manager Automation zur Ausführung von Runbooks übernehmen soll. Weitere Informationen finden Sie unter Erstellen Sie die Servicerollen für Automation mithilfe der Konsole.
- `AWSSSMDiagnosisAdminRole` – Eine Administratorrolle, die zum Starten von Automatisierungen verwendet wird, die Diagnose-Runbooks verwenden. Weitere Informationen zu den Richtlinien finden Sie unter AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy, AWS-SSM-Automation-DiagnosisBucketPolicy und AWS-SSM-DiagnosisAutomation-OperationalAccountAdministrationRolePolicy.
- `AWSSSMDiagnosisExecutionRole` – Eine Automatisierungsausführungsrolle für das Diagnose-Runbook. Weitere Informationen zu den Richtlinien finden Sie unter AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy und AWS-SSM-Automation-DiagnosisBucketPolicy.
- `AWSSSMRemediationAdminRole` – Eine Administratorrolle, die zum Starten von Automatisierungen verwendet wird, die Reparatur-Runbooks verwenden. Weitere Informationen zu den Richtlinien finden Sie unter AWS-SSM-RemediationAutomation-AdministrationRolePolicy, AWS-SSM-Automation-DiagnosisBucketPolicy und AWS-SSM-RemediationAutomation-OperationalAccountAdministrationRolePolicy.
- `AWSSSMRemediationExecutionRole` – Eine Automatisierungsausführungsrolle für das Remediation-Runbook. Weitere Informationen zu den Richtlinien finden Sie unter AWS-SSM-RemediationAutomation-ExecutionRolePolicy und AWS-SSM-Automation-DiagnosisBucketPolicy.
- `ManagedInstanceCrossAccountManagementRole` – Ermöglicht Systems Manager, Informationen über verwaltete Knoten kontenübergreifend zu sammeln.

State Manager-Zuordnungen

- `EnableDHMCAssociation` – Wird täglich ausgeführt und stellt sicher, dass die Standard-Host-Verwaltungskonfiguration aktiviert ist.
- `SystemAssociationForEnablingExplorer` – Wird täglich ausgeführt und stellt sicher, dass Explorer aktiviert ist. Explorer wird verwendet, um Daten aus Ihren verwalteten Knoten zu synchronisieren.

- `EnableAREXAssociation` – Wird täglich ausgeführt und stellt sicher, dass AWS Resource Explorer aktiviert ist. Resource Explorer wird verwendet, um zu ermitteln, welche Amazon-EC2-Instances in Ihrer Organisation nicht durch Systems Manager verwaltet werden.
- `SSMAgentUpdateAssociation` – Wird alle 14 Tage ausgeführt und stellt sicher, dass die neueste verfügbare Version von SSM Agent in Ihren verwalteten Knoten installiert ist.
- `SystemAssociationForInventoryCollection` – Wird alle 12 Stunden ausgeführt und sammelt Bestandsdaten von Ihren verwalteten Knoten.

S3-Buckets

- `DiagnosisBucket` – Speichert Daten, die bei der Ausführung des Diagnose-Runbooks gesammelt wurden.

Lambda-Funktionen

- `SSMLifecycleOperatorLambda` – Ermöglicht Prinzipalen Zugriff auf alle AWS Systems Manager Quick Setup-Aktionen.
- `SSMLifecycleResource` – Benutzerdefinierte Ressource zur Verwaltung des Lebenszyklus von Ressourcen, die während des Einrichtungsprozesses erstellt wurden.

Darüber hinaus können Sie nach Abschluss des Einrichtungsvorgangs die Knotenaufgabe diagnostizieren und korrigieren auswählen, um automatisch Korrekturen auf Knoten anzuwenden, die nicht als von Systems Manager verwaltet gemeldet werden. Dies kann die Identifizierung von Problemen wie Netzwerkverbindungsproblemen zu den Systems-Manager-Endpunkten und mehr beinhalten. Weitere Informationen finden Sie unter Diagnose und Abhilfemaßnahmen.

Einrichten der vereinheitlichten Konsole

So richten Sie Systems Manager für eine Organisation ein

1. Anmeldung das Verwaltungskonto für Ihre Organisation.
2. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
3. Geben Sie die ID des Kontos ein, das Sie als delegierter Administrator registrieren möchten.

4. Nachdem das delegierte Administratorkonto erfolgreich registriert wurde, melden Sie sich bei dem delegierten Administratorkonto an, das Sie gerade registriert haben, und kehren Sie zur Systems-Manager-Konsole zurück, um die Einrichtung von Systems Manager abzuschließen.
5. Wählen Sie Systems Manager aktivieren aus.
6. Im Abschnitt Heimatregion bestimmen Sie eine Region, in der Systems Manager Ihre Knotendaten aggregieren soll. Standardmäßig wählt Systems Manager die Region aus, die Sie gerade verwenden. Um eine andere Heimatregion auszuwählen, ändern Sie die Konsole in die Region, die Sie verwenden möchten, bevor Sie Systems Manager einrichten. Knotendaten werden über Konten und Regionen Ihrer Organisation hinweg repliziert und in der Heimatregion gespeichert. Die von Ihnen gewählte Region kann nach der Einrichtung von Systems Manager nicht mehr geändert werden. Um eine andere Region als Heimatregion für Ihre Organisation zu verwenden, müssen Sie die einheitliche Konsole deaktivieren und den Einrichtungsvorgang erneut abschließen. Wenn Ihre Organisation IAM Identity Center verwendet, müssen Sie dieselbe Region, in der Sie IAM Identity Center eingerichtet haben, als Ihre Heimatregion auswählen.
7. Wählen Sie im Abschnitt Regionen die Regionen aus, in denen Sie Systems Manager aktivieren möchten.
8. Wählen Sie im Abschnitt Feature-Konfigurationen die Optionen aus, die Sie für Ihre Konfiguration aktivieren möchten.

   Aktivieren der Standardkonfiguration für die Host-Verwaltung (Default Host Management Configuration, DHMC)

   Ermöglicht Systems Manager, DHMC zu konfigurieren. Mit diesem Feature kann Systems Manager eine IAM-Rolle verwenden, um sicherzustellen, dass alle Amazon-EC2-Instances im Konto und in der Region über die erforderlichen Berechtigungen verfügen, um von Systems Manager verwaltet zu werden. Sie können auch die Häufigkeit der Korrektur von Abweichungen angeben. Eine Konfigurationsabweichung tritt immer dann auf, wenn ein Benutzer Änderungen an einem Service oder einem Feature vornimmt, die mit der in Ihrer Konfiguration getroffenen Auswahl in Konflikt stehen. Systems Manager sucht nach Konfigurationsabweichungen und versucht, diese auf der Grundlage der von Ihnen angegebenen Häufigkeit zu beheben. Sie müssen einen Wert zwischen 1 und 31 Tagen angeben. Wenn Sie DHMC bereits in einer Region konfiguriert haben, ändert Systems Manager die zuvor ausgewählte IAM-Rolle nicht. Weitere Informationen zu DHMC finden Sie unter Automatisches Verwalten von EC2 Instanzen mit der Standard-Host-Management-Konfiguration.

DHMC ermöglicht die Verwaltung von EC2-Instances, ohne dass Sie manuell ein AWS Identity and Access Management (IAM)-Instance-Profil erstellen müssen. Wir empfehlen, diese Option zu wählen, um sicherzustellen, dass Ihre EC2-Instances über die erforderlichen Berechtigungen verfügen, um von Systems Manager verwaltet zu werden.

Aktivieren der Erfassung von Inventarmetadaten

Ermöglicht Systems Manager, die Erfassung der folgenden Arten von Metadaten von Ihren Knoten zu konfigurieren:

- AWS-Komponenten – EC2-Treiber, Agenten, Versionen und mehr.
- Anwendungen – Anwendungsnamen, Publisher, Versionen und mehr.
- Knoten-Details – Systemname, Name des Betriebssystems (OS), OS-Version, letzter Boot-Vorgang, DNS, Domain, Arbeitsgruppe, OS-Architektur und mehr.
- Netzwerkkonfiguration – IP-Adresse, MAC-Adresse, DNS-Gateway, Subnetzmaske und mehr.
- Services – Name, Anzeigename, Status, abhängige Services, Servicetyp, Starttyp und mehr (nur Windows Server-Knoten).
- Windows-Rollen – Name, Anzeigename, Pfad, Featuretyp, Installationsstatus und mehr (nur Windows Server-Knoten).
- Windows-Updates – Hotfix-ID, installiert von, Installationsdatum und mehr (nur Windows Server-Knoten).

Geben Sie die Häufigkeit an, mit der das Inventar erfasst wird. Sie müssen einen Wert zwischen 1 und 744 Stunden angeben. Weitere Informationen zu Inventory, ein Tool in AWS Systems Manager, finden Sie unter AWS Systems Manager-Bestand.

Aktivieren automatischer Systems Manager (SSM)-Agent-Updates

Ermöglicht dem Systems Manager, in der von Ihnen angegebenen Häufigkeit nach einer neuen Version des Agenten zu suchen. Der Wert für die Häufigkeit muss zwischen 1 und 31 Tagen liegen. Wenn es eine neue Version gibt, aktualisiert Systems Manager den Agenten auf Ihrem verwalteten Knoten automatisch auf die neueste veröffentlichte Version. Systems Manager installiert den Agenten nicht auf Instances, auf denen er noch nicht vorhanden ist. Informationen darüber, welche AMIs SSM Agent vorinstalliert haben, finden Sie unter Finden Sie AMIs mit dem vorinstallierten SSM Agent.

Wir empfehlen Ihnen, diese Option zu wählen, um sicherzustellen, dass Ihre Knoten immer die aktuellste Version von SSM Agent ausführen. Weitere Informationen zu SSM Agent, einschließlich Informationen zur manuellen Installation des Agenten, finden Sie unter Arbeiten mit SSM Agent.

9. Wählen Sie Absenden aus.

Je nach Größe Ihres Unternehmens kann es sehr lange dauern, bis die einheitliche Konsolenoberfläche von Systems Manager eingerichtet ist.

Einrichtung einer einheitlichen Systems-Manager-Konsole für ein einziges Konto und eine Region

Um die einheitliche Konsolenerfahrung von Systems Manager für ein einzelnes AWS-Konto und eine AWS-Region einzurichten, müssen Sie keine Organisationen verwenden oder ein delegiertes Administratorkonto registrieren. Der Einrichtungsprozess für die Systems-Manager-Konsole erledigt viele der erforderlichen Aufgaben für Sie. Abhängig von den Features, die Sie konfigurieren möchten, gehören dazu die Aktivierung der Standard-Host-Management-Konfiguration, um Ihren Knoten die erforderlichen IAM-Berechtigungen bereitzustellen, und vieles mehr. Im Folgenden finden Sie eine detaillierte Liste der Ressourcen, die von Systems Manager für die einheitliche Konsole erstellt wurden.

Einheitliche Konsolenressourcen

Abhängig von den Features, die Sie konfigurieren möchten, werden einige Ressourcen möglicherweise nicht erstellt.

IAM roles

- `RoleForOnboardingAutomation` – Ermöglicht Systems Manager, Ressourcen während des Einrichtungsvorgangs zu verwalten. Weitere Informationen zu der Richtlinie finden Sie unter AWSQuickSetupSSMManageResourcesExecutionPolicy.
- `RoleForLifecycleManagement` – Ermöglicht Lambda, den Lebenszyklus von Ressourcen zu verwalten, die durch den Einrichtungsprozess erstellt wurden. Weitere Informationen zu der Richtlinie finden Sie unter AWSQuickSetupSSMLifecycleManagementExecutionPolicy.

- `RoleForAutomation` – Eine Servicerolle, die Systems Manager Automation zur Ausführung von Runbooks übernehmen soll. Weitere Informationen finden Sie unter Erstellen Sie die Servicerollen für Automation mithilfe der Konsole.
- `AWSSSMDiagnosisAdminRole` – Eine Automatisierungsausführungsrolle für das Diagnose-Runbook. Weitere Informationen zu den Richtlinien finden Sie unter AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy, AWS-SSM-Automation-DiagnosisBucketPolicy und AWS-SSM-DiagnosisAutomation-OperationalAccountAdministrationRolePolicy.
- `AWSSSMRemediationAdminRole` – Eine Automatisierungsausführungsrolle für das Remediation-Runbook. Weitere Informationen zu den Richtlinien finden Sie unter AWS-SSM-RemediationAutomation-AdministrationRolePolicy, AWS-SSM-Automation-DiagnosisBucketPolicy und AWS-SSM-RemediationAutomation-OperationalAccountAdministrationRolePolicy.
- `ManagedInstanceCrossAccountManagementRole` – Ermöglicht Systems Manager, Informationen über verwaltete Knoten kontenübergreifend zu sammeln.

State Manager-Zuordnungen

- `EnableDHMCAssociation` – Wird täglich ausgeführt und stellt sicher, dass die Standard-Host-Verwaltungskonfiguration aktiviert ist.
- `SystemAssociationForEnablingExplorer` – Wird täglich ausgeführt und stellt sicher, dass Explorer aktiviert ist. Explorer wird verwendet, um Daten aus Ihren verwalteten Knoten zu synchronisieren.
- `EnableAREXAssociation` – Wird täglich ausgeführt und stellt sicher, dass AWS Resource Explorer aktiviert ist. Resource Explorer wird verwendet, um zu ermitteln, welche Amazon-EC2-Instances in Ihrer Organisation nicht durch Systems Manager verwaltet werden.
- `SSMAgentUpdateAssociation` – Wird alle 14 Tage ausgeführt und stellt sicher, dass die neueste verfügbare Version von SSM Agent in Ihren verwalteten Knoten installiert ist.
- `SystemAssociationForInventoryCollection` – Wird alle 12 Stunden ausgeführt und sammelt Bestandsdaten von Ihren verwalteten Knoten.

S3-Buckets

- `DiagnosisBucket` – Speichert Daten, die bei der Ausführung des Diagnose-Runbooks gesammelt wurden.

Lambda-Funktionen

- `SSMLifecycleOperatorLambda` – Ermöglicht Prinzipalen Zugriff auf alle AWS Systems Manager Quick Setup-Aktionen.
- `SSMLifecycleResource` – Benutzerdefinierte Ressource zur Verwaltung des Lebenszyklus von Ressourcen, die während des Einrichtungsprozesses erstellt wurden.

Darüber hinaus können Sie nach Abschluss des Einrichtungsvorgangs die Knotenaufgabe diagnostizieren und korrigieren auswählen, um automatisch Korrekturen auf Knoten anzuwenden, die nicht als von Systems Manager verwaltet gemeldet werden. Dies kann die Identifizierung von Problemen wie Netzwerkverbindungsproblemen zu den Systems-Manager-Endpunkten und mehr beinhalten.

Einrichten der vereinheitlichten Konsole

So richten Sie Systems Manager für ein einzelnes Konto und eine Region ein

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie Systems Manager aktivieren aus.
3. Wählen Sie im Abschnitt Feature-Konfigurationen die Optionen aus, die Sie für Ihre Konfiguration aktivieren möchten.

   Aktivieren der Standardkonfiguration für die Host-Verwaltung (Default Host Management Configuration, DHMC)

   Ermöglicht Systems Manager, DHMC zu konfigurieren. Mit diesem Feature kann Systems Manager eine IAM-Rolle verwenden, um sicherzustellen, dass alle Amazon-EC2-Instances im Konto und in der Region über die erforderlichen Berechtigungen verfügen, um von Systems Manager verwaltet zu werden. Sie können auch die Häufigkeit der Korrektur von Abweichungen angeben. Eine Konfigurationsabweichung tritt immer dann auf, wenn ein Benutzer Änderungen an einem Service oder einem Feature vornimmt, die mit der in Ihrer Konfiguration getroffenen Auswahl in Konflikt stehen. Systems Manager sucht nach Konfigurationsabweichungen und versucht, diese auf der Grundlage der von Ihnen angegebenen Häufigkeit zu beheben. Sie müssen einen Wert zwischen 1 und 31 Tagen angeben. Wenn Sie DHMC bereits in einer Region konfiguriert haben, ändert Systems Manager die zuvor ausgewählte IAM-Rolle nicht. Weitere Informationen zu DHMC finden

Sie unter Automatisches Verwalten von EC2 Instanzen mit der Standard-Host-Management-Konfiguration.

DHMC ermöglicht die Verwaltung von EC2-Instances, ohne dass Sie manuell ein AWS Identity and Access Management (IAM)-Instance-Profil erstellen müssen. Wir empfehlen, diese Option zu wählen, um sicherzustellen, dass Ihre EC2-Instances über die erforderlichen Berechtigungen verfügen, um von Systems Manager verwaltet zu werden.

Aktivieren der Erfassung von Inventarmetadaten

Ermöglicht Systems Manager, die Erfassung der folgenden Arten von Metadaten von Ihren Knoten zu konfigurieren:

- AWS-Komponenten – EC2-Treiber, Agenten, Versionen und mehr.
- Anwendungen – Anwendungsnamen, Publisher, Versionen und mehr.
- Knoten-Details – Systemname, Name des Betriebssystems (OS), OS-Version, letzter Boot-Vorgang, DNS, Domain, Arbeitsgruppe, OS-Architektur und mehr.
- Netzwerkkonfiguration – IP-Adresse, MAC-Adresse, DNS-Gateway, Subnetzmaske und mehr.
- Services – Name, Anzeigename, Status, abhängige Services, Servicetyp, Starttyp und mehr (nur Windows Server-Knoten).
- Windows-Rollen – Name, Anzeigename, Pfad, Featuretyp, Installationsstatus und mehr (nur Windows Server-Knoten).
- Windows-Updates – Hotfix-ID, installiert von, Installationsdatum und mehr (nur Windows Server-Knoten).

Geben Sie die Häufigkeit an, mit der das Inventar erfasst wird. Sie müssen einen Wert zwischen 1 und 744 Stunden angeben. Weitere Informationen zu Inventory, ein Tool in AWS Systems Manager, finden Sie unter AWS Systems Manager-Bestand.

Aktivieren automatischer Systems Manager (SSM)-Agent-Updates

Ermöglicht dem Systems Manager, in der von Ihnen angegebenen Häufigkeit nach einer neuen Version des Agenten zu suchen. Der Wert für die Häufigkeit muss zwischen 1 und 31 Tagen liegen. Wenn es eine neue Version gibt, aktualisiert Systems Manager den Agenten auf Ihrem verwalteten Knoten automatisch auf die neueste veröffentlichte Version. Systems Manager installiert den Agenten nicht auf Instances, auf denen er noch nicht vorhanden ist. Informationen darüber, welche AMIs SSM Agent vorinstalliert haben, finden Sie unter Finden Sie AMIs mit dem vorinstallierten SSM Agent.

Wir empfehlen Ihnen, diese Option zu wählen, um sicherzustellen, dass Ihre Knoten immer die aktuellste Version von SSM Agent ausführen. Weitere Informationen zu SSM Agent, einschließlich Informationen zur manuellen Installation des Agenten, finden Sie unter Arbeiten mit SSM Agent.

4. Wählen Sie Absenden aus.

Deaktivieren der einheitlichen Systems-Manager-Konsole

Um die einheitliche Systems-Manager-Konsole für eine Organisation zu deaktivieren, benötigen Sie Zugriff auf das Konto, das Sie als delegierter Administrator für Systems Manager registriert haben. Nachdem Sie sich beim delegierten Administratorkonto für Ihre Organisation angemeldet haben, können Sie die Einrichtung für Ihre Organisation im Bereich Einstellungen der einheitlichen Konsole deaktivieren. Wenn Sie das vereinheitlichte Konsolen-Setup für Ihre Organisation deaktivieren, löscht Systems Manager die Ressourcen, einschließlich der verwalteten Ansichten von Ressourcen-Explorer, die während des Einrichtungsvorgangs erstellt wurden. Wenn Sie das Setup für Ihre Organisation deaktivieren, wird der vertrauenswürdige Zugriff nicht widerrufen oder die Registrierung der delegierten Administratorkonten für abhängige Services aufgehoben. Im folgenden Verfahren wird beschrieben, wie Sie das Setup für die einheitliche Konsole deaktivieren.

So deaktivieren Sie das Setup für die vereinheitlichte Systems-Manager-Konsole

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Einstellungen aus.
3. Wählen Sie Deaktivieren aus. Sie müssen bestätigen, dass Sie das Setup für Ihre Organisation deaktivieren möchten. Diese Aktion löscht die für die einheitliche Konsole erstellten Ressourcen und kann nicht rückgängig gemacht werden.

Wenn Sie nicht auf das delegierte Administratorkonto für Systems Manager zugreifen können und das Setup für die einheitliche Konsole deaktivieren möchten, können Sie dies auch über das Verwaltungskonto für Ihre Organisation tun. Rufen Sie mit dem SDK AWS CLI oder den `DeleteConfigurationManager` API-Vorgang auf und übergeben Sie den `ManagerArn` Wert für das Organisations-Setup in Ihrem Konto. Das Format für den Manager-ARN, der zum Einrichten der vereinheitlichten Konsole verwendet wird, lautet wie folgt:

`arn:aws:ssm-quicksetup:`*`account-id`*`:configuration-manager/`*`configuration-manager-id`*.

> **Note**
>
> Wenn Sie den Wert für nicht kennen*`configuration-manager-id`*, rufen Sie die `ListConfigurationManagers` API-Aktion auf und filtern Sie die Ergebnisse anhand des `AWSQuickSetupType-SSM` Typs.

Ausführen von Knotenverwaltungsaufgaben mit AWS Systems Manager

In den folgenden Themen wird beschrieben, wie Sie allgemeine Knotenaufgaben mithilfe der vereinheitlichten AWS Systems Manager-Konsole für eine AWS Organizations-Organisation und ein einzelnes AWS-Konten ausführen.

Themen

- Überprüfen von Erkenntnissen
- Erkunden von Knoten
- Just-in-Time-Knotenzugriff mit Systems Manager
- Diagnose und Abhilfemaßnahmen
- Systems-Manager-Einstellungen anpassen

Überprüfen von Erkenntnissen

Mithilfe der einheitlichen Systems Manager-Konsole können Sie Einblicke in den Gesamtstatus verwalteter Knoten und nicht verwalteter EC2 Instanzen in Ihrer Organisation oder Ihrem Konto gewinnen.

Systems Manager bietet einen visuellen Überblick über Ihre verwalteten Knoten und EC2 Instanzen, die noch nicht von Systems Manager verwaltet werden. (Ein verwalteter Knoten ist jede Maschine, die für die Verwendung mit Systems Manager in Hybrid- und Multi-Cloud-Umgebungen konfiguriert ist. Informationen zu unterstützten Maschinentypen finden Sie unter Unterstützte Maschinentypen in Hybrid- und Multi-Cloud-Umgebungen.)

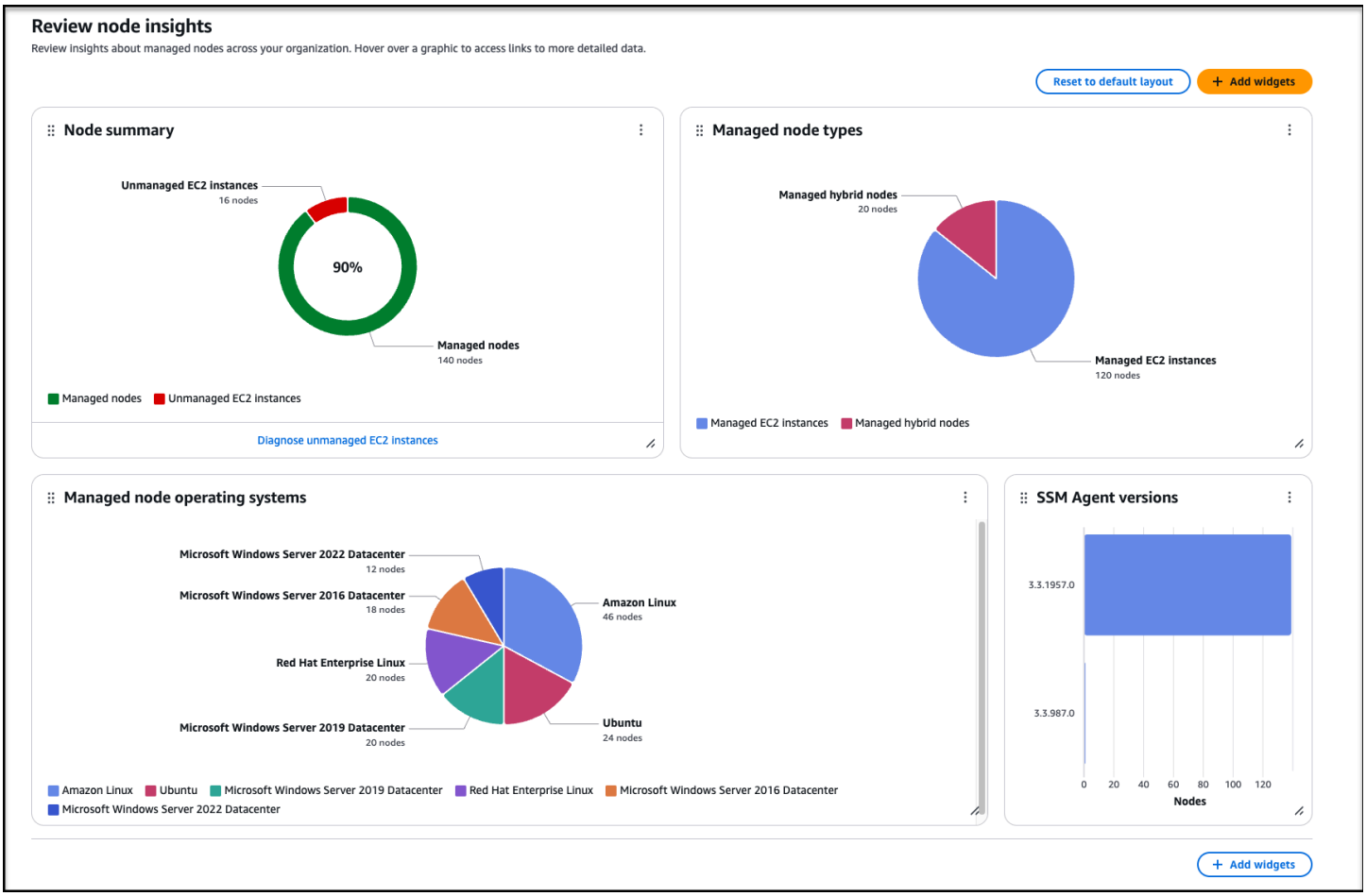
Dieser Überblick wird in einzelnen Berichtsfeldern, sogenannten Widgets, bereitgestellt, die interaktive Kreisdiagramme und andere Grafiken enthalten.

Bevor Sie beginnen

Um Knoteneinblicke zu überprüfen, müssen Sie zunächst Ihre Organisation oder Ihr Konto in die vereinheitlichte Systems-Manager-Konsole integrieren. Weitere Informationen finden Sie unter Einrichten von AWS Systems Manager.

Öffnen Sie nach dem Onboarding die Systems-Manager-Konsole und wählen Sie Knoteneinblicke überprüfen aus.

Die folgende Abbildung zeigt die einzelnen Berichtsfelder, Widgets, genannt, die auf der Seite Knoteneinblicke überprüfen verfügbar sind.



Die Anzeige unterstützt Widgets, die Ihnen die folgenden Informationen liefern.

Zusammenfassung des Knotens

Gibt an, wie viele EC2 Instanzen in Ihrer Organisation oder Ihrem Konto derzeit keine verwalteten Knoten sind und wie viele verwaltete Knoten sich in der Flotte Ihrer Organisation oder Ihres Kontos befinden.

Was ist eine nicht verwaltete Instance?

Wenn Sie eine verwaltete EC2 Instanz beenden, wird sie in der Systems Manager Manager-Konsole als „Nicht verwaltet“ gemeldet. Dieses Verhalten ist zu erwarten, da SSM Agent keine aktive Verbindung zum Service hat.

> **Note**
>
> Dies unterscheidet sich von der AWS Config Definition einer Instanz als nicht verwaltet. Wenn eine Instanz derzeit gestoppt ist, wird der Status der Instanz AWS Config gemeldet, als das letzte Mal eine „Heartbeat“ -Verbindung zwischen SSM Agent der Instanz und dem Systems Manager Manager-Dienst hergestellt wurde.

Wenn die Instanz neu gestartet wird, stellt sie automatisch wieder eine Verbindung zum Systems-Manager-Service her, und ihr Status in der vereinheitlichten Konsole wird innerhalb von fünf Minuten auf „Verwaltet“ zurückgesetzt. Es ist kein manuelles Eingreifen erforderlich, und alle Systems Manager Manager-Konfigurationen für die Instanz werden während des Stop/Start Zyklus beibehalten.

Wenn die Instance jedoch einige Minuten nach dem Start immer noch nicht als „Verwaltet“ gemeldet wird, ist sie wahrscheinlich nicht richtig für die Systems-Manager-Verwaltung konfiguriert. In diesem Fall empfehlen wir, eine Diagnose durchzuführen, um festzustellen, warum sich die Instance weiterhin in einem nicht verwalteten Zustand befindet. Weitere Informationen finden Sie unter Diagnose und Behebung nicht verwalteter Amazon-EC2-Instances in Systems Manager.

Wenn der Diagnosescan das Problem nicht ermitteln kann, überprüfen Sie anhand der folgenden ThemenSSM Agent, ob die Anforderungen für AWS Identity and Access Management (IAM) -Rollen und die Voraussetzungen für Systems Manager erfüllt sind:

- Fehlerbehebung für SSM Agent
- Konfigurieren von erforderlichen Instance-Berechtigungen für Systems Manager
- Problembehandlung bei der Verfügbarkeit verwalteter Knoten

Typen verwalteter Knoten

Gibt an, wie viele verwaltete Knoten in Ihrer Flotte EC2 Instanzen sind und wie viele andere Servertypen es gibt, darunter Server in Ihren eigenen Räumlichkeiten (lokale Server), AWS IoT Greengrass Kerngeräte AWS IoT und Geräte, die keine AWS Edge-Geräte sind, sowie virtuelle Maschinen (VMs), auch VMs in anderen Cloud-Umgebungen. Sie können den Mauszeiger über die Grafik mit den Knotentypen bewegen, um Links zu weiteren Details auf der Seite Knoten erkunden aufzurufen.

Weitere Informationen zur AWS Unterstützung von Hybrid- und Multicloud-Umgebungen finden Sie unter AWS Lösungen für Hybrid- und Multicloud-Umgebungen.

SSM Agent-Versionen

Enthält Informationen zu Installationen von AWS Systems Manager Agent (SSM Agent) in Ihrer Flotte. SSM Agentist Amazon-Software, die auf Ihren verwalteten Knoten läuft. SSM Agentermöglicht es Systems Manager, diese Ressourcen zu aktualisieren, zu verwalten und zu konfigurieren. Der Agent verarbeitet Anfragen vom Systems Manager Manager-Dienst in der AWS Cloud und führt sie dann wie in der Anfrage angegeben aus.

Für verwaltete Knoten in Ihrer Flotte berichtet dieses Widget über die SSM Agent-Versionen in Ihrer Flotte, von den neuesten bis zu den ältesten. Sie können den Mauszeiger über die SSM Agent-Versionsgrafik bewegen, um auf der Seite Knoten erkunden auf Links zu weiteren Details zuzugreifen.

Mehr über SSM Agent erfahren Sie unter Arbeiten mit SSM Agent.

Betriebssysteme für verwaltete Knoten

Bietet eine Aufschlüsselung des Prozentsatzes der einzelnen Betriebssysteme auf verwalteten Knoten in Ihrer Flotte. Sie können den Mauszeiger über die Grafik Verwaltete Knoten nach Betriebssystemen bewegen, um Links zu weiteren Informationen auf der Seite Knoten erkunden aufzurufen.

Sie können das Widget-Layout auf der Seite „Knoteninformationen überprüfen“ anpassen, indem Sie eine drag-and-drop Funktion verwenden und Widgets entfernen und der Anzeige hinzufügen.

Verwenden Sie die Informationen in den folgenden Themen, die Ihnen beim Arbeiten mit den Systems Manager Node Insights Widgets helfen.

Themen

- Hinzufügen oder Entfernen von Widgets auf der Seite mit den Erkenntnissen des Review-Knoten
- Widgets auf der Seite mit den Erkenntnissen des Überprüfungsknotens neu anordnen

Hinzufügen oder Entfernen von Widgets auf der Seite mit den Erkenntnissen des Review-Knoten

Sie können das Layout auf der Seite mit den Erkenntnissen zum Review-Knoten von Systems Manager anpassen, indem Sie Widgets hinzufügen und entfernen.

> **Note**
>
> Standardmäßig werden auf der Seite alle verfügbaren Widgets angezeigt.

So fügen Sie Widgets auf der Seite mit den Erkenntnissen des Review-Knoten hinzu oder entfernen diese

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im linken Navigationsbereich Erkenntnisse des Überprüfungsknotens aus.
3. Gehen Sie wie folgt vor, um ein Widget von der Anzeige zu entfernen:
   a. Wählen Sie das Menü Weitere Optionen (⋮) für das Widget.
   b. Wählen Sie Remove widget (Widget entfernen) aus.
4. Gehen Sie wie folgt vor, um der Anzeige ein Widget hinzuzufügen:
   a. Wählen Sie Widgets hinzufügen aus.
   b. Klicken Sie im Bereich Widgets hinzufügen auf den Ziehgriff (⠿) des Widgets, das Sie der Anzeige hinzufügen möchten, und halten Sie die Maustaste gedrückt.
   c. Ziehen Sie das Widget und legen Sie es im Hauptbereich ab.

Widgets auf der Seite mit den Erkenntnissen des Überprüfungsknotens neu anordnen

Sie können das Layout auf der Seite mit den Erkenntnissen des Review-Knotens anpassen, indem Sie die Widgets neu anordnen.

So können Sie Widgets auf der Seite mit den Erkenntnissen des Überprüfungsknotens neu anordnen

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Erkenntnisse des Überprüfungsknotens aus.
3. Um das Widget-Layout anzupassen, wählen Sie ein Widget aus, das Sie verschieben möchten. Klicken Sie auf den Namen des Ziehgriffs ( ), halten Sie ihn und ziehen Sie ihn dann an seine neue Position.
4. Wiederholen Sie diesen Vorgang für jedes Widget, das Sie neu positionieren möchten.

Wenn Sie sich entscheiden, dass Ihnen das neue Layout nicht gefällt, wählen Sie Auf benutzerdefiniertes Layout zurücksetzen, um alle Widgets wieder an ihren ursprünglichen Speicherort zu verschieben.

Erkunden von Knoten

Auf der Seite Knoten erkunden in Systems Manager können Sie die Details der verwalteten Knoten in Ihrer Organisation oder Ihrem Konto anhand der Kriterien überprüfen, die Sie in Filtern angeben. Sie können auch die Systems-Manager-Integration mit Amazon Q Developer (Amazon Q), einer AWS generativen KI-Lösung, verwenden, um mithilfe von Text-Prompts zu suchen.

Bevor Sie beginnen

Um das Feature Knoten erkunden verwenden zu können, müssen Sie zunächst Ihre Organisation oder Ihr Konto in die vereinheitlichte Systems Manager-Konsole integrieren. Weitere Informationen finden Sie unter Einrichten der einheitlichen Systems-Manager-Konsole für eine Organisation.

Öffnen Sie nach dem Onboarding die Systems-Manager-Konsole und wählen Sie Knoten erkunden.

> **Note**
>
> Wenn Sie einen Aggregatorindex für Resource Explorer in einer anderen Region als Ihrer Heimatregion erstellt haben, stuft Systems Manager den aktuellen Index herab. Anschließend bewirbt Systems Manager den lokalen Index in Ihrer Heimatregion als neuen Aggregatorindex. Während dieser Zeit werden nur Knoten für Ihre Heimatregion angezeigt. Dieser Vorgang kann bis zu 24 Stunden dauern.

Themen

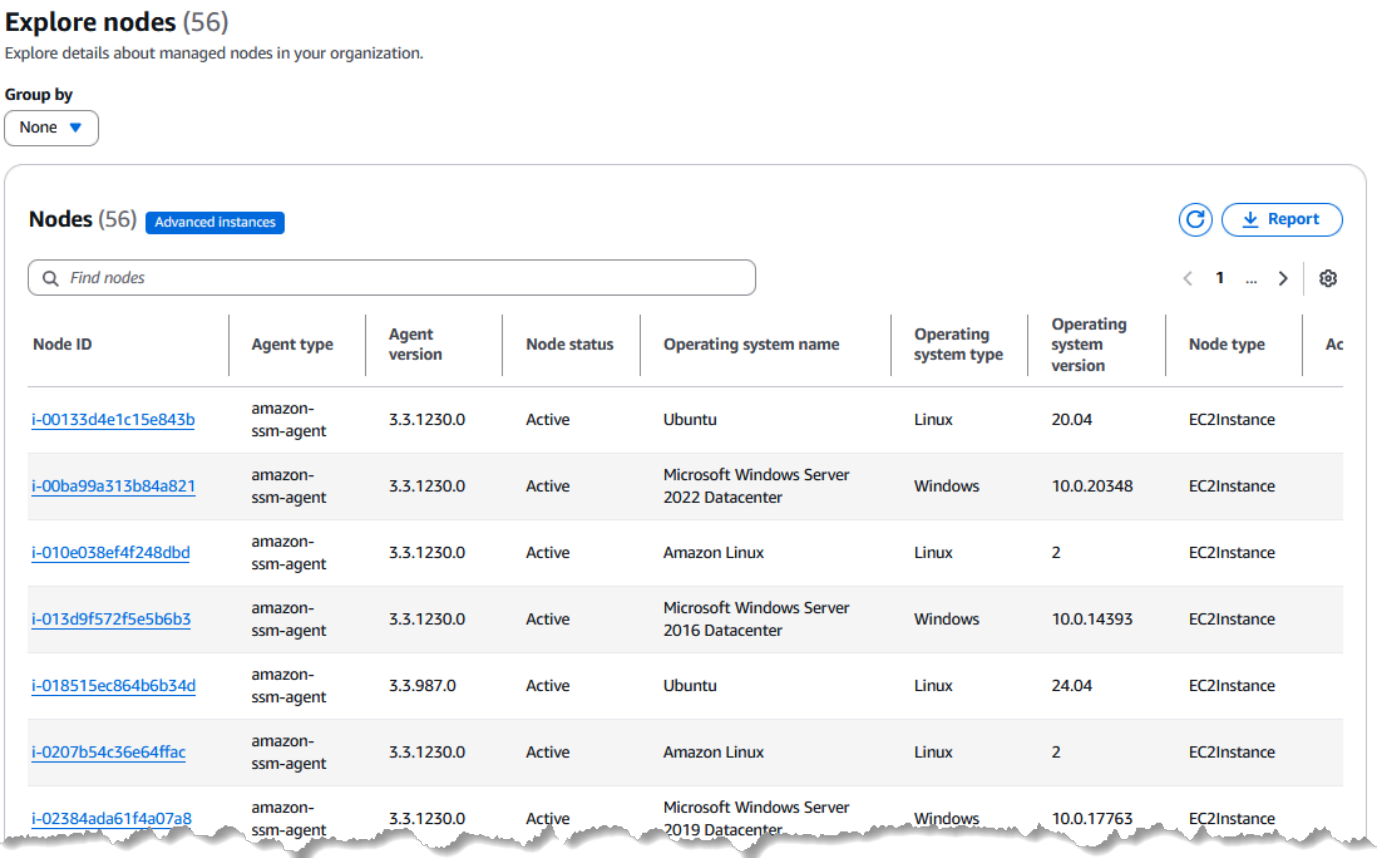
- Erkunden von Knoten mithilfe von Konsolenfiltern
- Erkunden von Knoten mithilfe von Text-Prompts in Amazon Q
- Details zu einzelnen Knoten anzeigen und Maßnahmen für einen Knoten ergreifen
- Bericht über verwaltete Knoten herunterladen oder exportieren
- Inhalt und Erscheinungsbild von Knotenberichten verwalten

Erkunden von Knoten mithilfe von Konsolenfiltern

In der Systems-Manager-Konsole können Sie dann Ihre verwalteten Knoten nach den folgenden Ansichten gruppieren:

All nodes (No filter)

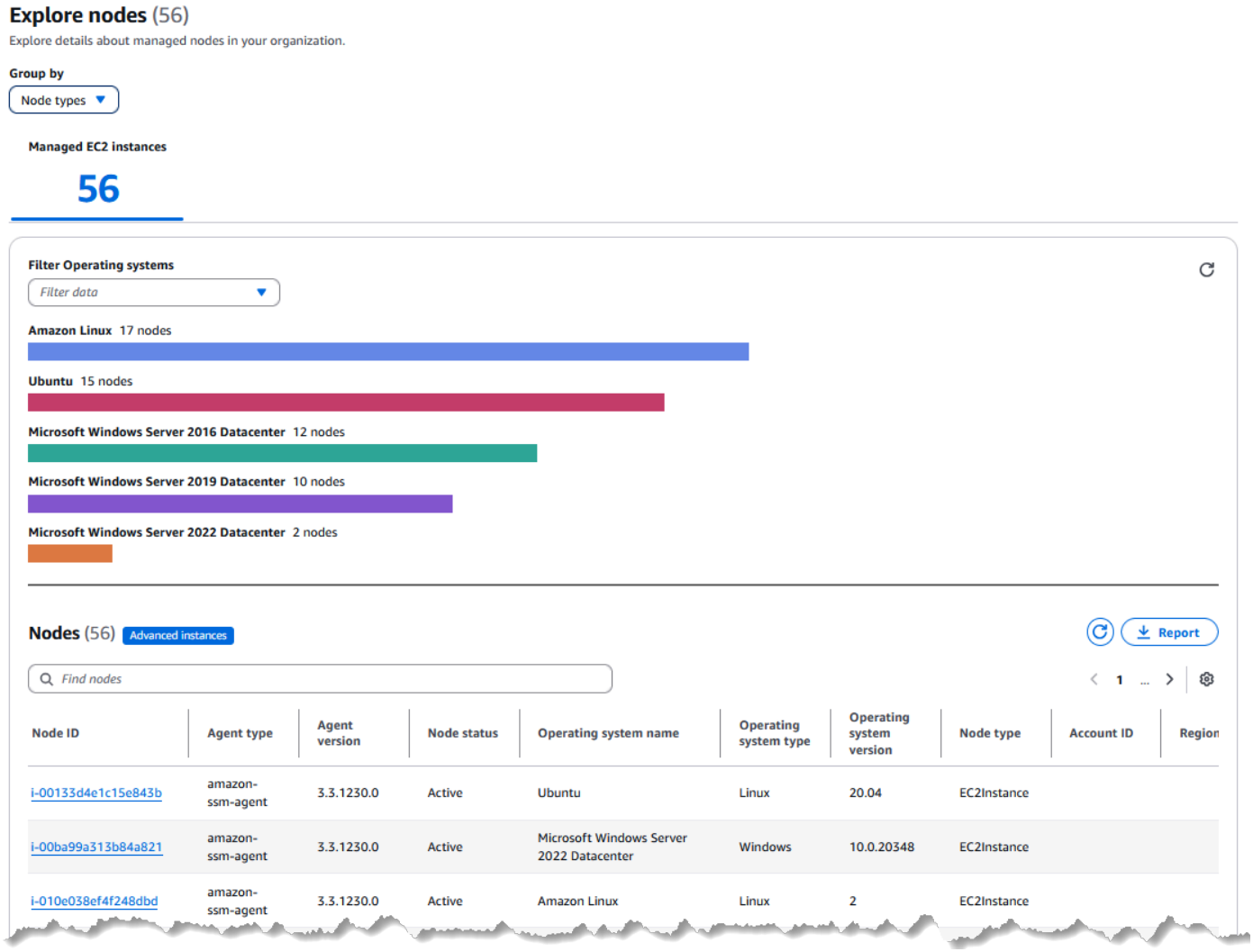
Listet alle verwalteten Knoten in Ihrem Unternehmen oder Konto auf.



Node types

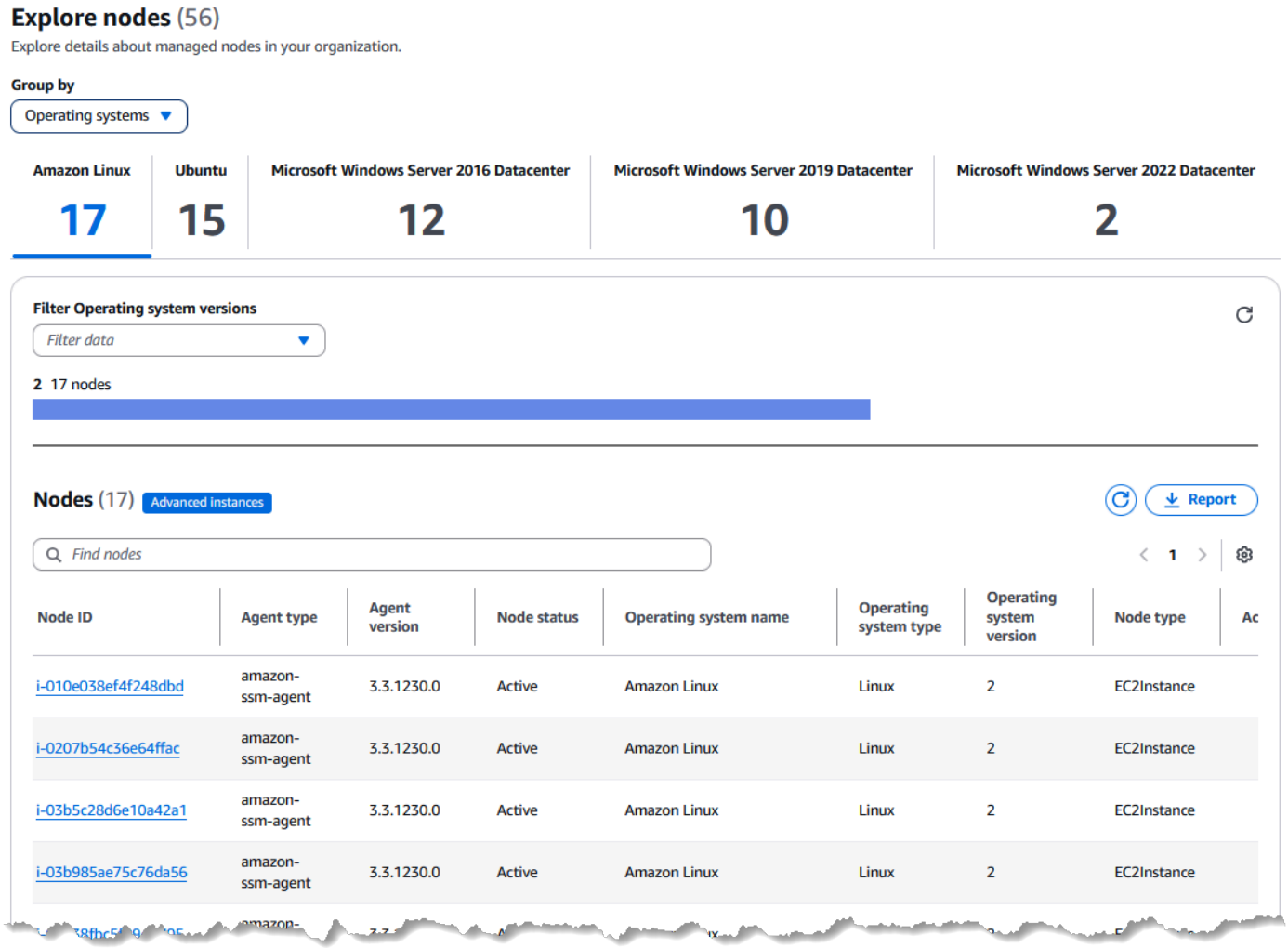
Bietet Tabs zum separaten Anzeigen von Daten für Amazon Elastic Compute Cloud (Amazon EC2)-Instances und andere Maschinentypen, einschließlich Server in Ihrem eigenen Gebäude

(On-Premises-Server), AWS IoT Greengrass-Kerngeräte, AWS IoT- und AWS-Nicht-Edge-Geräte und virtuelle Maschinen (VMs), einschließlich VMs in anderen Cloud-Umgebungen.



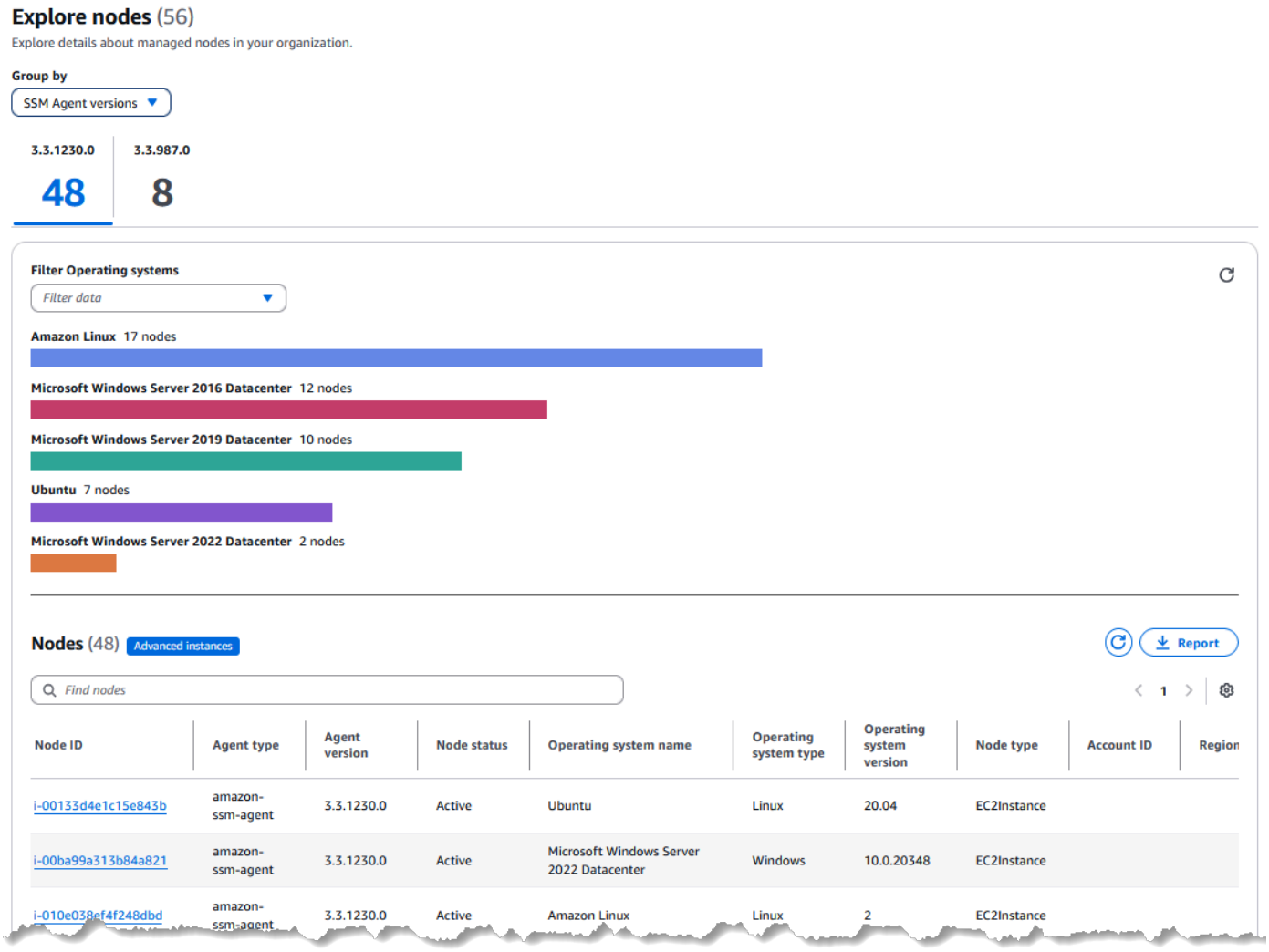
Operating systems

Stellt eine Registerkarte für jeden Betriebssystemtyp in Ihrer Organisation oder Ihrem Konto bereit, z. B. Amazon Linux und Microsoft Windows Server 2022 Datacenter. Auf jeder Registerkarte können Sie die Liste weiter filtern, indem Sie nur bestimmte Versionen der Betriebssysteme auswählen, z. B. Amazon Linux 2 und Amazon Linux 2023.



SSM-Agent versions

Stellt eine Registerkarte für jede Version von SSM Agent bereit, die auf verwalteten Knoten in Ihrer Flotte installiert ist. Auf jeder Registerkarte können Sie die Liste weiter filtern, indem Sie nur bestimmte Betriebssysteme wie Amazon Linux und Microsoft Windows Server 2022 Datacenter auswählen.



Darüber hinaus können Sie für jede dieser Ansichten die Liste der gemeldeten Knoten weiter verfeinern, indem Sie festlegen, dass nur Knoten für eine bestimmte Eigenschaft angezeigt werden, z. B. für den Knotenstatus, die AWS-Konto-ID, die ID der Organisationseinheit und mehr.

Sie können die Berichtsanzeige anpassen, indem Sie auswählen, welche der verfügbaren Datenspalten auf der Seite Knoten durchsuchen angezeigt werden. Sie können auch Berichte in CSV- oder JSON-Formaten herunterladen oder Berichte im CSV-Format nach Amazon S3 exportieren.

Themen

- Auswahl einer Filteransicht für Zusammenfassungen verwalteter Knoten

Auswahl einer Filteransicht für Zusammenfassungen verwalteter Knoten

Auf der Seite Knoten erkunden in Systems Manager können Sie aggregierte Daten über Ihre Flotte anhand einer Reihe verfügbarer Filteransichten anzeigen.

So wählen Sie eine Filteransicht für Zusammenfassungen verwalteter Knoten aus

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Knoten erkunden aus.
3. Wählen Sie für die Filteransicht eine der Filteroptionen aus und verfeinern Sie den Bericht optional weiter:

   - Verwaltete Knoten – Im Suchfeld
     ( )
     können Sie eine Eigenschaft und ein Trennzeichen auswählen, z. B. `Node type = Managed EC2 instances`.
   - Betriebssysteme – In der Liste Betriebssystemversionen filtern können
     Sie eine Betriebssystemversionsnummer auswählen. Im Suchfeld
     ( )
     können Sie eine Eigenschaft und ein Trennzeichen auswählen, z. B `Node type = Managed EC2 instances`.
   - SSM Agent-Versionen – In der Liste Betriebssysteme filtern
     können Sie einen Betriebssystemnamen auswählen. Im Suchfeld
     ( )
     können Sie eine Eigenschaft und ein Trennzeichen auswählen, z. B `Node type = Managed EC2 instances`.
   - Knotentypen – In der Liste Betriebssysteme filtern können Sie einen Betriebssystemnamen auswählen. Im Suchfeld
     ( )
     können Sie eine Eigenschaft und ein Trennzeichen auswählen, z. B `Node type = Managed EC2 instances`.

Nachdem Sie die Liste optional gefiltert haben, können Sie Details zu einem bestimmten verwalteten Knoten anzeigen, indem Sie dessen ID in der Spalte Knoten-ID auswählen. In dieser Detailansicht können Sie eine Reihe von Aktionen auf dem Knoten ausführen.

Erkunden von Knoten mithilfe von Text-Prompts in Amazon Q

Mithilfe der Systems-Manager-Integration mit Amazon Q Developer können Sie Text-Prompts verwenden, um Informationen zu Ihren verwalteten Knoten anzuzeigen, die von generativer KI erstellt wurden.

Amazon Q Developer ist ein generativer KI-gestützter Konversationsassistent, der Ihnen helfen kann, AWS-Anwendungen zu verstehen, zu erstellen, zu erweitern und zu betreiben. Damit Sie schneller auf AWS aufbauen können, wird das Modell, das Amazon Q zugrunde liegt, um qualitativ hochwertige AWS-Inhalte erweitert, um vollständigere, umsetzbarere und referenziertere Antworten zu erhalten. Weitere Informationen finden Sie unter Was ist Amazon Q Developer? im Benutzerhandbuch zu Amazon Q Developer.

Durch die Integration zwischen Systems Manager und Amazon Q erhalten Sie schnell Transparenz und Kontrolle über große, verteilte Umgebungen in mehreren AWS-Konten-Regionen. Sie können Abfragen in natürlicher Sprache verwenden, um Knotendaten schnell zu durchsuchen und dann Probleme zu identifizieren und schneller Maßnahmen zu ergreifen.

Wenn Sie eine Frage in natürlicher Sprache zu verwalteten Knoten oder verwalteten Instances stellen, verwendet Amazon Q die Systems-`ListNodes`-Manager-Aktion und erstellt Filter auf der Grundlage Ihrer Texteingabe, um Ergebnisse abzurufen.

Angenommen, Sie geben Amazon Q den folgenden Prompt:

`List my managed nodes running Red Hat Enterprise Linux 9.2`

Amazon Q bestimmt, welche Filter in eine Anfrage aufgenommen werden sollen, und führt dann eine Abfrage aus, die der folgenden ähnelt:

```
aws ssm list-nodes \
    --filters Key=PlatformName,Values='Red Hat Enterprise Linux',Type=Equal
 Key=PlatformVersion,Values=9.2,Type=Equal
```

Amazon Q generiert dann einen Bericht über Red Hat Enterprise Linux-Instances in Ihrem Konto, der Informationen wie die Anzahl der Instances, ihre IDs und ihre Regionen auflistet.

Sie können auch eine JSON-Zusammenfassung der Details jeder Instance anzeigen und einen Link öffnen, um die gesamte Liste der EC2-Instances oder verwalteten Knoten auf der Seite Knoten durchsuchen von Systems Manager anzuzeigen. Knoten erkunden zeigen Ergebnisse an, die den

Filterkriterien entsprechen, die Sie in Ihrem Prompt angegeben haben. Von dort aus können Sie die Filter für Ihre Anfrage ändern oder verfeinern, wie unter Erkunden von Knoten beschrieben.

Themen

- Lernen Sie, effektive Prompts zu erstellen, um Amazon Q nach Ihrer Flotte zu fragen
- Verwaltete Knoten mit Amazon Q erkunden

Lernen Sie, effektive Prompts zu erstellen, um Amazon Q nach Ihrer Flotte zu fragen

Je besser die Frage oder der Prompt ist, die Sie Amazon Q geben, desto besser ist das Ergebnis, das Sie erhalten.

Tipps für Abfrage-Prompts

Beachten Sie die folgenden Tipps, wenn Sie Amazon Q nach Ihrer Flotte fragen:

1. Um die Genauigkeit Ihrer Ergebnisse zu verbessern, verwenden Sie in Ihren Eingabe-Prompts die Begriffe „verwaltete Knoten“ und „verwaltete Instances“ statt nur „Knoten“ und „Instances“.
2. Um Ergebnisse für mehrere Konten abzufragen, die Teil einer Organisation sind, wie unter konfiguriert AWS Organizations, müssen Sie mit dem delegierten Administratorkonto in der angegebenen Heimatregion angemeldet sein.
3. Verwenden Sie im delegierten Administratorkonto Begriffe, um Amazon Q zu verdeutlichen, dass Sie nach Knoten und Instances in der gesamten Organisation fragen, indem Sie speziell Begriffe wie „in meiner Organisation“ oder „in meinem Konto 123456789012“ verwenden.

Themen

- Beispielfragen für Amazon Q
- Unterstützte Betriebssystemnamen und Versionen für Eingabe-Prompts

Beispielfragen für Amazon Q

In der folgenden Tabelle finden Sie Beispielfragen, die zeigen, wie Sie Abfragen von Amazon Q erstellen können, die zu besseren Ergebnissen führen.

Wir stellen auch Beispiele für die Filter bereit, die Amazon Q bei der Ausführung des ListNodesBefehls anwendet und die aus dem Inhalt Ihrer Aufforderung generiert werden.

Beispiel für eine Frage in natürlicher Sprache	Amazon Q hat Filter angewendet
Show me my Windows managed nodes.	`PlatformType = Windows`
List my managed instances in account 123456789012.	`AccountId = 123456789012`
Show me all managed nodes running Amazon Linux 2 across my organization.	`PlatformName = Amazon Linux` `PlatformVersion = 2`
Show me all managed instances running Microsoft Windows Server 2019 Datacenter in my organization.	`PlatformName = Microsoft Windows Server 2019 Datacenter`
Can you show me all managed nodes with SSM Agent version 3.3.1142.0?	`AgentType = amazon-ssm-agent` `AgentVersion = 3.3.1142.0`

List all Amazon Linux 2 managed instances in account 123456789012 that have SSM Agent version 3.3.1230.0.	`PlatformName = Amazon Linux`
	`PlatformVersion = 2`
	`AccountId = 123456789012`
	`AgentType = amazon-ssm-agent`
	`AgentVersion = 3.3.1230.0`

What Microsoft Windows Server 2012 R2 Enterprise managed nodes are running in the eu-central-1 region across my entire organization?	`PlatformName = Microsoft Windows Server 2012 R2 Enterprise` `Region = eu-central-1`

Show me all managed instances running Red Hat Linux 7 in ou-d6ty-gxdma6vm.	`PlatformName = RHEL Linux`
	`PlatformVersion = 7`
	`OrganizationalUnitId = ou-d6ty-gxdma6vm`

Beispiel für eine Frage in natürlicher Sprache	Amazon Q hat Filter angewendet
What Ubuntu managed instances are in account 123456789012?	`PlatformName = Ubuntu` `AccountId = 123456789012`
List my Linux managed instances.	`PlatformType = Linux`
Find my macOS managed nodes.	`PlatformType = macOS`
Show me all versions of Amazon Linux managed nodes in my org.	`PlatformName = Amazon Linux`
List managed nodes running Amazon Linux 2.	`PlatformName = Amazon Linux` `PlatformVersion = 2`

List the managed nodes with Ubuntu 16.04 in account 123456789012.	`PlatformName = Ubuntu`
	`PlatformVersion = 16.04`
	`AccountId = 123456789012`

Find all managed nodes that have an SSM Agent version that is not 3.3.987.0.	`AgentType = amazon-ssm-agent` `AgentVersion != 3.3.987.0`
List all managed instances that are not running a Linux operating system.	`PlatformType != Linux`

Unterstützte Betriebssystemnamen und Versionen für Eingabe-Prompts

Wenn Sie Amazon Q nach den verwalteten Knoten in Ihrem Konto fragen, ist es hilfreich, den Namen eines Betriebssystems mit der Bezeichnung im Systems Manager anzugeben. Sie können auch Versionsnummern angeben, um Ihre Ergebnisse weiter einzugrenzen. Wie in den folgenden Tabellen dargestellt, könnten Sie beispielsweise nach Ergebnissen fragen, die sich speziell auf **`macOS 14.5`**,

`Microsoft Windows Server 2019 Datacenter` und **`AlmaLinux 9.2 through 9.4`** beziehen, um nur einige Beispiele zu nennen.

Diese Listen sind möglicherweise nicht vollständig und werden nur als Beispiele angeboten.

macOS

Plattformname	Versionsnummern
macOS	13.2, 13.4, 13.7, 14.1, 14.5, 14.6.1, 15.0

Windows

Versionen	Versionsnummern
Microsoft Windows Server 2012 R2 Rechenzentrum	6.3.9600
Microsoft Windows Server 2012 R2 Standard	6,3,9600
Microsoft Windows Server 2012 Standard	6.2.9200
Microsoft Windows Server 2016 Rechenzentrum	N/A
Microsoft Windows Server 2016 Standard	10.0.14393
Microsoft Windows Server 2019 Rechenzentrum	N/A
Microsoft Windows Server 2019 Standard	N/A
Microsoft Windows Server 2022 Rechenzentrum	N/A
Microsoft Windows Server 2022 Standard	10,0,20348

Linux

Plattformnamen	Versionsnummern
AlmaLinux	8.10, 9.2, 9.3, 9.4
Amazon Linux 2	2.0 und höher
Amazon Linux 2023	2023.0.20230315.0 und höher
BottleRocket	1.14.3, 1.16.1, 1.18.0, 1.19.1, 1.19.2, 1.19.5, 1.20.0, 1.20.1, 1.20.2, 1.20.3, 1.20.5, 1.21.1, 1.23.0, 1.24.0, 1.24.1, 1.25.0, 1.26.1,
CentOS Stream	9
Debian GNU/Linux	11-12
Oracle Linux Server	7.8, 8.2, 8.3, 8.8, 8.9, 8.10, 9.4
Red Hat Enterprise Linux	8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8, 8.9, 8.10, 9.2, 9.3, 9.4
Red Hat Enterprise Linux-Server	17.3, 7.6, 7.7, 7.8, 7.9
Rocky Linux	8.6, 8.7, 8.8, 8.9, 8.10, 9.1, 9.2, 9.3, 9.4
Ubuntu Server	16.04, 18.04, 20.04, 22.04, 24.04

Verwaltete Knoten mit Amazon Q erkunden

Durch die Systems-Manager-Integration mit Amazon Q Developer können Sie von überall in der AWS-Managementkonsole, wo die Amazon-Q-Schnittstelle verfügbar ist, Fragen zu verwalteten Knoten in Ihrer Flotte stellen.

Weitere Informationen zur Interaktion mit Amazon Q finden Sie unter Chatten mit Amazon Q Developer überr AWS im Benutzerhandbuch zu Amazon Q Developer.

So können Sie verwaltete Knoten mit Amazon Q erkunden

1. Wählen Sie an einer beliebigen Stelle in der AWS-Managementkonsole das Amazon-Q-Symbol ( ).
2. Stellen Sie im Prompt-Feld unten im Amazon-Q-Bereich eine Frage zu verwalteten Knoten in Ihrem Konto oder Ihrer Organisation.

   > **Tip**
   >
   > Tipps zur Erstellung effektiver Prompts finden Sie in Lernen Sie, effektive Prompts zu erstellen, um Amazon Q nach Ihrer Flotte zu fragen den Informationen unter.

3. Prüfen Sie die Informationen zu bestimmten Knoten, oder wählen Sie AWS Systems Manager-Konsole öffnen, um mit der Suche fortzufahren.

Details zu einzelnen Knoten anzeigen und Maßnahmen für einen Knoten ergreifen

Aus einer Liste auf der Seite Knoten durchsuchen in Systems Manager können Sie einen einzelnen Knoten auswählen, um umfassende Details über den Computer anzuzeigen oder eine Vielzahl von Aktionen auf dem Knoten auszuführen. Die Seite Allgemein auf der Detailseite enthält umfassende Informationen über den Knoten.

So zeigen Sie die Details einzelner Knoten an und ergreifen Maßnahmen für einen Knoten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Knoten erkunden aus.
3. (Optional) Gehen Sie wie unter Auswahl einer Filteransicht für Zusammenfassungen verwalteter Knoten beschrieben vor, um die Liste der verwalteten Knoten, die für Ihre Organisation oder Ihr Konto angezeigt wird, zu verfeinern.
4. Wählen Sie in der Spalte Knoten-ID die verknüpfte ID eines Knotens aus.
5. Um weitere Details über den Knoten anzuzeigen, wählen Sie im linken Navigationsbereich in der Eigenschaftenliste eine Eigenschaft aus, zu der Sie weitere Informationen erhalten möchten:

- Tags – Zeigt eine Liste von Tags an, die auf den Knoten angewendet werden. Sie können auch Markierungen hinzufügen oder entfernen.
- Inventar – Wählen Sie einen Inventartyp aus, z. B. AWS:Application oder AWS:Network, um Inventardetails für den Knoten anzuzeigen.
- Verknüpfungen – Zeigt Details zu allen State Manager-Zuordnungen an, die auf den Knoten angewendet wurden, einschließlich Details wie Status und Name des zugehörigen SSM-Dokuments.
- Patches – Zeigt zusammenfassende Informationen über Patches und den Patch-Status für den Knoten an.
- Konformität mit der Konfiguration – Zeigt Konformitätsdetails für den Knoten an, z. B. den Konformitätsstatus und den Schweregrad des Kompatibilitätsproblems.

Weitere Informationen zu Details auf den Registerkarten finden Sie unter the section called “Was ist die vereinheitlichte Konsole?”.

6. Verwenden Sie die folgenden Optionen im Menü Knotenaktionen, um Aktionen auf dem Knoten auszuführen:

> **Note**
>
> Diese Aktionen sind nur für verwaltete Knoten in der Region AWS-Konto und der Region verfügbar, in der Sie gerade arbeiten. Für verwaltete Knoten, auf die Sie möglicherweise in anderen Konten oder Regionen Zugriff haben, können Sie stattdessen auf eine Eigenschaftenliste zugreifen.

- Terminalsitzung verbinden, starten – Stellen Sie eine Verbindung zum Knoten mit AWS Systems Manager Session Manager her.
- Tools
  - Dateisystem anzeigen – Durchsuchen Sie den Inhalt der Verzeichnisstruktur des Knotens. Verzeichnisse hinzufügen, umbenennen und entfernen. Dateien ausschneiden oder kopieren und einfügen.
  - Leistungsindikatoren anzeigen – Zeigt Leistungsinformationen über den Knoten an, z. B. CPU-Auslastung, Netzwerkverkehr und andere Nutzungsarten.

- Verwaltete Prozesse – Zeigt Informationen zur Ressourcennutzung auf dem Knoten an. Starten oder stoppen von Prozessen auf dem Knoten.
- Benutzer und Gruppen verwalten – Zeigen Sie Benutzerkonten und Benutzergruppen auf dem Knoten an, fügen Sie sie hinzu oder löschen Sie sie.
- Run-Befehl ausführen — Wird verwendetAWS Systems Manager Run Command, um die Konfiguration des Knotens zu verwalten. Run Commandverwendet Systems Manager Manager-Dokumente, um bei Bedarf Änderungen vorzunehmen, z. B. Anwendungen zu aktualisieren oder Linux-Shell-Skripts und PowerShell Windows-Befehle auszuführen.
- Patch-Knoten – Verwenden Sie das Feature Jetzt patchen in AWS Systems Manager Patch Manager, um von der Konsole aus einen On-Demand-Patching-Vorgang auf dem Knoten auszuführen.

> **Note**
>
> Die vorherigen Aufgaben können auch über das Menü Tools in der linken Navigationsleiste gestartet werden.

- Knoteneinstellungen
  - Tags hinzufügen – Wendet zusätzliche Tag-Schlüsselwertpaare auf den Knoten an.
  - Benutzerkennwort für den Knoten zurücksetzen – Legen Sie ein neues Passwort für einen bestimmten Benutzer auf dem Knoten fest.
  - IAM-Rolle ändern – Ändern Sie die IAM-Rolle, die dem Knoten zugeordnet ist. Erstellen Sie eine neue IAM-Rolle, die an den Knoten angefügt werden.

Bericht über verwaltete Knoten herunterladen oder exportieren

Sie können das Feature Knoten durchsuchen von Systems Manager verwenden, um gefilterte oder ungefilterte Listen verwalteter Knoten für Ihre AWS Organisation oder Ihr Konto in der Systems-Manager-Konsole anzuzeigen. In Fällen, in denen Sie die Daten offline anzeigen oder in einer anderen Anwendung verarbeiten möchten, können Sie den Bericht als `CSV`- oder `JSON`-Datei speichern.

Je nach Größe des Berichts werden Sie aufgefordert, den Bericht auf Ihren lokalen Computer herunterzuladen oder in einen Amazon-S3-Bucket zu exportieren. Berichte werden nur im `CSV`-Format in S3-Buckets gespeichert.

Um einen Bericht über verwaltete Knoten herunterzuladen oder zu exportieren

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Knoten erkunden aus.
3. (Optional) Gehen Sie wie unter Auswahl einer Filteransicht für Zusammenfassungen verwalteter Knoten beschrieben vor, um die Liste der verwalteten Knoten, die für Ihre Organisation oder Ihr Konto angezeigt wird, zu verfeinern.
4. Wählen Sie Bericht
   ( ).
5. Wenn das Dialogfeld Bericht herunterladen angezeigt wird, gehen Sie wie folgt vor:
   a. Geben Sie in das Feld Dateiname einen Namen für die Datei ein. Es wird empfohlen, einen Namen anzugeben, der den Umfang des Berichts repräsentiert, z. B. `all-organization-nodes` oder `ec2-instances-out-of-date-agent`.
   b. Geben Sie unter Eingeschlossene Spalten an, ob Spalten für alle verfügbaren Knotendetails oder nur die Spalten enthalten sein sollen, die Sie für Ihre aktuelle Anzeige ausgewählt haben.

      > **Tip**
      >
      > Informationen zur Verwaltung der Spalten in Ihrer Berichtsanzeige finden Sie unter Inhalt und Erscheinungsbild von Knotenberichten verwalten.

   c. Wählen Sie als Dateiformat CSV oder JSON aus, je nachdem, wie Sie die Datei verwenden möchten.
   d. Wählen Sie unter Tabellenüberschrift die Option Zeile mit Spaltennamen einbeziehen aus, um eine Zeile mit Spaltenüberschriften in eine `CSV`-Datei aufzunehmen.
   e. Wählen Sie Herunterladen aus.

   Der Bericht wird gemäß den Einstellungen Ihres Browsers am Standard-Download-Speicherort gespeichert.
6. Wenn das Dialogfeld Nach Amazon S3 exportieren angezeigt wird, gehen Sie wie folgt vor:
   a. Geben Sie für S3-URI den URI für den Bucket ein, in den der Bericht exportiert werden soll.

> **Tip**
>
> Wählen Sie Ansicht aus, um eine Liste Ihrer Buckets in der Amazon-S3-Konsole anzuzeigen. Um aus einer Liste von Buckets in Ihrem Konto auszuwählen, wählen Sie S3 durchsuchen.

b. Geben Sie unter Autorisierungsmethode die Servicerolle an, die verwendet werden soll, um Berechtigungen für den Export des Berichts in den Bucket zu erteilen.

   Wenn Sie sich dafür entscheiden, dass Systems Manager die Rolle für Sie erstellt, werden alle erforderlichen Berechtigungen und Vertrauenserklärungen für den Vorgang bereitgestellt.

   Wenn Sie Ihre eigene Rolle verwenden oder erstellen möchten, muss die Rolle die erforderlichen Berechtigungen und Vertrauenserklärungen enthalten. Weitere Informationen zum Erstellen dieser Rolle finden Sie unter Erstellen einer benutzerdefinierten Servicerolle zum Exportieren von Diagnoseberichten nach S3.

c. Wählen Sie Absenden aus.

Erstellen einer benutzerdefinierten Servicerolle zum Exportieren von Diagnoseberichten nach S3

Wenn Sie gefilterte oder ungefilterte Listen verwalteter Knoten für Ihre AWS -Organisation oder Ihr Konto auf der Seite Knoten durchsuchen von Systems Manager anzeigen, können Sie die Liste als Bericht als CSV-Datei in einen Amazon-S3-Bucket exportieren.

Dazu müssen Sie eine Servicerolle mit den erforderlichen Berechtigungen und Vertrauensrichtlinien für den Vorgang angeben. Sie können festlegen, dass Systems Manager die Rolle während des Herunterladens des Berichts für Sie erstellt. Optional können Sie die Rolle und die erforderliche Richtlinie selbst erstellen.

Erstellen einer benutzerdefinierten Servicerolle zum Exportieren von Diagnoseberichten nach S3

1. Folgen Sie den Schritten unter Richtlinien mithilfe des JSON-Editors erstellen im IAM-Benutzerhandbuch.

   - Verwenden Sie für den Inhalt der Richtlinie Folgendes und stellen Sie sicher, dass Sie ihn durch Ihre eigenen Informationen ersetzen. *placeholder values*

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketAcl",
        "s3:ListBucket",
        "s3:PutLifecycleConfiguration",
        "s3:GetLifecycleConfiguration"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:ListNodes"
      ],
      "Resource": "*"
    }
  ]
}
```

- Geben Sie der Richtlinie einen Namen, damit Sie sie im nächsten Schritt leichter erkennen können.

2. Folgen Sie den Schritten unter Erstellen einer IAM-Rolle mithilfe einer benutzerdefinierten Vertrauensrichtlinie (Konsole) im IAM-Benutzerhandbuch.

   - Geben Sie für Schritt 4 die folgende Vertrauensrichtlinie ein und achten Sie darauf, diese durch Ihre eigenen Informationen zu ersetzen. *placeholder values*

     JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "SSMAssumeRole",
      "Effect": "Allow",
      "Principal": {
        "Service": "ssm.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        }
      }
    }
  ]
}
```

3. Wählen Sie für Schritt 10 Schritt 2: Berechtigungen hinzufügen aus und wählen Sie den Namen der im vorherigen Schritt erstellten Richtlinie aus.

Nachdem Sie die Rolle erstellt haben, können Sie sie auswählen, wenn Sie die unter Bericht über verwaltete Knoten herunterladen oder exportieren beschriebenen Schritte ausführen.

Inhalt und Erscheinungsbild von Knotenberichten verwalten

Sie können das Feature Knoten durchsuchen von Systems Manager verwenden, um gefilterte oder ungefilterte Listen verwalteter Knoten für Ihre AWS Organisation oder Ihr Konto in der Systems-Manager-Konsole anzuzeigen. Sie können aus über einem Dutzend Feldern wählen, die Sie in Ihre Knotenlisten aufnehmen möchten, z. B. Knoten-ID, Betriebssystemname, Region und mehr. Sie

können auch die Spalten für Ihre Listen und Berichte neu anordnen und ändern, wie die Liste in der Konsole angezeigt wird.

So können Sie Inhalt und das Erscheinungsbild von Knotenberichten verwalten

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Knoten erkunden aus.
3. Klicken Sie auf den Bereich Knoten und wählen Sie das Zahnradsymbol ( ).
4. Gehen Sie im Dialogfeld Einstellungen wie folgt vor:
   a. Wählen Sie unter Seitengröße aus, wie viele Zeilen in jeder Konsolenansicht enthalten sein sollen: 10, 25 oder 50.
   b. Wählen Sie für Zeilenumbruch das Feld aus, um den gesamten Inhalt einer Zelle in der verfügbaren Spaltenbreite anzuzeigen.
   c. Wählen Sie für Gestreifte Zeilen das Feld aus, um abwechselnd Reihen mit durchsichtigem und schattiertem Hintergrund anzuzeigen.
   d. Gehen Sie für Sichtbaren Inhalt auswählen wie folgt vor:
      - Schalten Sie einzelne Spalten für Ihre Listenanzeige und Berichte ein oder aus.
      - Um die Reihenfolge der Spalten zu ändern, klicken Sie auf den Ziehgriff ( ) eines Spaltennamens, halten Sie ihn gedrückt und ziehen Sie ihn in der Liste nach oben oder unten.
5. Wählen Sie Bestätigen aus.

Just-in-Time-Knotenzugriff mit Systems Manager

Nutzen Sie den Just-in-Time-Zugriff von Systems Manager, um die Sicherheit Ihrer Knoten zu verbessern. Der Just-in-Time-Zugriff auf Knoten ermöglicht es Benutzern, temporären, zeitgebundenen Zugriff auf Knoten anzufordern, den Sie nur genehmigen können, wenn der Zugriff wirklich benötigt wird. Dadurch entfällt die Notwendigkeit, permanenten Zugriff auf Knoten zu gewähren, die durch IAM-Richtlinien verwaltet werden. Darüber hinaus bietet Systems Manager Sitzungsaufzeichnungen für RDP-Sitzungen auf Windows Server-Knoten, um Sie bei der Erfüllung

von Compliance-Anforderungen, der Durchführung von Ursachenanalysen und vielem mehr zu unterstützen. Um den Just-in-Time-Knotenzugriff zu nutzen, müssen Sie die vereinheitlichte Systems Manager-Konsole einrichten.

Mit dem Just-in-Time-Knotenzugriff erstellen Sie detaillierte IAM-Richtlinien, die dafür sorgen, dass nur die Benutzer, denen Sie die Genehmigung erteilen, Knotenzugriffsanforderungen stellen können. Anschließend erstellen Sie Genehmigungsrichtlinien, in denen die erforderlichen Genehmigungen für die Knotenverbindung definiert sind. Für den Just-in-Time-Knotenzugriff gibt es Richtlinien für die automatische Genehmigung und für die manuelle Genehmigung. Eine automatische Genehmigungsrichtlinie definiert, mit welchen Knoten Benutzer automatisch eine Verbindung herstellen können. Richtlinien für manuelle Genehmigungen definieren die Anzahl und die Stufen der manuellen Genehmigungen, die für den Zugriff auf die von Ihnen angegebenen Knoten erteilt werden müssen. Sie können auch eine Richtlinie zur Zugriffsverweigerung erstellen. Eine Zugriffsverweigerungsrichtlinie verhindert ausdrücklich die automatische Genehmigung von Zugriffsanforderungen an die von Ihnen angegebenen Knoten. Eine Zugriffsverweigerungsrichtlinie gilt für alle Konten in einer AWS Organizations-Organisation. Richtlinien für die automatische Genehmigung und die manuelle Genehmigung gelten nur für die AWS-Konten und AWS-Regionen an dem Ort, an dem sie erstellt wurden.

Wenn ein Benutzer versucht, eine Verbindung zu einem Knoten herzustellen, wird er aufgefordert, einen Grund für den Knotenzugriff einzugeben. Anschließend werden Ihre Genehmigungsrichtlinien ausgewertet. Abhängig von Ihren Richtlinien stellen Benutzer entweder automatisch eine Verbindung zum Zielknoten her oder Systems Manager erstellt automatisch eine manuelle Genehmigungsanforderung im Namen des Anforderers. Die Genehmiger, die in der für den Knoten geltenden Richtlinie für die manuelle Genehmigung angegeben sind, werden über die Zugriffsanforderung informiert und können die Anforderung genehmigen oder ablehnen. Genehmigende und Anforderer können entweder per E-Mail oder über die Amazon-Q-Developer-Integration in Chat-Anwendungen wie Slack oder Microsoft Teams benachrichtigt werden. Systems Manager gewährt nur dann Zugriff auf angeforderte Knoten, wenn die angegebenen Genehmiger alle erforderlichen Genehmigungen erteilt haben. Sobald alle erforderlichen Genehmigungen eingegangen sind, kann der Benutzer für die Dauer des in der Genehmigungsrichtlinie angegebenen Zugriffsfensters beliebig viele Sitzungen auf dem Knoten starten. Systems Manager beendet Just-in-Time-Knotenzugriffssitzungen nicht automatisch. Es hat sich bewährt, Werte für die maximale Sitzungsdauer und das Timeout bei Sitzungsinaktivität anzugeben. Diese Einstellungen verhindern, dass Benutzer auch nach Ablauf des Zeitraums, für den der Zugriff genehmigt wurde, mit Knoten verbunden bleiben.

Wir empfehlen, eine Kombination von Genehmigungsrichtlinien zu verwenden, um Knoten mit kritischeren Daten zu schützen und Benutzern gleichzeitig zu ermöglichen, sich ohne Eingreifen mit weniger kritischen Knoten zu verbinden. Sie können beispielsweise manuelle Genehmigungen für Zugriffsanforderungen an Datenbankknoten vorschreiben und Sitzungen für nicht-persistente Knoten der Präsentationsstufe automatisch genehmigen.

Systems Manager unterstützt Just-in-Time-Knotenzugriff für Benutzer, die mit IAM Identity Center oder IAM verbunden sind. Wenn ein Verbundbenutzer eine Zugriffsanforderung einreicht, gibt er den Zielknoten und den Grund an, warum eine Verbindung zum Knoten hergestellt werden muss. Systems Manager vergleicht die Benutzeridentität mit den Parametern, die in den Genehmigungsrichtlinien Ihrer Organisation definiert sind. Wenn die Richtlinienbedingungen für die automatische Genehmigung erfüllt sind oder Genehmigungsberechtigte Genehmigungen manuell erteilen, kann der Anforderer eine Verbindung zum Zielknoten herstellen. Wenn ein Benutzer versucht, eine Verbindung zu einem zugelassenen Knoten herzustellen, erstellt Systems Manager ein temporäres Token und verwendet es, um die Sitzung einzurichten.

Da der Systems Manager-Dienst die Authentifizierung für Zugriffsanforderungen und die Einrichtung von Sitzungen übernimmt, müssen Sie keine IAM-Richtlinien verwenden, um den Zugriff auf Ihre Knoten zu verwalten. Der Just-in-Time-Knotenzugriff von Systems Manager ist eine Möglichkeit, wie Ihre Organisation die permanente Vergabe von Zugriffsrechten vermeiden kann, da Sie Benutzern nur das Erstellen von Zugriffsanforderungen gestatten müssen, anstatt ihnen zu erlauben, Sitzungen mit dauerhaften Zugriffsrechten auf Ihre Knoten zu starten. Um Ihnen zu helfen, die Compliance-Anforderungen zu erfüllen, speichert Systems Manager alle Zugriffsanforderungen ein Jahr lang. Systems Manager versendet außerdem EventBridge-Ereignisse für den Just-in-Time-Knotenzugriff bei fehlgeschlagenen Zugriffsanforderungen und Statusaktualisierungen für Zugriffsanforderungen für manuelle Genehmigungen. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge.

Themen

- just-in-timeZugriff mit Systems Manager einrichten
- Starten Sie eine just-in-time Knotenzugriffssitzung
- Verwaltung von just-in-time Zugriffsanfragen
- Übergang zum just-in-time Knotenzugriff von Session Manager
- just-in-timeZugriff mit Systems Manager deaktivieren
- Just-in-time Häufig gestellte Fragen zum Knotenzugriff

just-in-timeZugriff mit Systems Manager einrichten

Die Einrichtung des just-in-time Knotenzugriffs mit Systems Manager umfasste mehrere Schritte. Zunächst wählen Sie die Ziele aus, für die Sie den just-in-time Knotenzugriff einrichten möchten. Ziele bestehen aus AWS Organizations Organisationseinheiten (OUs) und AWS-Regionen. Standardmäßig werden dieselben Ziele, die Sie bei der Einrichtung der einheitlichen Systems Manager Manager-Konsole ausgewählt haben, für den just-in-time Knotenzugriff ausgewählt. Sie können wählen, ob Sie den just-in-time Knotenzugriff für alle Ziele oder für eine Teilmenge der Ziele einrichten möchten, die Sie bei der Einrichtung der einheitlichen Systems Manager Manager-Konsole angegeben haben. Das Hinzufügen neuer Ziele, die bei der Einrichtung der einheitlichen Systems Manager Manager-Konsole nicht ausgewählt wurden, wird nicht unterstützt.

Als Nächstes erstellen Sie Genehmigungsrichtlinien, um festzulegen, wann Knotenverbindungen manuell genehmigt werden müssen und wann diese automatisch genehmigt werden. Genehmigungsrichtlinien werden von jedem Konto in Ihrer Organisation verwaltet. Sie können auch eine Richtlinie aus dem delegierten Administratorkonto teilen, um die automatische Genehmigung von Verbindungen zu bestimmten Knoten ausdrücklich zu verweigern.

> **Note**
>
> Die Einrichtung des just-in-time Knotenzugriffs hat keine Auswirkungen auf bestehende IAM-Richtlinien oder -Einstellungen, für Session Manager die Sie konfiguriert haben. Sie müssen die Genehmigung für die `StartSession` API-Aktion aus Ihren IAM-Richtlinien entfernen, um sicherzustellen, dass nur der just-in-time Knotenzugriff verwendet wird, wenn Benutzer versuchen, eine Verbindung zu Ihren Knoten herzustellen. Nachdem Sie den just-in-time Knotenzugriff eingerichtet haben, empfehlen wir, Ihre Genehmigungsrichtlinien mit einer Untergruppe von Benutzern und Knoten zu testen, um sicherzustellen, dass Ihre Richtlinien wie gewünscht funktionieren, bevor Sie die Berechtigungen für entfernen. Session Manager

Unterstützung der Authentifizierung

Beachten Sie die folgenden Details zur Authentifizierungsunterstützung für den just-in-time Knotenzugriff:

- Just-in-time Der Knotenzugriff unterstützt den Authentifizierungstyp Single Sign-On nicht, wenn eine Verbindung zu Windows Server Instanzen über Remote Desktop hergestellt wird.

- Nur AWS Security Token Service (AWS STS) `AssumeRole` temporäre Sicherheitsanmeldedaten werden unterstützt. Weitere Informationen finden Sie unter folgenden Themen im IAM-Benutzerhandbuch:
- • Temporäre Sicherheitsanmeldeinformationen in IAM
  - Anmeldeinformationen vergleichen AWS STS
  - Temporäre Sicherheitsnachweise anfordern

Die folgenden IAM-Richtlinien beschreiben die Berechtigungen, die für die Verwaltung erforderlich sind, und ermöglichen es Benutzern, mit Systems Manager just-in-time Knotenzugriffsanforderungen für Knoten zu erstellen. Nachdem Sie überprüft haben, dass Sie über die erforderlichen Berechtigungen für die Verwendung des just-in-time Knotenzugriffs mit Systems Manager verfügen, können Sie den Einrichtungsvorgang fortsetzen. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen.

IAM-Richtlinie zur Aktivierung just-in-time des Knotenzugriffs

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QuickSetupConfigurationManagers",
            "Effect": "Allow",
            "Action": [
                "ssm-quicksetup:CreateConfigurationManager",
                "ssm-quicksetup:DeleteConfigurationManager",
                "ssm-quicksetup:GetConfiguration",
                "ssm-quicksetup:GetConfigurationManager",
                "ssm-quicksetup:GetServiceSettings",
                "ssm-quicksetup:ListConfigurationManagers",
                "ssm-quicksetup:ListConfigurations",
                "ssm-quicksetup:ListQuickSetupTypes",
                "ssm-quicksetup:ListTagsForResource",
                "ssm-quicksetup:TagResource",
                "ssm-quicksetup:UntagResource",
                "ssm-quicksetup:UpdateConfigurationDefinition",
                "ssm-quicksetup:UpdateConfigurationManager",
                "ssm-quicksetup:UpdateServiceSettings"
```

```
            ],
            "Resource": "*"
        },
        {
            "Sid": "QuickSetupDeployments",
            "Effect": "Allow",
            "Action": [
                "cloudformation:DescribeStackSetOperation",
                "cloudformation:ListStacks",
                "cloudformation:DescribeStacks",
                "cloudformation:DescribeStackResources",
                "cloudformation:ListStackSetOperations",
                "cloudformation:ListStackInstances",
                "cloudformation:DescribeStackSet",
                "cloudformation:ListStackSets",
                "cloudformation:DescribeStackInstance",
                "cloudformation:DescribeOrganizationsAccess",
                "cloudformation:ActivateOrganizationsAccess",
                "cloudformation:GetTemplate",
                "cloudformation:ListStackSetOperationResults",
                "cloudformation:DescribeStackEvents",
                "cloudformation:UntagResource",
                "ssm:DescribeAutomationExecutions",
                "ssm:GetAutomationExecution",
                "ssm:ListAssociations",
                "ssm:DescribeAssociation",
                "ssm:GetDocument",
                "ssm:ListDocuments",
                "ssm:DescribeDocument",
                "ssm:GetOpsSummary",
                "organizations:DeregisterDelegatedAdministrator",
                "organizations:DescribeAccount",
                "organizations:DescribeOrganization",
                "organizations:ListDelegatedAdministrators",
                "organizations:ListRoots",
                "organizations:ListParents",
                "organizations:ListOrganizationalUnitsForParent",
                "organizations:DescribeOrganizationalUnit",
                "organizations:ListAWSServiceAccessForOrganization",
                "iam:ListRoles",
                "iam:ListRolePolicies",
                "iam:GetRole",
                "iam:CreatePolicy",
                "cloudformation:TagResource"
```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:RollbackStack",
                "cloudformation:CreateStack",
                "cloudformation:UpdateStack",
                "cloudformation:DeleteStack"
            ],
            "Resource": [
                "arn:aws:cloudformation:*:*:stack/StackSet-AWS-QuickSetup-
JITNA*",
                "arn:aws:cloudformation:*:*:stack/AWS-QuickSetup-*",
                "arn:aws:cloudformation:*:*:type/resource/*",
                "arn:aws:cloudformation:*:*:stack/StackSet-SSMQuickSetup"
            ]
        },
        {
            "Sid": "StackSetOperations",
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateStackSet",
                "cloudformation:UpdateStackSet",
                "cloudformation:DeleteStackSet",
                "cloudformation:DeleteStackInstances",
                "cloudformation:CreateStackInstances",
                "cloudformation:StopStackSetOperation"
            ],
            "Resource": [
                "arn:aws:cloudformation:*:*:stackset/AWS-QuickSetup-JITNA*",
                "arn:aws:cloudformation:*:*:type/resource/*",
                "arn:aws:cloudformation:*:*:stackset-target/AWS-QuickSetup-
JITNA*:*"
            ]
        },
        {
            "Sid": "IamRolesMgmt",
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:GetRole",
```

```
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy",
                "iam:DetachRolePolicy",
                "iam:GetRolePolicy",
                "iam:ListRolePolicies"
            ],
            "Resource": [
                "arn:aws:iam::*:role/AWS-QuickSetup-JITNA*",
                "arn:aws:iam::*:role/service-role/AWS-QuickSetup-JITNA*"
            ]
        },
        {
            "Sid": "IamPassRole",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/AWS-QuickSetup-JITNA*",
                "arn:aws:iam::*:role/service-role/AWS-QuickSetup-JITNA*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ssm.amazonaws.com",
                        "ssm-quicksetup.amazonaws.com",
                        "cloudformation.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "SSMAutomationExecution",
            "Effect": "Allow",
            "Action": "ssm:StartAutomationExecution",
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/AWS-EnableExplorer",
                "arn:aws:ssm:us-east-1:111122223333:automation-execution/*"
            ]
        },
        {
            "Sid": "SSMAssociationPermissions",
            "Effect": "Allow",
            "Action": [
```

```
                "ssm:DeleteAssociation",
                "ssm:CreateAssociation",
                "ssm:StartAssociationsOnce"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:association/*"
        },
        {
            "Sid": "SSMResourceDataSync",
            "Effect": "Allow",
            "Action": [
                "ssm:CreateResourceDataSync",
                "ssm:UpdateResourceDataSync"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:resource-data-sync/
AWS-QuickSetup-*"
        },
        {
            "Sid": "ListResourceDataSync",
            "Effect": "Allow",
            "Action": [
                "ssm:ListResourceDataSync"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CreateServiceLinkedRoles",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "accountdiscovery.ssm.amazonaws.com",
                        "ssm.amazonaws.com",
                        "ssm-quicksetup.amazonaws.com",
                        "stacksets.cloudformation.amazonaws.com"
                    ]
                }
            },
            "Resource": "*"
        },
        {
            "Sid": "CreateStackSetsServiceLinkedRole",
```

```
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-
service-role/stacksets.cloudformation.amazonaws.com/
AWSServiceRoleForCloudFormationStackSetsOrgAdmin"
        },
        {
            "Sid": "AllowSsmJitnaPoliciesCrudOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:CreateDocument",
                "ssm:UpdateDocument",
                "ssm:UpdateDocumentDefaultVersion",
                "ssm:GetDocument",
                "ssm:DescribeDocument",
                "ssm:DeleteDocument"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/SSM-
JustInTimeAccessDenyAccessOrgPolicy"
            ],
            "Condition": {
                "StringEquals": {
                    "ssm:DocumentType": [
                        "AutoApprovalPolicy"
                    ]
                }
            }
        },
        {
            "Sid": "AllowAccessRequestOpsItemOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:GetOpsItem",
                "ssm:DescribeOpsItems",
                "ssm:GetOpsSummary",
                "ssm:DeleteOpsItem",
                "ssm:ListOpsItemEvents"
            ],
            "Resource": "*"
        },
        {
```

```
            "Sid": "IdentityCenterPermissions",
            "Effect": "Allow",
            "Action": [
                "sso:DescribeRegisteredRegions",
                "sso:ListDirectoryAssociations",
                "identitystore:GetUserId",
                "identitystore:DescribeUser",
                "identitystore:DescribeGroup",
                "identitystore:ListGroupMembershipsForMember"
            ],
            "Resource": "*"
        }
    ]
}
```

IAM-Richtlinie für die Konfiguration des Knotenzugriffs just-in-time

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSsmJitnaPoliciesCrudOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:CreateDocument",
                "ssm:UpdateDocument",
                "ssm:UpdateDocumentDefaultVersion",
                "ssm:GetDocument",
                "ssm:DescribeDocument",
                "ssm:DeleteDocument"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ssm:DocumentType": [
                        "ManualApprovalPolicy",
                        "AutoApprovalPolicy"
                    ]
```

```
                }
            }
        },
        {
            "Sid": "AllowSsmJitnaPoliciesListOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:ListDocuments",
                "ssm:ListDocumentVersions"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/SSM-
JustInTimeAccessTokenRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "justintimeaccess.ssm.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "AllowAccessRequestOpsItemOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:GetOpsItem",
                "ssm:DescribeOpsItems",
                "ssm:GetOpsSummary",
                "ssm:DeleteOpsItem",
                "ssm:ListOpsItemEvents"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowSessionManagerPreferencesOperation",
            "Effect": "Allow",
            "Action": [
                "ssm:CreateDocument",
                "ssm:GetDocument",
                "ssm:DescribeDocument",
```

```
                "ssm:UpdateDocument",
                "ssm:DeleteDocument"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell",
            "Condition": {
                "StringEquals": {
                    "ssm:DocumentType": "Session"
                }
            }
        },
        {
            "Sid": "AllowSessionManagerOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeSessions",
                "ssm:GetConnectionStatus",
                "ssm:TerminateSession"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowRDPConnectionRecordingOperations",
            "Effect": "Allow",
            "Action": [
                "ssm-guiconnect:UpdateConnectionRecordingPreferences",
                "ssm-guiconnect:GetConnectionRecordingPreferences",
                "ssm-guiconnect:DeleteConnectionRecordingPreferences"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowRDPConnectionRecordingKmsOperation",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/SystemsManagerJustInTimeNodeAccessManaged":
 "true"
                },
                "StringLike": {
```

```
                    "kms:ViaService": "ssm-guiconnect.*.amazonaws.com"
                },
                "Bool": {
                    "aws:ViaAWSService": "true"
                }
            }
        },
        {
            "Sid": "AllowFleetManagerOperations",
            "Effect": "Allow",
            "Action": [
                "ssm-guiconnect:GetConnection",
                "ssm-guiconnect:ListConnections"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SNSTopicManagement",
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:SetTopicAttributes"
            ],
            "Resource": [
                "arn:aws:sns:us-east-1:111122223333:SSM-JITNA*"
            ]
        },
        {
            "Sid": "SNSListTopics",
            "Effect": "Allow",
            "Action": [
                "sns:ListTopics"
            ],
            "Resource": "*"
        },
        {
            "Sid": "EventBridgeRuleManagement",
            "Effect": "Allow",
            "Action": [
                "events:PutRule",
                "events:PutTargets"
            ],
            "Resource": [
                "arn:aws:events:us-east-1::rule/SSM-JITNA*"
```

```
            ]
        },
        {
            "Sid": "ChatbotSlackManagement",
            "Effect": "Allow",
            "Action": [
                "chatbot:CreateSlackChannelConfiguration",
                "chatbot:UpdateSlackChannelConfiguration",
                "chatbot:DescribeSlackChannelConfigurations",
                "chatbot:DescribeSlackWorkspaces",
                "chatbot:DeleteSlackChannelConfiguration",
                "chatbot:RedeemSlackOauthCode",
                "chatbot:DeleteSlackWorkspaceAuthorization",
                "chatbot:GetSlackOauthParameters"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ChatbotTeamsManagement",
            "Effect": "Allow",
            "Action": [
                "chatbot:ListMicrosoftTeamsChannelConfigurations",
                "chatbot:CreateMicrosoftTeamsChannelConfiguration",
                "chatbot:UpdateMicrosoftTeamsChannelConfiguration",
                "chatbot:ListMicrosoftTeamsConfiguredTeams",
                "chatbot:DeleteMicrosoftTeamsChannelConfiguration",
                "chatbot:RedeemMicrosoftTeamsOauthCode",
                "chatbot:DeleteMicrosoftTeamsConfiguredTeam",
                "chatbot:GetMicrosoftTeamsOauthParameters",
                "chatbot:TagResource"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SSMEmailSettings",
            "Effect": "Allow",
            "Action": [
                "ssm:UpdateServiceSetting",
                "ssm:GetServiceSetting"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:servicesetting/ssm/access-
request/email-role-mapping",
```

```
                "arn:aws:ssm:us-east-1:111122223333:servicesetting/ssm/access-
request/enabled-email-notifications"
            ]
        },
        {
            "Sid": "AllowViewingJitnaCloudWatchMetrics",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "AWS/SSM/JustInTimeAccess"
                }
            }
        },
        {
            "Sid": "QuickSetupConfigurationManagers",
            "Effect": "Allow",
            "Action": [
                "ssm-quicksetup:ListConfigurationManagers",
                "ssm-quicksetup:ListConfigurations",
                "ssm-quicksetup:ListQuickSetupTypes",
                "ssm-quicksetup:GetConfiguration",
                "ssm-quicksetup:GetConfigurationManager"
            ],
            "Resource": "*"
        },
        {
            "Sid": "QuickSetupDeployments",
            "Effect": "Allow",
            "Action": [
                "cloudformation:ListStacks",
                "cloudformation:DescribeStacks",
                "organizations:DescribeOrganization",
                "organizations:ListDelegatedAdministrators"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ManualPolicy",
```

```
            "Effect": "Allow",
            "Action": [
                "sso:DescribeRegisteredRegions",
                "ssm:GetServiceSetting",
                "iam:ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SessionPreference",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowIamListForKMS",
            "Effect": "Allow",
            "Action": [
                "iam:ListUsers"
            ],
            "Resource": "arn:aws:iam::111122223333:user/*"
        },
        {
            "Sid": "KMSPermission",
            "Effect": "Allow",
            "Action": [
                "kms:TagResource",
                "kms:ListAliases",
                "kms:CreateAlias"
            ],
            "Resource": "*"
        },
        {
            "Sid": "KMSCreateKey",
            "Effect": "Allow",
            "Action": [
                "kms:CreateKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
```

```
                    "aws:RequestTag/SystemsManagerJustInTimeNodeAccessManaged":
 "true"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "SystemsManagerJustInTimeNodeAccessManaged"
                    ]
                }
            }
        },
        {
            "Sid": "AllowIamRoleForChatbotAction",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/role name",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "chatbot.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "AllowIamServiceRoleForChat",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/
management.chatbot.amazonaws.com/AWSServiceRoleForAWSChatbot"
        },
        {
            "Sid": "CloudWatchLogs",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": "arn:aws:logs:*:111122223333:log-group::log-stream:"
        },
        {
            "Sid": "IdentityStorePermissions",
```

```
            "Effect": "Allow",
            "Action": [
                "sso:ListDirectoryAssociations",
                "identitystore:GetUserId",
                "sso-directory:SearchUsers",
                "sso-directory:SearchGroups",
                "identitystore:DescribeGroup",
                "identitystore:DescribeUser"
            ],
            "Resource": "*"
        }
    ]
}
```

IAM-Richtlinie für Genehmiger von Zugriffsanforderungen

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccessRequestDescriptions",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeOpsItems",
                "ssm:GetOpsSummary",
                "ssm:ListOpsItemEvents"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowGetSpecificAccessRequest",
            "Effect": "Allow",
            "Action": [
                "ssm:GetOpsItem"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:opsitem/*"
        },
        {
            "Sid": "AllowApprovalRejectionSignal",
            "Effect": "Allow",
```

```
            "Action": [
                "ssm:SendAutomationSignal"
            ],
            "Resource": "arn:aws:ssm:*:*:automation-execution/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/SystemsManagerJustInTimeNodeAccessManaged":
"true"
                }
            }
        },
        {
            "Sid": "QuickSetupConfigurationManagers",
            "Effect": "Allow",
            "Action": [
                "ssm-quicksetup:ListConfigurationManagers",
                "ssm-quicksetup:ListConfigurations",
                "ssm-quicksetup:GetConfigurationManager",
                "ssm-quicksetup:ListQuickSetupTypes",
                "ssm-quicksetup:GetConfiguration"
            ],
            "Resource": "*"
        },
        {
            "Sid": "QuickSetupDeployments",
            "Effect": "Allow",
            "Action": [
                "cloudformation:ListStacks",
                "cloudformation:DescribeStacks",
                "organizations:DescribeOrganization",
                "organizations:ListDelegatedAdministrators"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowSsmJitnaPoliciesCrudOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:GetDocument",
                "ssm:DescribeDocument"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/*"
            ],
```

```
            "Condition": {
                "StringEquals": {
                    "ssm:DocumentType": [
                        "ManualApprovalPolicy",
                        "AutoApprovalPolicy"
                    ]
                }
            }
        },
        {
            "Sid": "AllowSsmJitnaPoliciesListOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:ListDocuments",
                "ssm:ListDocumentVersions"
            ],
            "Resource": "*"
        },
        {
            "Sid": "IDCPermissions",
            "Effect": "Allow",
            "Action": [
                "sso:DescribeRegisteredRegions",
                "sso:ListDirectoryAssociations",
                "identitystore:GetUserId",
                "identitystore:DescribeUser",
                "identitystore:DescribeGroup",
                "identitystore:ListGroupMembershipsForMember"
            ],
            "Resource": "*"
        }
    ]
}
```

IAM-Richtlinie für Benutzer mit just-in-time Knotenzugriff

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "AllowJITNAOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:StartAccessRequest",
                "ssm:GetAccessToken"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowOpsItemCreationAndRetrieval",
            "Effect": "Allow",
            "Action": [
                "ssm:CreateOpsItem",
                "ssm:GetOpsItem"
            ],
            "Resource": "arn:aws:ssm:*:*:opsitem/*"
        },
        {
            "Sid": "AllowListAccessRequests",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeOpsItems",
                "ssm:GetOpsSummary",
                "ssm:ListOpsItemEvents",
                "ssm:DescribeSessions"
            ],
            "Resource": "*"
        },
        {
            "Sid": "RequestManualApprovals",
            "Action": "ssm:StartAutomationExecution",
            "Effect": "Allow",
            "Resource": "arn:aws:ssm:*:*:document/*",
            "Condition": {
                "StringEquals": {
                    "ssm:DocumentType": "ManualApprovalPolicy"
                }
            }
        },
        {
            "Sid": "StartManualApprovalsAutomationExecution",
            "Effect": "Allow",
            "Action": "ssm:StartAutomationExecution",
            "Resource": "arn:aws:ssm:*:*:automation-execution/*"
```

```
        },
        {
            "Sid": "CancelAccessRequestManualApproval",
            "Effect": "Allow",
            "Action": "ssm:StopAutomationExecution",
            "Resource": "arn:aws:ssm:*:*:automation-execution/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/SystemsManagerJustInTimeNodeAccessManaged":
"true"
                }
            }
        },
        {
            "Sid": "DescribeEC2Instances",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeTags",
                "ec2:GetPasswordData"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowListSSMManagedNodesAndTags",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeInstanceInformation",
                "ssm:ListTagsForResource"
            ],
            "Resource": "*"
        },
        {
            "Sid": "QuickSetupConfigurationManagers",
            "Effect": "Allow",
            "Action": [
                "ssm-quicksetup:ListConfigurationManagers",
                "ssm-quicksetup:GetConfigurationManager",
                "ssm-quicksetup:ListConfigurations",
                "ssm-quicksetup:ListQuickSetupTypes",
                "ssm-quicksetup:GetConfiguration"
            ],
            "Resource": "*"
        },
```

```
        {
            "Sid": "AllowSessionManagerOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeSessions",
                "ssm:GetConnectionStatus"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowRDPOperations",
            "Effect": "Allow",
            "Action": [
                "ssm-guiconnect:ListConnections",
                "ssm:GetConnectionStatus"
            ],
            "Resource": "*"
        },
        {
            "Sid": "QuickSetupDeployments",
            "Effect": "Allow",
            "Action": [
                "cloudformation:ListStacks",
                "cloudformation:DescribeStacks",
                "organizations:DescribeOrganization",
                "organizations:ListDelegatedAdministrators"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowSsmJitnaPoliciesReadOnly",
            "Effect": "Allow",
            "Action": [
                "ssm:GetDocument",
                "ssm:DescribeDocument"
            ],
            "Resource": [
                "arn:aws:ssm:*:111122223333:document/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ssm:DocumentType": [
                        "ManualApprovalPolicy",
                        "AutoApprovalPolicy"
```

```
                    ]
                }
            }
        },
        {
            "Sid": "AllowSsmJitnaPoliciesListOperations",
            "Effect": "Allow",
            "Action": [
                "ssm:ListDocuments",
                "ssm:ListDocumentVersions"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ExploreNodes",
            "Effect": "Allow",
            "Action": [
                "ssm:ListNodesSummary",
                "ssm:ListNodes",
                "ssm:DescribeInstanceProperties"
            ],
            "Resource": "*"
        },
        {
            "Sid": "IdentityStorePermissions",
            "Effect": "Allow",
            "Action": [
                "sso:DescribeRegisteredRegions",
                "sso:ListDirectoryAssociations",
                "identitystore:GetUserId",
                "identitystore:DescribeUser",
                "identitystore:DescribeGroup"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Um den Zugriff auf API-Operationen einzuschränken, die Genehmigungsrichtlinien erstellen, aktualisieren oder löschen, verwenden Sie den Bedingungsschlüssel `ssm:DocumentType`

für die Dokumenttypen `AutoApprovalPolicy` und `ManualApprovalPolicy` . Die API-Vorgänge `StartAccessRequest` und `GetAccessToken` unterstützen die folgenden globalen Kontextschlüssel nicht:

- `aws:SourceVpc`
- `aws:SourceVpce`
- `aws:VpcSourceIp`
- `aws:UserAgent`
- `aws:MultiFactorAuthPresent`

Weitere Informationen zu Bedingungskontextschlüsseln für Systems Manager finden Sie unter Bedingungsschlüssel für AWS Systems Manager in der Service Authorization-Referenz.

Das folgende Verfahren beschreibt, wie der erste Einrichtungsschritt für den just-in-time Knotenzugriff abgeschlossen wird.

Um den just-in-time Knotenzugriff einzurichten

1. Melden Sie sich beim delegierten Systems Manager-Administratorkonto für Ihre Organisation an.
2. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
3. Wählen Sie im Navigationsbereich Just-in-time Knotenzugriff aus.
4. Wählen Sie Das neue Erlebnis aktivieren aus.
5. Wählen Sie die Regionen aus, in denen Sie den just-in-time Knotenzugriff aktivieren möchten. Standardmäßig werden dieselben Regionen, die Sie bei der Einrichtung der einheitlichen Systems Manager Manager-Konsole ausgewählt haben, für den just-in-time Knotenzugriff ausgewählt. Die Auswahl neuer Regionen, die bei der Einrichtung der vereinheitlichten Systems-Manager-Konsole nicht bereits ausgewählt wurden, ist nicht möglich.
6. Wählen Sie just-in-timeKnotenzugriff aktivieren aus.

Die Nutzung des just-in-time Knotenzugriffs für 30 Tage nach Aktivierung der Funktion ist kostenlos. Nach Ablauf der 30-tägigen Testphase ist die Nutzung des just-in-time Node Access kostenpflichtig. Weitere Informationen finden Sie unter AWS Systems Manager - Preise.

Erstellen Sie Genehmigungsrichtlinien für Ihre Knoten

Genehmigungsrichtlinien definieren, welche Genehmigungen Benutzer für den Zugriff auf einen Knoten benötigen. Da just-in-time beim Knotenzugriff keine seit langem bestehenden Berechtigungen für Knoten über IAM-Richtlinien erforderlich sind, müssen Sie Genehmigungsrichtlinien erstellen, um den Zugriff auf Ihre Knoten zu ermöglichen. Wenn für einen Knoten keine Genehmigungsrichtlinien gelten, können Benutzer keinen Zugriff auf den Knoten anfordern.

Beim just-in-time Knotenzugriff gibt es drei Arten von Richtlinien. Die Richtlinientypen sind automatische Genehmigung, Zugriffsverweigerung und manuelle Genehmigung.

Just-in-time Richtlinientypen für den Knotenzugriff

- Eine automatische Genehmigungsrichtlinie definiert, mit welchen Knoten Benutzer automatisch eine Verbindung herstellen können.
- Richtlinien für manuelle Genehmigungen definieren die Anzahl und die Stufen der manuellen Genehmigungen, die für den Zugriff auf die von Ihnen angegebenen Knoten erteilt werden müssen.
- Eine Zugriffsverweigerungsrichtlinie verhindert ausdrücklich die automatische Genehmigung von Zugriffsanforderungen an die von Ihnen angegebenen Knoten.

Eine Richtlinie zur Zugriffsverweigerung gilt für alle Konten in einer AWS Organizations Organisation. Beispielsweise könnten Sie ausdrücklich automatische Genehmigungen für die `Intern`-Gruppe für Knoten verweigern, die mit dem `Production`-Schlüssel gekennzeichnet sind. Richtlinien für die automatische Genehmigung und die manuelle Genehmigung gelten nur für die AWS-Konten und AWS-Regionen wo sie erstellt wurden. Jedes Mitgliedskonto Ihrer Organisation verwaltet seine eigenen Genehmigungsrichtlinien. Genehmigungsrichtlinien werden in der folgenden Reihenfolge ausgewertet:

1. Zugriffsverweigerung
2. Automatische Genehmigung
3. Manuell

Sie können zwar nur eine Zugriffsverweigerungsrichtlinie pro Organisation und eine automatische Genehmigungsrichtlinie pro Konto und Region einrichten, aber Sie werden wahrscheinlich mehrere Richtlinien für manuelle Genehmigungen in einem Konto haben. Bei der Bewertung manueller Genehmigungsrichtlinien wird beim just-in-time Knotenzugriff immer die spezifischere Richtlinie

für einen Knoten bevorzugt. Richtlinien für die manuelle Genehmigung werden in der folgenden Reihenfolge ausgewertet:

1. Spezifisches Tag
2. Alle Knoten

Sie haben beispielsweise einen Knoten, der mit dem `Demo`-Schlüssel markiert ist. In demselben Konto gibt es eine Richtlinie für manuelle Genehmigungen, die auf alle Knoten abzielt und eine Genehmigung von einer Ebene erfordert. Sie haben auch eine Richtlinie zur manuellen Genehmigung, die zwei Genehmigungen von zwei Ebenen für Knoten erfordert, die mit dem `Demo`-Schlüssel gekennzeichnet sind. Systems Manager wendet die Richtlinie, die auf das `Demo`-Tag abzielt, auf den Knoten an, da sie spezifischer ist als die Richtlinie, die auf alle Knoten abzielt. Auf diese Weise können Sie eine allgemeine Richtlinie für alle Knoten in Ihrem Konto erstellen, um sicherzustellen, dass Benutzer Zugriffsanforderungen stellen können, und Sie gleichzeitig bei Bedarf detailliertere Richtlinien erstellen können.

Je nach Ihrer Organisation werden Ihren Knoten möglicherweise mehrere Tags zugewiesen. Wenn in diesem Szenario mehrere Richtlinien zur manuellen Genehmigung für einen Knoten gelten, schlagen Zugriffsanforderungen fehl. Ein Knoten ist beispielsweise mit den Schlüsseln `Production` und `Database` gekennzeichnet. In demselben Konto haben Sie eine Richtlinie zur manuellen Genehmigung, die für Knoten gilt, die mit dem `Production`-Schlüssel markiert sind, und eine weitere Richtlinie zur manuellen Genehmigung, die für Knoten gilt, die mit dem `Database`-Schlüssel markiert sind. Dies führt zu einem Konflikt für den Knoten, der mit beiden Schlüsseln markiert ist, und die Zugriffsanforderungen schlagen fehl. Systems Manager leitet den Benutzer zu der fehlgeschlagenen Anforderung weiter. Dort können sie Details zu den widersprüchlichen Richtlinien und Tags einsehen, sodass sie die erforderlichen Anpassungen vornehmen können, sofern sie über die erforderlichen Berechtigungen verfügen. Andernfalls können sie einen Kollegen in ihrer Organisation benachrichtigen, der über die erforderlichen Berechtigungen verfügt, um die Richtlinien zu ändern. Richtlinienkonflikte, die zu fehlgeschlagenen Zugriffsanfragen führen, lösen EventBridge Ereignisse aus, sodass Sie flexibel Ihre eigenen Antwort-Workflows erstellen können. Darüber hinaus sendet Systems Manager E-Mail-Benachrichtigungen für Richtlinienkonflikte, die zu fehlgeschlagenen Zugriffsanforderungen führen, an die von Ihnen angegebenen Empfänger. Weitere Informationen zur Konfiguration von E–Mail-Benachrichtigungen für Richtlinienkonflikte finden Sie unter Benachrichtigungen für just-in-time Zugriffsanfragen konfigurieren.

In einer Zugriffsverweigerungsrichtlinie verwenden Sie die Cedar-Richtliniensprache, um zu definieren, mit welchen Knoten Benutzer in Ihrer Organisation explizit nicht automatisch eine

Verbindung herstellen können. Diese Richtlinie wird über das delegierte Administratorkonto für Ihre Organisation erstellt und gemeinsam genutzt. Die Richtlinie „Zugriff verweigern“ hat Vorrang vor allen Richtlinien für die automatische Genehmigung. Sie können pro Organisation nur eine Zugriffsverweigerungsrichtlinie einrichten.

In einer Richtlinie für die automatische Genehmigung verwenden Sie die Cedar-Richtliniensprache, um zu definieren, welche Benutzer ohne manuelle Genehmigung automatisch eine Verbindung zu den angegebenen Knoten herstellen können. Die Zugriffsdauer für eine Zugriffsanforderung, die automatisch genehmigt wird, beträgt 1 Stunde. Dieser Wert kann nicht geändert werden. Sie können nur eine Richtlinie für die automatische Genehmigung pro Konto und Region einrichten.

In einer manuellen Genehmigungsrichtlinie geben Sie die Zugriffsdauer, die Anzahl der erforderlichen Genehmigungsebenen, die Anzahl der pro Ebene erforderlichen Genehmiger und die Knoten an, für die sie just-in-time Zugriffsanfragen genehmigen können. Die Zugriffsdauer für eine Richtlinie zur manuellen Genehmigung muss zwischen 1 und 336 Stunden liegen. Wenn Sie mehrere Genehmigungsebenen angeben, werden die Genehmigungen für die Zugriffsanforderung Ebene für Ebene verarbeitet. Das bedeutet, dass alle Genehmigungen, die Sie für eine Ebene benötigen, erteilt werden müssen, bevor der Genehmigungsprozess auf nachfolgende Ebenen übergeht. Wenn Sie in einer Richtlinie zur manuellen Genehmigung mehrere Tags angeben, werden diese als `or`-Anweisungen und nicht `and`-Anweisungen bewertet. Wenn Sie beispielsweise eine Richtlinie zur manuellen Genehmigung erstellen, die die Tags`Application`, `Web` und `Test` enthält, gilt die Richtlinie für jeden Knoten, der mit einem dieser Schlüssel markiert ist. Die Richtlinie gilt nicht nur für Knoten, die mit allen drei Schlüsseln markiert sind.

Wir empfehlen, eine Kombination aus manuellen Richtlinien mit Ihrer Richtlinie zur automatischen Genehmigung zu verwenden, um Knoten mit kritischeren Daten zu schützen und Benutzern gleichzeitig zu ermöglichen, sich ohne Eingreifen mit weniger kritischen Knoten zu verbinden. Sie können beispielsweise manuelle Genehmigungen für Zugriffsanforderungen an Datenbankknoten vorschreiben und Sitzungen für nicht-persistente Knoten der Präsentationsstufe automatisch genehmigen.

In den folgenden Verfahren wird beschrieben, wie Genehmigungsrichtlinien für den just-in-time Knotenzugriff erstellt werden.

Themen

- Manuelle Genehmigungsrichtlinien für den just-in-time Knotenzugriff erstellen
- Struktur der Anweisungen und integrierte Operatoren für Richtlinien zur automatischen Genehmigung und Zugriffsverweigerung

- Erstellen Sie eine automatische Genehmigungsrichtlinie für den just-in-time Knotenzugriff
- Erstellen Sie eine Deny-Access-Richtlinie für just-in-time den Knotenzugriff
- Erstellen Sie Genehmigungsrichtlinien für den just-in-time Knotenzugriff mit Amazon Q

Manuelle Genehmigungsrichtlinien für den just-in-time Knotenzugriff erstellen

Im folgenden Verfahren wird das Erstellen von Richtlinien zur manuellen Genehmigung beschrieben. Mit Systems Manager können Sie bis zu 50 manuelle Genehmigungsrichtlinien pro AWS-Konto und erstellen AWS-Region.

Erstellen Sie eine Richtlinie zur manuellen Genehmigung wie folgt:

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Knotenzugriff verwalten aus.
3. Geben Sie im Abschnitt Richtliniendetails des Schritts Richtlinie zur manuellen Genehmigung erstellen einen Namen und eine Beschreibung für die Genehmigungsrichtlinie ein.
4. Geben Sie einen Wert für die Zugriffsdauer ein. Dies ist die maximale Zeitspanne, für die ein Benutzer Sitzungen zu einem Knoten starten kann, nachdem eine Zugriffsanforderung genehmigt wurde. Dieser Wert muss zwischen 1 und 336 Stunden liegen.
5. Geben Sie im Abschnitt Knotenziele Tag-Schlüssel-Wert-Paare ein, die den Knoten zugeordnet sind, für die Sie die Richtlinie anwenden möchten. Wenn keines der in der Richtlinie angegebenen Tags einem Knoten zugeordnet ist, wird die Richtlinie nicht auf den Knoten angewendet.
6. Geben Sie im Abschnitt Genehmiger für Zugriffsanforderungen die Benutzer oder Gruppen ein, die Zugriffsanforderungen für die Knotenziele in der Richtlinie genehmigen sollen. Genehmiger von Zugriffsanforderungen können Benutzer und Gruppen von IAM Identity Center oder IAM-Rollen sein. Sie können bis zu 5 Genehmiger pro Ebene und bis zu 5 Genehmigerebenen angeben.
7. Wählen Sie Richtlinie zur manuellen Genehmigung erstellen aus.

Struktur der Anweisungen und integrierte Operatoren für Richtlinien zur automatischen Genehmigung und Zugriffsverweigerung

Die folgende Tabelle enthält die Struktur der Richtlinien für automatische Genehmigung und Zugriffsverweigerung.

Komponente	Syntax
Auswirkung	`permit | forbid`
scope	`(principal, action, resource)`
Bedingungsklausel	`when { principal or resource has attribute name };`

Richtlinienkomponenten

Eine Richtlinie für automatische Genehmigung oder Zugriffsverweigerung umfasst die folgenden Komponenten:

- Effekt: entweder `permit` (zulassen) oder `forbid` (verweigern).
- Umfang: Die Prinzipale, Aktionen und Ressourcen, für die der Effekt gilt. Sie können den Umfang in Cedar undefiniert lassen, indem Sie keine Prinzipale, Aktionen oder Ressourcen angeben. Die Richtlinie gilt dann für alle möglichen Prinzipale, Aktionen und Ressourcen. Für den Zugriff auf just-in-time Knoten gilt das `action` immer. `AWS::SSM::Action::"getTokenForInstanceAccess"`
- Bedingungsklausel: Der Kontext, in dem der Effekt gilt.

Kommentare

Sie können Kommentare in Ihre Richtlinien einfügen. Kommentare sind als Zeilen definiert, die mit `//` beginnen und mit einem Zeilenumbruchzeichen enden.

Das folgende Beispiel zeigt Kommentare in einer Richtlinie:

```
// Allows users in the Engineering group from the Platform org to automatically connect
 to nodes tagged with Engineering and Production keys.
permit (
    principal in AWS::IdentityStore::Group::"d4q81745-r081-7079-d789-14da1EXAMPLE",
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
```

```
when {
    principal has organization && resource.hasTag("Engineering") &&
 resource.hasTag("Production") && principal.organization == "Platform"
};
```

Mehrere Klauseln

Mit dem Operator && können Sie in einer Richtlinienanweisung mehrere Bedingungsklauseln angeben.

```
// Allow access if node has tag where the tag key is Environment
// & tag value is Development

permit(principal, action == AWS::SSM::getTokenForInstanceAccess, resource)
when {
    resource.hasTag("Environment") &&
    resource.getTag("Environment") == "Development"
};
```

Reservierte Zeichen

Das folgende Beispiel zeigt, wie eine Richtlinie geschrieben wird, wenn eine Kontexteigenschaft ein : (Semikolon) verwendet – ein reserviertes Zeichen in der Richtliniensprache.

```
permit (
    principal,
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {
    principal has employeeNumber && principal.employeeNumber like "E-1*" &&
 resource.hasTag("Purpose") && resource.getTag("Purpose") == "Testing"
}
```

Weitere Beispiele finden Sie unter Beispiele für Richtlinienanweisungen.

Just-in-time Schema für den Knotenzugriff

Das Folgende ist das Cedar-Schema für den just-in-time Knotenzugriff.

```
namespace AWS::EC2 {
    entity Instance tags String;
}
```

```
namespace AWS::IdentityStore {
    entity Group;

    entity User in [Group] {
    employeeNumber?: String,
    costCenter?: String,
    organization?: String,
    division?: String,
    };

}


namespace AWS::IAM {

    entity Role;

    type AuthorizationContext = {
        principalTags: PrincipalTags,
    };

    entity PrincipalTags tags String;
}

namespace AWS::SSM {

    entity ManagedInstance tags String;

    action "getTokenForInstanceAccess" appliesTo {
    principal: [AWS::IdentityStore::User],
    resource: [AWS::EC2::Instance, AWS::SSM::ManagedInstance],
    context: {
        "iam": AWS::IAM::AuthorizationContext
        }
    };
}
```

Integrierte Operatoren

Wenn Sie den Kontext einer Richtlinie für automatische Genehmigung oder Zugriffsverweigerung unter Verwendung verschiedener Bedingungen erstellen, können Sie mit dem Operator &&

zusätzliche Bedingungen angeben. Es gibt viele andere integrierte Operatoren, mit denen Sie Ihren Richtlinienbedingungen zusätzliche Aussagekraft verleihen können. Die folgende Tabelle enthält alle integrierten Operatoren zur Referenz.

Operator	Typen und Überladungen	Description
!	Boolean → boolescher Wert	Logisches Nicht.
==	any → any	Gleichheit. Kann in Argumente n jedes Typs verwendet werden, auch wenn die Typen nicht übereinstimmen. Werte verschiedener Typen sind einander niemals gleich.
!=	any → any	Ungleichheit; das genaue Gegenteil von Gleichheit (siehe oben).
<	(long, long) → boolescher Wert	Lange Ganzzahl kleiner als.
<=	(long, long) → boolescher Wert	Lange Ganzzahl less-than-or-equal -to.
>	(long, long) → boolescher Wert	Lange Ganzzahl größer als.
>=	(long, long) → boolescher Wert	Lange Ganzzahl greater-than-or-equal -to.
in	(entity, entity) → boolescher Wert	Hierarchiezugehörigkeit (reflexiv: A in A ist immer wahr).
	(entity, set(entity)) → boolescher Wert	Hierarchiezugehörigkeit: A in [B, C, …] ist wahr, wenn (A und B) || (A in C) || … Fehler, wenn die Menge eine Nicht-Entität enthält.

Operator	Typen und Überladungen	Description
&&	(Boolean, Boolean) → boolescher Wert	Logisches Und (Kurzschl ießen).
||	(Boolean, Boolean) → boolescher Wert	Logisches Oder (Kurzschl ießen).
.exists()	entity → boolescher Wert	Vorhandensein einer Entität.
has	(entity, attribute) → boolescher Wert	Infix-Operator. `e has f` testet, ob der Datensatz oder die Entität `e` eine Bindung für das Attribut `f` hat. Gibt `false` zurück, wenn `e` nicht existiert oder wenn `e` existiert , aber nicht das Attribut `f` hat. Attribute können als Bezeichner oder Zeichenfo lgeliterale ausgedrückt werden.
like	(string, string) → boolescher Wert	Infix-Operator. `t like p` prüft, ob der Text `t` dem Muster `p` entspricht, das Platzhalterzeichen `*` enthalten kann, die 0 oder mehr eines beliebigen Zeichens entsprech en. Um in `t` ein tatsächliches Sternzeichen abzugleichen, können Sie die Escape-Ze ichenfolge `\*` in `p` verwenden.
.hasTag()	(entity, string) → boolescher Wert	Prüft, ob das angegebene Tag auf die Entität angewendet wurde.

Operator	Typen und Überladungen	Description
.getTag()	(entity, string) → boolescher Wert	Gibt den Wert des angegeben en Tag-Schlüssels zurück.
.contains()	(set, any) → boolescher Wert	Mitgliedschaft festlegen (ist B ein Element von A).
.containsAll()	(set, set) → boolescher Wert	Testet, ob Menge A alle Elemente in Menge B enthält.
.containsAny()	(set, set) → boolescher Wert	Testet, ob Menge A beliebige Elemente aus Menge B enthält.

Beispiele für Richtlinienanweisungen

Nachfolgend finden Sie beispielhafte Richtlinienanweisungen.

```
// Users assuming IAM roles with a principal tag of "Elevated" can automatically access
 nodes tagged with the "Environment" key when the value equals "prod"
permit(principal, action == AWS::SSM::getTokenForInstanceAccess, resource)
when {
    // Verify IAM role principal tag
    context.iam.principalTags.getTag("AccessLevel") == "Elevated" &&

    // Verify the node has a tag with "Environment" tag key and a tag value of "prod"
    resource.hasTag("Environment") &&
    resource.getTag("Environment") == "prod"
};
```

```
// Identity Center users in the "Contractor" division can automatically access nodes
 tagged with the "Environment" key when the value equals "dev"
permit(principal, action == AWS::SSM::getTokenForInstanceAccess, resource)
when {
    // Verify that the user is part of the "Contractor" division
    principal.division == "Contractor" &&

    // Verify the node has a tag with "Environment" tag key and a tag value of "dev"
    resource.hasTag("Environment") &&
```

```
    resource.getTag("Environment") == "dev"
};
```

```
// Identity Center users in a specified group can automatically access nodes tagged
 with the "Environment" key when the value equals "Production"
permit(principal in AWS::IdentityStore::Group::"d4q81745-r081-7079-d789-14da1EXAMPLE",
    action == AWS::SSM::getTokenForInstanceAccess,
    resource)
when {
    resource.hasTag("Environment") &&
    resource.getTag("Environment") == "Production"
};
```

Erstellen Sie eine automatische Genehmigungsrichtlinie für den just-in-time Knotenzugriff

Richtlinien für automatische Genehmigungen verwenden die Richtliniensprache Cedar, um festzulegen, welche Benutzer ohne manuelle Genehmigung automatisch eine Verbindung zu den angegebenen Knoten herstellen können. Eine automatische Genehmigungsrichtlinie enthält mehrere `permit` Anweisungen, die `principal` und `resource` angeben. Jede Anweisung enthält eine `when`-Klausel, in der die Bedingungen für die automatische Genehmigung definiert sind.

Es folgt eine Beispielrichtlinie für die automatische Genehmigung.

```
permit (
    principal in AWS::IdentityStore::Group::"e8c17310-e011-7089-d989-10da1EXAMPLE",
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {
    principal has costCenter && resource.hasTag("CostCenter") && principal.costCenter
 == resource.getTag("CostCenter")
};

permit (
    principal in AWS::IdentityStore::Group::"d4q81745-r081-7079-d789-14da1EXAMPLE",
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {
    principal has organization && resource.hasTag("Engineering") &&
 resource.hasTag("Production") && principal.organization == "Platform"
};
```

```
permit (
    principal,
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {
    principal has employeeNumber && principal.employeeNumber like "E-1*" &&
 resource.hasTag("Purpose") && resource.getTag("Purpose") == "Testing"
};
```

Das folgende Verfahren beschreibt, wie Sie eine automatische Genehmigungsrichtlinie für den just-in-time Knotenzugriff erstellen. Die Zugriffsdauer für eine Zugriffsanforderung, die automatisch genehmigt wird, beträgt 1 Stunde. Dieser Wert kann nicht geändert werden. Sie können nur eine automatische Genehmigungsrichtlinie pro AWS-Konto und AWS-Region haben. Weitere Informationen zum Erstellen einer Richtlinienanweisung finden Sie unter Struktur der Anweisungen und integrierte Operatoren für Richtlinien zur automatischen Genehmigung und Zugriffsverweigerung.

Erstellen Sie eine Richtlinie für automatische Genehmigungen wie folgt:

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Knotenzugriff verwalten aus.
3. Wählen Sie auf der Registerkarte Genehmigungsrichtlinien die Option Richtlinie für automatische Genehmigung erstellen aus.
4. Geben Sie Ihre Richtlinienanweisung für die Richtlinie zur automatischen Genehmigung im Abschnitt Richtlinienanweisung ein. Sie können die bereitgestellten Beispielanweisungen als Hilfe bei der Erstellung Ihrer Richtlinie verwenden.
5. Wählen Sie Richtlinie für automatische Genehmigung erstellen aus.

Erstellen Sie eine Deny-Access-Richtlinie für just-in-time den Knotenzugriff

Richtlinien zur Zugriffsverweigerung verwenden die Cedar-Richtliniensprache, um zu definieren, mit welchen Knoten sich Benutzer ohne manuelle Genehmigung nicht automatisch verbinden können. Eine Zugriffsverweigerungsrichtlinie enthält mehrere `forbid`-Anweisungen, in denen das `principal` und `resource` angegeben wird. Jede Anweisung enthält eine `when`-Klausel, in der die Bedingungen für die ausdrückliche Ablehnung der automatischen Genehmigung definiert sind.

Im Folgenden finden Sie ein Beispiel für eine Zugriffsverweigerungsrichtlinie.

```
forbid (
    principal in AWS::IdentityStore::Group::"e8c17310-e011-7089-d989-10da1EXAMPLE",
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {
    resource.hasTag("Environment") && resource.getTag("Environment") == "Production"
};

forbid (
    principal,
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {
    principal has division && principal.division != "Finance" &&
 resource.hasTag("DataClassification") && resource.getTag("DataClassification") ==
 "Financial"
};


forbid (
    principal,
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {

    principal has employeeNumber && principal.employeeNumber like "TEMP-*" &&
 resource.hasTag("Criticality") && resource.getTag("Criticality") == "High"
};
```

Das folgende Verfahren beschreibt, wie Sie eine Zugriffsverweigerungsrichtlinie für den just-in-time Knotenzugriff erstellen. Weitere Informationen zum Erstellen einer Richtlinienanweisung finden Sie unter Struktur der Anweisungen und integrierte Operatoren für Richtlinien zur automatischen Genehmigung und Zugriffsverweigerung.

> **Note**
>
> Notieren Sie die folgenden Informationen:

- Sie können Richtlinien zur Zugriffsverweigerung erstellen, während Sie beim AWS -Verwaltungskonto oder Konto eines delegierten Administrators angemeldet sind. Sie können nur eine Zugriffsverweigerungsrichtlinie pro AWS Organizations -Organisation haben.
- Just-in-time node access verwendet AWS Resource Access Manager (AWS RAM), um Ihre Zugriffsverweigerungsrichtlinie mit Mitgliedskonten in Ihrer Organisation zu teilen. Wenn Sie Ihre Zugriffsverweigerungsrichtlinie mit den Mitgliedskonten in Ihrer Organisation teilen möchten, muss die gemeinsame Nutzung von Ressourcen über das Verwaltungskonto Ihrer Organisation aktiviert werden. Weitere Informationen finden Sie unter Ressourcenfreigabe für AWS Organizations aktivieren im AWS RAM -Benutzerhandbuch.

Erstellen Sie eine Zugriffsverweigerungsrichtlinie wie folgt:

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Knotenzugriff verwalten aus.
3. Wählen Sie auf der Registerkarte Genehmigungsrichtlinien die Option Richtlinie zur Zugriffsverweigerung erstellen aus.
4. Geben Sie Ihre Richtlinienanweisung für die Richtlinie zur Zugriffsverweigerung im Abschnitt Richtlinienanweisung ein. Sie können die bereitgestellten Beispielanweisungen als Hilfe bei der Erstellung Ihrer Richtlinie verwenden.
5. Wählen Sie Richtlinie zur Zugriffsverweigerung erstellen aus.

Erstellen Sie Genehmigungsrichtlinien für den just-in-time Knotenzugriff mit Amazon Q

Die Verwendung von Amazon Q Developer für die Befehlszeile bietet Beratung und Unterstützung in verschiedenen Aspekten der Softwareentwicklung. Für den just-in-time Knotenzugriff hilft Ihnen Amazon Q bei der Erstellung von Genehmigungsrichtlinien, indem es den Code für die Richtlinien generiert und aktualisiert, Richtlinienerklärungen analysiert und vieles mehr. Die folgenden Informationen beschreiben, wie Genehmigungsrichtlinien mithilfe von Amazon Q für die Befehlszeile erstellt werden.

Identifizieren Sie Ihren Anwendungsfall

Der erste Schritt beim Erstellen von Genehmigungsrichtlinien besteht darin, Ihren Anwendungsfall klar zu definieren. In Ihrer Organisation möchten Sie beispielsweise Zugriffsanforderungen für Knoten mit einem `Environment:Testing`-Tag automatisch genehmigen. Möglicherweise möchten Sie auch explizit automatische Genehmigungen für Knoten mit einem `Environment:Production`-Tag verweigern, wenn eine Mitarbeiter-ID mit `TEMP` beginnt. Für Knoten mit einem `Tier:Database`-Tag möchten Sie möglicherweise zwei Stufen manueller Genehmigungen vorschreiben.

In einem bestimmten Szenario bevorzugen Sie möglicherweise eine Richtlinie oder Bedingung gegenüber einer anderen. Daher empfiehlt es sich, das Richtlinienverhalten ganz klar zu definieren, das bestimmen soll, welche Aussagen am besten zu Ihrem Anwendungsfall und Ihren Voreinstellungen passen.

Einrichten Ihrer Entwicklungsumgebung

Installieren Sie Amazon Q für die Befehlszeile, in der Sie Ihre Genehmigungsrichtlinien entwickeln möchten. Informationen zur Installation von Amazon Q für die Befehlszeile finden Sie unter Amazon Q für die Befehlszeile installieren im Benutzerhandbuch für Amazon Q Developer.

Aus AWS Dokumentationsgründen empfehlen wir außerdem, den MCP-Server zu installieren. Dieser MCP-Server verbindet Amazon Q über die Befehlszeile mit den aktuellsten Dokumentationsressourcen. Informationen zur Verwendung von MCP mit Amazon Q für die Befehlszeile finden Sie unter Verwenden von MCP mit Amazon Q Developer im Benutzerhandbuch für Amazon Q Developer.

Weitere Informationen zum AWS Documentation MCP Server finden Sie unter AWS Documentation MCP Server.

Installieren und konfigurieren Sie den AWS CLI, falls Sie dies noch nicht getan haben. Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

Entwickeln der Inhalte für Genehmigungsrichtlinien

Nachdem Ihr Anwendungsfall identifiziert und die Umgebung eingerichtet ist, können Sie den Inhalt für Ihre Richtlinien entwickeln. Ihr Anwendungsfall und Ihre Einstellungen bestimmen weitgehend die Art der Genehmigungsrichtlinien und -anweisungen, die Sie verwenden.

Wenn Sie sich nicht sicher sind, wie Sie eine bestimmte Richtlinie verwenden sollen, oder weitere Informationen zum Schema für eine Richtlinie benötigen, finden Sie weitere Informationen unter Erstellen Sie Genehmigungsrichtlinien für Ihre Knoten und in den nachfolgenden Themen. In

diesen Themen wird detailliert erläutert, wie Richtlinien bewertet werden, und es werden praktische Beispielanweisungen bereitgestellt.

Im folgenden Verfahren wird beschreiben, wie Sie Genehmigungsrichtlinien mit Amazon Q für die Befehlszeile erstellen.

> **Note**
>
> Amazon Q Developer verwendet generative KI. Möglicherweise müssen Sie die Antworten überprüfen. Weitere Informationen finden Sie in der AWS -Richtlinie für verantwortungsvolle KI.

Erstellen Sie Genehmigungsrichtlinien mit Amazon Q für die Befehlszeile wie folgt:

1. Öffnen Sie Ihr Terminal und starten Sie eine Sitzung mit Amazon Q.

   ```
   $ q chat
   ```

2. Geben Sie einen Prompt ein und beschreiben Sie, welche Art von Richtlinienanweisung Sie erstellen möchten. Wenn Sie beispielsweise das vorherige Szenario für eine Genehmigungsrichtlinie verwenden, könnten Sie den folgenden Prompt eingeben:

   ```
   Create a Cedar policy statement that allows a principal to use the
    “getTokenForInstanceAccess" action for a resource
   that has a tag key “Environment” and tag value of “Testing”. Use the following
    example statement for reference to verify the syntax.

   permit (
       principal in AWS::IdentityStore::Group::"e8c17310-e011-7089-d989-10da1EXAMPLE",
       action == AWS::SSM::Action::"getTokenForInstanceAccess",
       resource
   )
   when {
       principal has organization && resource.hasTag("Engineering") &&
    resource.hasTag("Production") && principal.organization == "Platform"
   };

   The Cedar policy statement you create must be valid. Validate the Cedar statement
    using the AWS documentation provided for just-in-time node access policy statement
    structure and make sure you abide by the schema.
   ```

3. Die Eingabe dieses Beispiel-Prompts führte zu der folgenden Ausgabe:

```
Based on the AWS documentation for Cedar policy statements and the examples
 provided, I'll create a valid Cedar policy
statement that allows a principal to use the "getTokenForInstanceAccess" action for
 resources with the tag key "Environment" and tag value "Testing".

Here's the Cedar policy statement:

permit (
    principal,
    action == AWS::SSM::Action::"getTokenForInstanceAccess",
    resource
)
when {
    resource.hasTag("Environment") &&
    resource.getTag("Environment") == "Testing"
};


This policy statement:
1. Uses the permit effect to allow access
2. Applies to any principal (since no specific principal is defined)
3. Specifically targets the "getTokenForInstanceAccess" action
4. Applies to resources that have a tag with key "Environment" and value "Testing"
5. Uses the proper syntax for the condition clause with the hasTag() and getTag()
 operators

The statement follows the Cedar policy language structure as documented in the AWS
 Systems Manager documentation for just-in-time node access policies.
```

4. Fügen Sie die Richtlinienanweisung in eine neue Cedar-Datei ein, um die Syntax zu überprüfen. Sie können die Anweisung auch mithilfe des Cedar-Playgrounds überprüfen.
5. Nachdem Sie die Anweisung validiert haben, speichern Sie die Cedar-Datei.
6. Führen Sie auf Ihrem Terminal den folgenden AWS CLI Befehl aus, um ein SSM-Dokument mit Ihrem Richtlinieninhalt zu erstellen. Bevor Sie Genehmigungsrichtlinien in einer Produktionsumgebung verwenden, testen Sie Ihre Genehmigungsrichtlinien mit einer Teilmenge von Identitäten und Knoten in einem und. AWS-Konto AWS-Region

> **Note**
>
> Für eine Richtlinie zur automatische Genehmigung muss der Dokumentname `SSM-JustInTimeAccessAutoApprovalPolicy` lauten. Es kann nur eine automatische Genehmigungsrichtlinie pro AWS-Konto und AWS-Region geben. Bei einer Zugriffsverweigerungsrichtlinie muss der Dokumentname `SSM-JustInTimeAccessDenyAccessOrgPolicy` lauten. Pro AWS Organizations Organisation kann es nur eine Zugriffsverweigerungsrichtlinie geben, und die Richtlinie muss im delegierten Administratorkonto für Systems Manager erstellt werden. Die Benennungsbeschränkungen für Richtlinien zur manuellen Genehmigung entsprechen denen anderer SSM-Dokumente. Weitere Informationen finden Sie unter CreateDocument.

Linux & macOS

```
aws ssm create-document \
    --content file://path/to/file/policyContent.cedar \
    --name "SSM-JustInTimeAccessAutoApprovalPolicy" \
    --document-type "AutoApproval"
```

Windows

```
aws ssm create-document ^
    --content file://C:\path\to\file\policyContent.cedar ^
    --name "SSM-JustInTimeAccessAutoApprovalPolicy" ^
    --document-type "AutoApproval"
```

PowerShell

```
$cedar = Get-Content -Path "C:\path\to\file\policyContent.cedar" | Out-String
New-SSMDocument `
    -Content $cedar `
    -Name "SSM-JustInTimeAccessAutoApprovalPolicy" `
    -DocumentType "AutoApproval"
```

Sitzungseinstellungen für den just-in-time Knotenzugriff aktualisieren

Mit dem just-in-time Knotenzugriff können Sie allgemeine Sitzungs- und Protokollierungspräferenzen in allen AWS-Konto Bereichen und AWS-Region in Ihrer Organisation festlegen. Alternativ können Sie ein Dokument CloudFormation StackSets mit den Sitzungseinstellungen für mehrere Konten und Regionen erstellen, um einheitliche Sitzungseinstellungen zu gewährleisten. Informationen zum Schema für Dokumente mit Sitzungseinstellungen finden Sie unter Schema des Sitzungsdokuments.

Für Protokollierungszwecke empfehlen wir die Verwendung der Streaming-Option mit Amazon CloudWatch Logs. Mit dieser Funktion können Sie einen kontinuierlichen Stream von Sitzungsdatenprotokollen an CloudWatch Logs senden. Beim Streamen von Sitzungsdaten sind wichtige Details enthalten, z. B. die Befehle, die ein Benutzer in einer Sitzung ausgeführt hat, die ID des Benutzers, der die Befehle ausgeführt hat, und Zeitstempel, wann die Sitzungsdaten in CloudWatch Logs gestreamt werden. Beim Streamen von Sitzungsdaten werden die Protokolle JSON-formatiert, um Ihnen bei der Integration in Ihre vorhandenen Protokollierungslösungen zu helfen.

Systems Manager beendet just-in-time Knotenzugriffssitzungen nicht automatisch. Es hat sich bewährt, Werte für die maximale Sitzungsdauer und das Timeout bei Sitzungsinaktivität anzugeben. Mit diesen Einstellungen können Sie verhindern, dass ein Benutzer länger mit einem Knoten verbunden bleibt, als in der Zugriffsanforderung genehmigt wurde. Das folgende Verfahren beschreibt, wie Sie die Sitzungseinstellungen für den just-in-time Knotenzugriff aktualisieren.

> **Important**
>
> Sie müssen die AWS KMS Schlüssel, die für die Session Manager Verschlüsselung und RDP-Aufzeichnung beim just-in-time Knotenzugriff verwendet werden, mit dem Tag-Schlüssel `SystemsManagerJustInTimeNodeAccessManaged` und dem Tag-Wert `true` kennzeichnen.
>
> Weitere Informationen zum Kennzeichnen von KMS-Schlüsseln finden Sie unter Tags in AWS KMS im Entwicklerhandbuch für AWS Key Management Service .

Aktualisieren Sie die Sitzungseinstellungen wie folgt

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Einstellungen aus.

3. Wählen Sie die Registerkarte Just-in-time Knotenzugriff aus.
4. Wählen Sie im Abschnitt Sitzungseinstellungen die Option Bearbeiten aus.
5. Aktualisieren Sie die allgemeinen Einstellungen und die Protokollierungseinstellungen und wählen Sie Speichern aus.

Benachrichtigungen für just-in-time Zugriffsanfragen konfigurieren

Sie können Systems Manager so konfigurieren, dass Benachrichtigungen gesendet werden, wenn ein Benutzer eine just-in-time Knotenzugriffsanfrage an die E-Mail-Adressen oder den Chat-Client für Genehmiger und den Anforderer erstellt. Die Benachrichtigung enthält den Grund für die vom Anforderer bereitgestellte Zugriffsanfrage, den AWS-Konto, AWS-Region, Status der Anfrage und die ID des Zielknotens. Derzeit unterstützt Systems Manager Slack und Microsoft Teams über die Integration mit Amazon Q Developer in Chat-Anwendungen. Wenn die Benachrichtigung über Chat-Client erfolgt, können die Genehmiger direkt mit den Zugriffsanforderungen interagieren. Sie müssen sich nicht mehr an der Konsole anmelden, um auf Zugriffsanforderungen zu reagieren.

Bevor Sie beginnen

Bevor Sie einen Chat-Client für Benachrichtigungen über den just-in-time Knotenzugriff konfigurieren, beachten Sie die folgende Anforderung:

- Wenn Sie IAM-Rollen verwenden, um Benutzeridentitäten in Ihrem Konto zu verwalten, müssen Sie die E-Mail-Adressen der Genehmiger oder Anforderer, an die Sie Benachrichtigungen senden möchten, manuell mit der zugehörigen Rolle verknüpfen. Andernfalls können die vorgesehenen Empfänger nicht per E-Mail benachrichtigt werden.

In den folgenden Verfahren wird beschrieben, wie Benachrichtigungen für just-in-time Knotenzugriffsanforderungen konfiguriert werden.

Um einen Chat-Client für Benachrichtigungen über den just-in-time Knotenzugriff zu konfigurieren

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Einstellungen aus.
3. Wählen Sie die Registerkarte Just-in-time Knotenzugriff aus.
4. Wählen Sie im Abschnitt Chat die Option Neuen Client konfigurieren aus.

5. Wählen Sie in der Dropdownliste Clienttyp auswählen den Typ des Chat-Clients aus, den Sie konfigurieren möchten, und wählen Sie Weiter aus.
6. Sie werden aufgefordert, Amazon Q Developer den Zugriff auf Chat-Clients zu erlauben. Wählen Sie Zulassen aus.
7. Geben Sie im Abschnitt Kanal konfigurieren die Informationen für Ihren Chat-Client-Kanal ein und wählen Sie die Arten von Benachrichtigungen aus, die Sie erhalten möchten.
8. Wenn Sie Benachrichtigungen für Slack konfigurieren, laden Sie „@Amazon Q“ zu jedem Slack-Kanal ein, für den Benachrichtigungen konfiguriert werden.
9. Wählen Kanal konfigurieren aus.

> **Note**
>
> Um approving/rejecting Zugriffsanfragen direkt von einem Slack-Channel aus zuzulassen, stellen Sie sicher, dass die IAM-Rolle, die mit dem Slack-Channel konfiguriert ist, über `ssm:SendAutomationSignal` Berechtigungen verfügt und über eine Vertrauensrichtlinie verfügt, die Chatbot einschließt:
>
> ```
> {
>             "Effect": "Allow",
>             "Principal": {
>                 "Service": "chatbot.amazonaws.com"
>             },
>             "Action": "sts:AssumeRole"
> }
> ```

Um E-Mail-Benachrichtigungen für Benachrichtigungen über den Knotenzugriff zu konfigurieren just-in-time

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Einstellungen aus.
3. Wählen Sie die Registerkarte Just-in-time Knotenzugriff aus.
4. Wählen Sie im Bereich E-Mail die Option Bearbeiten aus.
5. Wählen Sie E-Mails hinzufügen und dann die IAM-Rolle aus, der Sie E-Mail-Adressen manuell zuordnen möchten.

6. Geben Sie eine E-Mail-Adresse in das Feld E-Mail-Adresse ein. Immer wenn eine Zugriffsanforderung erstellt wird, für die eine Genehmigung durch die von Ihnen angegebene IAM-Rolle erforderlich ist, werden die E-Mail-Adressen, die der Rolle zugeordnet sind, benachrichtigt.
7. Wählen Sie E-Mail-Adresse hinzufügen aus.

Aufzeichnen von RDP-Verbindungen

Just-in-time Der Knotenzugriff beinhaltet die Möglichkeit, RDP-Verbindungen aufzuzeichnen, die zu Ihren Windows Server Knoten hergestellt wurden. Für die Aufzeichnung von RDP-Verbindungen sind ein S3-Bucket und ein kundenverwalteter AWS Key Management Service -Schlüssel (AWS KMS) erforderlich. Das AWS KMS key wird verwendet, um die Aufzeichnungsdaten vorübergehend zu verschlüsseln, während sie generiert und auf Systems Manager Manager-Ressourcen gespeichert werden. Bei dem kundenverwalteten Schlüssel muss es sich um einen symmetrischen Schlüssel handeln, der zum Verschlüsseln und Entschlüsseln verwendet wird. Sie können entweder einen Schlüssel für mehrere Regionen für Ihre Organisation verwenden, oder Sie müssen in jeder Region, in der Sie den just-in-time Knotenzugriff aktiviert haben, einen vom Kunden verwalteten Schlüssel erstellen.

Wenn Sie die KMS-Verschlüsselung für den S3-Bucket aktiviert haben, in dem Sie die Aufzeichnungen speichern, müssen Sie dem `ssm-guiconnect`-Service-Prinzipal Zugriff auf den kundenverwalteten Schlüssel gewähren, der für die Bucket-Verschlüsselung verwendet wird. Bei dem kundenverwalteten Schlüssel kann es sich um einen anderen Schlüssel handeln als den, den Sie in den Einstellungen für die Aufzeichnung angegeben haben. Darin muss auch festgelegt sein, wofür die `kms:CreateGrant`-Berechtigung zum Herstellen von Verbindungen erforderlich ist.

Konfigurieren der S3-Bucket-Verschlüsselung für RDP-Aufzeichnungen

Verbindungsaufzeichnungen werden in dem S3-Bucket gespeichert, den Sie angeben, wenn Sie die RDP-Aufzeichnung aktivieren.

Wenn Sie einen KMS-Schlüssel als Standardverschlüsselungsmechanismus für den S3-Bucket (SSE-KMS) verwenden, müssen Sie dem `ssm-guiconnect`-Service-Prinzipal Zugriff auf `kms:GenerateDataKey`-Aktionen für den Schlüssel gewähren. Wir empfehlen die Verwendung eines kundenverwalteten Schlüssels, wenn Sie die SSE-KMS-Verschlüsselung mit dem S3-Bucket verwenden. Der Grund ist, dass Sie für einen kundenverwalteten Schlüssel die zugehörige Schlüsselrichtlinie aktualisieren können. Sie können die wichtigsten Richtlinien für Von AWS verwaltete Schlüssel nicht aktualisieren.

> **Important**
>
> Sie müssen die AWS KMS Schlüssel, die für die Session Manager Verschlüsselung und RDP-Aufzeichnung beim just-in-time Knotenzugriff verwendet werden, mit dem Tag-Schlüssel `SystemsManagerJustInTimeNodeAccessManaged` und dem Tag-Wert `true` kennzeichnen.
> Weitere Informationen zum Kennzeichnen von KMS-Schlüsseln finden Sie unter Tags in AWS KMS im Entwicklerhandbuch für AWS Key Management Service .

Verwenden Sie die folgende Richtlinie für kundenverwaltete Schlüssel, um dem `ssm-guiconnect`-Service Zugriff auf den KMS-Schlüssel für S3-Speicher zu gewähren. Informationen zur Aktualisierung eines kundenverwalteten Schlüssels finden Sie unter Ändern einer Schlüsselrichtlinie im Entwicklerhandbuch für AWS Key Management Service .

Ersetzen Sie sie jeweils *`example resource placeholder`* durch Ihre eigenen Informationen:

- *`account-id`*steht für die ID AWS-Konto desjenigen, der die Verbindung initiiert.
- *`region`*steht für den AWS-Region Ort, an dem sich der S3-Bucket befindet. (Sie können * verwenden, wenn der Bucket Aufzeichnungen aus mehreren Regionen empfängt. Beispiel: `s3.*.amazonaws.com`.)

> **Note**
>
> Sie können in der Richtlinie `aws:SourceOrgID` anstatt `aws:SourceAccount` verwenden, wenn das Konto einer Organisation in AWS Organizations angehört.

```
{
    "Sid": "Allow the GUI Connect service principal to access S3",
    "Effect": "Allow",
    "Principal": {
        "Service": "ssm-guiconnect.amazonaws.com"
    },
    "Action": [
        "kms:GenerateDataKey*"
    ],
    "Resource": "*",
```

```
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "account-id"
        },
        "StringLike": {
            "kms:ViaService": "s3.region.amazonaws.com"
        }
    }
}
```

Konfigurieren von IAM-Berechtigungen für die Aufzeichnung von RDP-Verbindungen

Zusätzlich zu den erforderlichen IAM-Berechtigungen für den just-in-time Knotenzugriff müssen dem Benutzer oder der Rolle, die Sie verwenden, je nach der Aufgabe, die Sie ausführen müssen, die folgenden Berechtigungen gewährt werden.

Berechtigungen für die Konfiguration der Verbindungsaufzeichnung

Zum Konfigurieren der RDP-Verbindungsaufzeichnung sind die folgenden Berechtigungen erforderlich:

- `ssm-guiconnect:UpdateConnectionRecordingPreferences`
- `ssm-guiconnect:GetConnectionRecordingPreferences`
- `ssm-guiconnect:DeleteConnectionRecordingPreferences`
- `kms:CreateGrant`

Berechtigungen für das Initiieren von Verbindungen

Um RDP-Verbindungen mit just-in-time Knotenzugriff herzustellen, sind die folgenden Berechtigungen erforderlich:

- `ssm-guiconnect:CancelConnection`
- `ssm-guiconnect:GetConnection`
- `ssm-guiconnect:StartConnection`
- `kms:CreateGrant`

Bevor Sie beginnen

Um Ihre Verbindungsaufzeichnungen zu speichern, müssen Sie zunächst einen S3-Bucket erstellen und die folgende Bucket-Richtlinie hinzufügen. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

(Informationen zum Hinzufügen oder Ändern einer Bucket-Richtlinie finden Sie unter Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole im Benutzerhandbuch für Amazon Simple Storage Service.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ConnectionRecording",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "ssm-guiconnect.amazonaws.com"
                ]
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ],
            "Condition":{
            "StringEquals":{
                "aws:SourceAccount":"111122223333"
                }
            }
        }
    ]
}
```

Aktivierung und Konfiguration der RDP-Verbindungsaufzeichnung

Im folgenden Verfahren wird beschrieben, wie Sie die RDP-Verbindungsaufzeichnung aktivieren und konfigurieren.

Aktivieren und Konfigurieren Sie die RDP-Verbindungsaufzeichnung wie folgt

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Einstellungen aus.
3. Wählen Sie die Registerkarte Just-in-time Knotenzugriff aus.
4. Wählen Sie im Bereich RDP-Aufzeichnung die Option RDP-Aufzeichnung aktivieren aus.
5. Wählen Sie den S3-Bucket aus, in den Sie Sitzungsaufzeichnungen hochladen möchten.
6. Wählen Sie den kundenverwalteten Schlüssel aus, der verwendet werden soll, um die Aufzeichnungsdaten vorübergehend zu verschlüsseln, während sie generiert und auf Systems Manager-Ressourcen gespeichert werden. (Dabei kann es sich um einen anderen kundenverwalteten Schlüssel handeln als den, den Sie zum Verschlüsseln des Buckets verwenden.)
7. Wählen Sie Speichern aus.

Statuswerte für die RDP-Verbindungsaufzeichnung

Zu den gültigen Statuswerten für RPD-Verbindungsaufzeichnungen gehören:

- `Recording`: Die Verbindung wird gerade aufgezeichnet
- `Processing`: Das Video wird verarbeitet, nachdem die Verbindung beendet wurde.
- `Finished`: Endstatus „Erfolg“: Das aufgezeichnete Video wurde verarbeitet und in den angegebenen Bucket hochgeladen.
- `Failed`: Endstatus „Fehler“. Die Verbindung wurde nicht aufgezeichnet.
- `ProcessingError`- Bei der Videoverarbeitung ist ein oder mehrere Zwischenprodukte failures/errors aufgetreten. Mögliche Ursachen sind Fehler bei der Serviceabhängigkeit oder fehlende Berechtigungen aufgrund einer Fehlkonfiguration des S3-Buckets, der für das Speichern von Aufzeichnungen angegeben wurde. In diesem Status wird der Service weiterhin versuchen, die Aufzeichnung zu verarbeiten.

> **Note**
>
> `ProcessingError` kann darauf zurückzuführen sein, dass der `ssm-guiconnect`-Service-Prinzipal nicht berechtigt ist, Objekte in den S3-Bucket hochzuladen, nachdem die Verbindung hergestellt wurde. Eine weitere mögliche Ursache sind fehlende KMS-

Berechtigungen für den KMS-Schlüssel, der für die S3-Bucket-Verschlüsselung verwendet wird.

Ändern der Ziele

Wenn Sie den just-in-time Knotenzugriff einrichten, wählen Sie die Ziele aus, für die Sie den just-in-time Knotenzugriff einrichten möchten. Ziele bestehen aus AWS Organizations Organisationseinheiten (OUs) und AWS-Regionen. Standardmäßig werden dieselben Ziele, die Sie bei der Einrichtung der einheitlichen Systems Manager Manager-Konsole ausgewählt haben, für den just-in-time Knotenzugriff ausgewählt. Sie können wählen, ob Sie den just-in-time Knotenzugriff für alle Ziele oder für eine Teilmenge der Ziele einrichten möchten, die Sie bei der Einrichtung der einheitlichen Systems Manager Manager-Konsole angegeben haben. Das Hinzufügen neuer Ziele, die bei der Einrichtung der einheitlichen Systems Manager Manager-Konsole nicht ausgewählt wurden, wird nicht unterstützt. Sie können die ausgewählten Ziele ändern, nachdem Sie den just-in-time Knotenzugriff eingerichtet haben.

Das folgende Verfahren beschreibt, wie Sie die Ziele für den just-in-time Knotenzugriff ändern.

Ändern Sie Ziele wie folgt:

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Einstellungen aus.
3. Wählen Sie die Registerkarte Just-in-time Knotenzugriff aus.
4. Wählen Sie im Bereich Ziele die Option Bearbeiten aus.
5. Wählen Sie die Organisationseinheiten und Regionen aus, in denen Sie den just-in-time Knotenzugriff verwenden möchten.
6. Wählen Sie Speichern.

Ändern von Identitätsanbietern

Standardmäßig verwendet der just-in-time Knotenzugriff IAM als Identitätsanbieter. Nach der Aktivierung des just-in-time Knotenzugriffs können Kunden, die die einheitliche Konsole mit einer Organisation verwenden, diese Einstellung ändern, um IAM Identity Center zu verwenden. Just-in-timeDer Knotenzugriff unterstützt IAM Identity Center nicht als Identitätsanbieter, wenn es für ein einzelnes Konto und eine Region eingerichtet ist.

Das folgende Verfahren beschreibt, wie Sie den Identitätsanbieter für den just-in-time Knotenzugriff ändern.

Ändern Sie Identitätsanbieter wie folgt:

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Einstellungen aus.
3. Wählen Sie die Registerkarte Just-in-time Knotenzugriff aus.
4. Wählen Sie im Abschnitt Benutzeridentität die Option Bearbeiten aus.
5. Wählen Sie AWS IAM Identity Center.
6. Wählen Sie Speichern.

Starten Sie eine just-in-time Knotenzugriffssitzung

Nach der Aktivierung und Einrichtung des just-in-time Knotenzugriffs sowie der Konfiguration der Sitzungs- und Benachrichtigungseinstellungen sind Benutzer bereit, just-in-time Knotenzugriffssitzungen zu starten. Sie können Sitzungen mithilfe des just-in-time Knotenzugriffs über die Systems Manager Manager-Konsole oder AWS Command Line Interface über das Session Manager Plug-In starten. Just-in-timeKnotenzugriffssitzungen können auf Knoten im selben Konto und in derselben Region gestartet werden. In den folgenden Verfahren wird beschrieben, wie Sitzungen mit just-in-time Knotenzugriff gestartet werden.

> **Note**
>
> Wenn Ihre Benutzer zuvor Verbindungen Session Manager zu Knoten hergestellt haben, müssen Sie beispielsweise die Session Manager Berechtigungen aus ihren IAM-Richtlinien entfernen`ssm:StartSession`, um Sitzungen mit just-in-time Knotenzugriff zu starten. Andernfalls wird weiterhin Session Manager zum Herstellen von Knotenverbindungen verwendet.

Um eine Sitzung mit just-in-time Knotenzugriff über die Konsole zu starten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Knoten erkunden aus.

3. Wählen Sie den Knoten aus, zu dem Sie eine Verbindung herstellen möchten.
4. Wählen Sie in der Dropdownliste Aktionen die Option Verbinden aus.

Wenn die Genehmigungsrichtlinien Ihrer Organisation nicht zulassen, dass automatisch eine Verbindung zum Knoten hergestellt wird, werden Sie aufgefordert, eine Zugriffsanforderung zu senden. Nachdem Sie die angeforderten Informationen ausgefüllt und die Zugriffsanforderung abgesendet haben, können Sie Sitzungen für diesen Knoten starten, sobald alle erforderlichen Genehmigungen eingegangen sind.

Um eine Sitzung mit just-in-time Knotenzugriff zu starten, verwenden Sie den AWS CLI

1. Führen Sie den folgenden Befehl aus, um den Zugriffsanforderungsworkflow zu starten. Achten Sie dabei darauf, den *placeholder values* durch Ihre eigenen Informationen zu ersetzen.

   ```
   aws ssm start-access-request \
       --targets  Key=InstanceIds,Values=i-02573cafcfEXAMPLE
       --reason "Troubleshooting networking performance issue"
   ```

   Abhängig von den Genehmigungsrichtlinien Ihrer Organisation werden Sie entweder automatisch mit dem Knoten verbunden oder der manuelle Genehmigungsprozess wird gestartet. Notieren Sie sich bei Anforderungen, die manuell genehmigt werden müssen, die ID der Zugriffsanforderung, die in der Antwort zurückgegeben wird.
2. Warten Sie, bis alle erforderlichen Genehmigungen vorliegen.
3. Nachdem alle erforderlichen Genehmigungen erteilt wurden, führen Sie den folgenden Befehl aus, um ein Zugriffstoken mit temporären Anmeldeinformationen zu erhalten. Ersetzen Sie *placeholder values* durch Ihre Informationen.

   ```
   aws ssm get-access-token \
       --access-request-id oi-12345abcdef
   ```

   Notieren Sie das in der Antwort zurückgegebene Zugriffstoken.
4. Führen Sie den folgenden Befehl aus, um die temporären Anmeldeinformationen in der zu verwenden AWS CLI, und stellen Sie sicher, dass Sie sie durch Ihre eigenen Informationen ersetzen. *placeholder values*

   ```
   export AWS_SESSION_TOKEN=AQoDYXdzEJr...<remainder of session token>
   ```

5. Führen Sie den folgenden Befehl aus, um eine Sitzung *placeholder values* mit dem Knoten zu starten, und achten Sie darauf, die durch Ihre eigenen Informationen zu ersetzen.

```
aws ssm start-session \
    --target i-02573cafcfEXAMPLE
```

Verwaltung von just-in-time Zugriffsanfragen

Für mehr Transparenz in Ihrem Unternehmen repliziert Systems Manager Zugriffsanforderungen an das delegierte Administratorkonto für Ihre Organisation. Um Ihnen zu helfen, die Compliance-Anforderungen zu erfüllen, speichert Systems Manager alle Zugriffsanforderungen ein Jahr lang. In den folgenden Themen wird beschrieben, wie Zugriffsanforderungen für just-in-time Knoten verwaltet werden. Diese Informationen sind für Genehmiger von Zugriffsanforderungen bestimmt. Bevor Sie beginnen, empfehlen wir Ihnen, Ihre IAM-Richtlinien zu überprüfen und sicherzustellen, dass Sie über die erforderlichen Berechtigungen für die Verwaltung des just-in-time Knotenzugriffs verfügen. Weitere Informationen finden Sie unter just-in-timeZugriff mit Systems Manager einrichten.

Themen

- Zugriffsanfragen für just-in-time Knoten genehmigen und ablehnen

Zugriffsanfragen für just-in-time Knoten genehmigen und ablehnen

Genehmiger von Zugriffsanfragen können just-in-time Knotenzugriffsanforderungen über die einheitliche Systems Manager Manager-Konsole oder mithilfe Ihres bevorzugten Befehlszeilentools genehmigen oder ablehnen. Diese Informationen sind für Genehmiger von Zugriffsanforderungen bestimmt. Wenn Sie nicht über die erforderlichen Berechtigungen verfügen, um Zugriffsanforderungen zu genehmigen oder zu verweigern, wenden Sie sich an Ihren Administrator. In den folgenden Verfahren wird beschrieben, wie Zugriffsanforderungen für just-in-time Knoten genehmigt oder verweigert werden.

So genehmigen oder verweigern Sie just-in-time Knotenzugriffsanfragen mit der Konsole

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Knotenzugriff verwalten aus.
3. Wählen Sie die Registerkarte Zugriffsanforderungen aus.

4. Wählen Sie den Schalter Anforderungen für mich aus.
5. Aktivieren Sie das Kontrollkästchen neben der Zugriffsanforderung, die Sie genehmigen oder verweigern möchten.
6. Wählen Sie Genehmigen oder Verweigern aus.

Nachdem Sie eine Zugriffsanforderung genehmigt haben, können Sie Ihre Genehmigung jederzeit widerrufen, indem Sie auf Widerrufen klicken.

Um just-in-time Knotenzugriffsanfragen über die Befehlszeile zu genehmigen oder abzulehnen

1. Notieren Sie sich die Zugriffsanforderungs-ID aus der Benachrichtigung. Beispiel, *oi-12345abcdef*.
2. Führen Sie den folgenden Befehl aus, um Details zum Workflow für die Genehmigung von Zugriffsanfragen zurückzugeben, und stellen Sie sicher, dass Sie diese durch Ihre eigenen Informationen ersetzen. *placeholder values*

   ```
   aws ssm get-ops-item \
       --ops-item-id oi-12345abcdef
   ```

   Notieren Sie sich den `automationExecutionId`-Wert im `/aws/accessrequest`-Feld für die `OperationalData`. Beispiel, *9231944f-61c6-40be-8bce-8ee2bEXAMPLE*.
3. Führen Sie den folgenden Befehl aus, um die Zugriffsanforderung zu genehmigen oder zu verweigern. Verwenden Sie den `Approve`-Signaltyp, um die Anfrage zu genehmigen. und `Deny`, um sie zu verweigern. Ersetzen Sie *placeholder values* durch Ihre eigenen Informationen.

   ```
   aws ssm send-automation-signal \
       --automation-execution-id 9231944f-61c6-40be-8bce-8ee2bEXAMPLE \
       --signal-type "Approve"
   ```

Übergang zum just-in-time Knotenzugriff von Session Manager

Wenn Sie den just-in-time Knotenzugriff aktivieren, nimmt Systems Manager keine Änderungen an Ihren vorhandenen Ressourcen für vorSession Manager. Dadurch wird sichergestellt, dass Ihre bestehende Umgebung nicht unterbrochen wird und Benutzer weiterhin Sitzungen starten können, während Sie Genehmigungsrichtlinien erstellen und validieren. Sobald Sie bereit sind, Ihre Genehmigungsrichtlinien zu testen, müssen Sie Ihre bestehenden IAM-Richtlinien ändern, um

den Übergang zum just-in-time Knotenzugriff abzuschließen. Dazu gehören das Hinzufügen der erforderlichen Berechtigungen für den just-in-time Knotenzugriff auf Identitäten und das Entfernen der Berechtigungen für den `StartSession` API-Vorgang für. Session Manager Wir empfehlen, Genehmigungsrichtlinien mit einer Teilmenge von Identitäten und Knoten in einem und zu testen. AWS-Konto AWS-Region

Weitere Hinweise zu den für den just-in-time Knotenzugriff erforderlichen Berechtigungen finden Sie unter. just-in-timeZugriff mit Systems Manager einrichten

Weitere Informationen zum Ändern und den IAM-Berechtigungen der Identität finden Sie unter Hinzufügen und Entfernen von IAM-Identitäts-Berechtigungen im IAM-Benutzerhandbuch.

Im Folgenden wird detailliert beschrieben, wie Sie von zum just-in-time Knotenzugriff wechseln könnenSession Manager.

Die Umstellung von Session Manager auf just-in-time Node Access erfordert sorgfältige Planung und Tests, um einen reibungslosen Übergang ohne Betriebsunterbrechung zu gewährleisten. In den folgenden Abschnitten wird beschrieben, wie Sie diesen Prozess abschließen können.

Voraussetzungen

Überprüfen Sie zu Beginn, ob Sie die folgenden Aufgaben ausgeführt haben:

- Einrichten der vereinheitlichten Systems-Manager-Konsole.
- Es wurde überprüft, ob Sie berechtigt sind, die IAM-Richtlinien in Ihrem Konto zu ändern.
- Es wurden alle IAM-Richtlinien und -Rollen identifiziert, die derzeit Session Manager-Berechtigungen erteilen.
- Es wurde Ihre aktuelle Session Manager-Konfiguration dokumentiert, einschließlich der Sitzungs- und Protokollierungseinstellungen.

Bewertung

Beurteilen Sie Ihre aktuelle Umgebung und skizzieren Sie das gewünschte Genehmigungsverhalten, indem Sie die folgenden Aufgaben ausführen:

1. Inventarisieren Ihrer Knoten – Identifizieren Sie alle Knoten, auf die Benutzer derzeit über Session Manager zugreifen.
2. Identifizieren von Benutzerzugriffsmustern – Dokumentieren Sie, welche Benutzer oder Rollen Zugriff auf welche Knoten benötigen und unter welchen Umständen.

3. Zuordnen von Genehmigungsworkflows – Legen Sie fest, wer Zugriffsanforderungen für verschiedene Knotentypen genehmigen soll.
4. Überprüfen der Tagging-Strategie – Stellen Sie sicher, dass Ihre Knoten ordnungsgemäß gekennzeichnet sind, um Ihre geplanten Genehmigungsrichtlinien zu unterstützen.
5. Prüfen vorhandener IAM-Richtlinien – Identifizieren Sie alle Richtlinien, die Session Manager-Berechtigungen beinhalten.

Planung

Schrittweiser Ansatz

Beim Übergang vom Knotenzugriff Session Manager zum just-in-time Knotenzugriff empfehlen wir, einen schrittweisen Ansatz wie den folgenden zu verwenden:

1. Phase 1: Einrichtung und Konfiguration — Aktivieren Sie den just-in-time Knotenzugriff, ohne bestehende Session Manager Berechtigungen zu ändern.
2. Phase 2: Richtlinienentwicklung – Erstellen und testen Sie Genehmigungsrichtlinien für Ihre Knoten.
3. Phase 3: Pilotmigration — Ändern Sie eine kleine Gruppe unkritischer Knoten und Benutzer oder Rollen vom Knotenzugriff Session Manager zum just-in-time Knotenzugriff.
4. Phase 4: Vollständige Migration – Schrittweise Migration aller verbleibenden Knoten und Benutzer oder Rollen.

Überlegungen zum Zeitplan

Berücksichtigen Sie bei der Erstellung Ihres Zeitplans für den Übergang vom Knotenzugriff Session Manager zum just-in-time Knotenzugriff die folgenden Faktoren:

- Nehmen Sie sich Zeit für die Schulung der Benutzer und die Anpassung an den neuen Genehmigungsablauf.
- Planen Sie Migrationen in Zeiten geringerer betrieblicher Aktivität.
- Schließen Sie Pufferzeit für Problembehebungen und Anpassungen mit ein.
- Planen Sie einen Parallelbetrieb ein, in dem beide Systeme verfügbar sind.

Implementierungsschritte

Phase 1: Einrichtung und Konfiguration

1. Aktivieren Sie den just-in-time Knotenzugriff in der Systems Manager Manager-Konsole. Die detaillierten Schritte finden Sie unter just-in-timeZugriff mit Systems Manager einrichten.
2. Konfigurieren Sie die Sitzungseinstellungen für den just-in-time Knotenzugriff so, dass sie Ihren aktuellen Session Manager Einstellungen entsprechen. Weitere Informationen finden Sie unter Sitzungseinstellungen für den just-in-time Knotenzugriff aktualisieren.
3. Richten Sie Benachrichtigungseinstellungen für Zugriffsanforderungen ein. Weitere Informationen finden Sie unter Benachrichtigungen für just-in-time Zugriffsanfragen konfigurieren.
4. Wenn Sie RDP-Verbindungen zu Windows Server-Knoten verwenden, konfigurieren Sie die RDP-Aufzeichnung. Weitere Informationen finden Sie unter Aufzeichnen von RDP-Verbindungen.

Phase 2: Richtlinienentwicklung

1. Erstellen Sie IAM-Richtlinien für Administratoren und Benutzer mit just-in-time Knotenzugriff.
2. Entwickeln Sie Genehmigungsrichtlinien auf der Grundlage Ihrer Sicherheitsanforderungen und Ihres Anwendungsfalls.
3. Testen Sie Ihre Richtlinien in einer Nicht-Produktionsumgebung, um sicherzustellen, dass sie wie erwartet funktionieren.

Phase 3: Pilot-Migration

1. Wählen Sie eine kleine Gruppe von Benutzern und unkritischen Knoten für das Pilotprojekt aus.
2. Erstellen Sie neue IAM-Richtlinien für Pilotbenutzer, die just-in-time Knotenzugriffsberechtigungen beinhalten.
3. Entfernen Sie die Session Manager-Berechtigungen (`ssm:StartSession`) aus den IAM-Richtlinien der Pilotbenutzer.
4. Schulen Sie Pilotbenutzer im neuen Workflow für Zugriffsanforderungen.
5. Überwachen Sie das Pilotprojekt auf Probleme und holen Sie Feedback ein.
6. Passen Sie die Richtlinien und Verfahren auf der Grundlage der Ergebnisse des Pilotprojekts an.

Beispiel für eine Änderung der IAM-Richtlinie für Pilotbenutzer

Ursprüngliche Richtlinie mit Session Manager-Berechtigungen:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:StartSession",
        "ssm:ResumeSession",
        "ssm:TerminateSession"
      ],
      "Resource": "*"
    }
  ]
}
```

Geänderte Richtlinie für den just-in-time Knotenzugriff:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:StartAccessRequest",
        "ssm:GetAccessToken",
        "ssm:ResumeSession",
        "ssm:TerminateSession"
      ],
      "Resource": "*"
    }
  ]
}
```

Phase 4: Vollständige Migration

Entwickeln Sie einen Zeitplan für die stapelweise Migration der verbleibenden Benutzer und Knoten.

Testmethode

Führen Sie während des gesamten Migrationsprozesses die folgenden Tests durch:

- Richtlinienvalidierung – Stellen Sie sicher, dass die Genehmigungsrichtlinien für die vorgesehenen Knoten und Benutzer gelten.
- Workflow für Zugriffsanforderungen – Testen Sie den gesamten Workflow von der Zugriffsanforderung bis zur Sitzungseinrichtung sowohl für automatische Genehmigungsszenarien als auch für manuelle Genehmigungsszenarien.
- Benachrichtigungen – Stellen Sie sicher, dass Genehmiger Benachrichtigungen über konfigurierte Kanäle (E-Mail, Slack, Microsoft Teams) erhalten.
- Protokollierung und Überwachung – Stellen Sie sicher, dass Sitzungsprotokolle und Zugriffsanforderungen ordnungsgemäß erfasst und gespeichert werden.

Bewährte Methoden für eine erfolgreiche Migration

- Kommunizieren Sie frühzeitig und häufig — Informieren Sie die Benutzer über den Zeitplan für die Migration und die Vorteile des just-in-time Knotenzugriffs.
- Starten mit unkritischen Systemen – Beginnen Sie mit der Migration mit Entwicklungs- oder Testumgebungen, bevor Sie zur Produktion übergehen.
- Dokumentieren von allem – Führen Sie detaillierte Aufzeichnungen über Ihre Genehmigungsrichtlinien, IAM-Richtlinienänderungen und Konfigurationseinstellungen.
- Überwachen und Anpassen – Überwachen Sie kontinuierlich Zugriffsanforderungen und Genehmigungsworkflows und passen Sie die Richtlinien nach Bedarf an.
- Etablieren von Governance – Richten Sie einen Prozess ein, mit dem die Genehmigungsrichtlinien regelmäßig überprüft und aktualisiert werden, wenn sich Ihre Umgebung ändert.

just-in-timeZugriff mit Systems Manager deaktivieren

Das folgende Verfahren beschreibt, wie Sie den just-in-time Knotenzugriff deaktivieren. Nach dem Deaktivieren des just-in-time Knotenzugriffs können Benutzer in Ihrer Organisation

möglicherweise keine Verbindung zu Ihren Knoten herstellen, es sei denn, Sie haben bereits andere Verbindungsmethoden implementiert.

Um den just-in-time Knotenzugriff zu deaktivieren

1. Melden Sie sich beim delegierten Systems Manager-Administratorkonto für Ihre Organisation an.
2. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
3. Wählen Sie im Navigationsbereich Einstellungen aus.
4. Wählen Sie auf der Registerkarte Just-in-time Knotenzugriff die Option Deaktivieren aus.

Just-in-time Häufig gestellte Fragen zum Knotenzugriff

Wie wechsle ich vom Knotenzugriff Session Manager zum just-in-time Knotenzugriff?

Nachdem Sie die einheitliche Konsole eingerichtet und den just-in-time Knotenzugriff aktiviert haben, müssen Sie Ihre bestehenden IAM-Richtlinien ändern, um die Umstellung auf den just-in-time Knotenzugriff abzuschließen. Dazu gehören das Hinzufügen der erforderlichen Berechtigungen für den just-in-time Knotenzugriff und das Entfernen der Berechtigungen für den `StartSession` API-Vorgang fürSession Manager. Weitere Informationen zu IAM-Richtlinien für den just-in-time Knotenzugriff finden Sie unter just-in-timeZugriff mit Systems Manager einrichten.

Muss ich die einheitliche Konsole einrichten, um den just-in-time Knotenzugriff nutzen zu können?

Ja, die Einrichtung der einheitlichen Konsole ist eine Voraussetzung für den just-in-time Knotenzugriff. Nachdem Sie die einheitliche Konsole eingerichtet und den just-in-time Knotenzugriff aktiviert haben, gibt es jedoch mehrere Methoden, um eine Verbindung zu Ihren Knoten herzustellen. Sie können beispielsweise just-in-time Knotenzugriffssitzungen von der EC2 Amazon-Konsole aus starten und die AWS CLI. Weitere Informationen zum Einrichten der einheitlichen Konsole finden Sie unter Einrichten der einheitlichen Systems-Manager-Konsole für eine Organisation.

Sind mit dem just-in-time Knotenzugriff Kosten verbunden?

Systems Manager bietet eine kostenlose 30-Tage-Testversion für den just-in-time Knotenzugriff. Nach der Testversion fallen für den just-in-time Knotenzugriff Kosten an. Weitere Informationen finden Sie unter AWS Systems Manager – Preise.

Welchen Vorrang haben Richtlinien zur Genehmigung des just-in-time Knotenzugriffs?

Genehmigungsrichtlinien werden in der folgenden Reihenfolge ausgewertet:

1. Zugriffsverweigerung
2. Automatische Genehmigung
3. Manuell

Wie werden Richtlinien für die manuelle Genehmigung ausgewertet?

Just-in-time Beim Knotenzugriff wird immer die spezifischere Richtlinie für einen Knoten bevorzugt. Richtlinien für die manuelle Genehmigung werden in der folgenden Reihenfolge ausgewertet:

1. Spezifisches Tag
2. Alle Knoten

Was passiert, wenn für einen Knoten keine Genehmigungsrichtlinie gilt?

Um mithilfe des Knotenzugriffs eine Verbindung zu einem just-in-time Knoten herzustellen, muss für den Knoten eine Genehmigungsrichtlinie gelten. Wenn für einen Knoten keine Genehmigungsrichtlinien gelten, können Benutzer keinen Zugriff auf den Knoten anfordern.

Können mehrere Genehmigungsrichtlinien für ein Tag gelten?

Ein Tag kann in Ihren Genehmigungsrichtlinien nur einmal als Ziel verwendet werden.

Was passiert, wenn aufgrund von Tag-Überschneidungen mehrere Richtlinien für die manuelle Genehmigung für einen Knoten gelten?

Wenn mehrere Richtlinien für die manuelle Genehmigung für einen Knoten gelten, führt dies zu einem Konflikt und Benutzer können keinen Zugriff auf den Knoten beantragen. Denken Sie daran, wenn Sie Ihre Richtlinien für die manuelle Genehmigung erstellen. Einige Instances können je nach Fall nämlich mehrere Tags haben.

Kann ich den just-in-time Knotenzugriff verwenden, um Zugriff zu beantragen und Sitzungen auf Knoten über Konten und Regionen hinweg zu starten?

Just-in-time Der Knotenzugriff unterstützt das Anfordern von Zugriff auf und das Starten von Sitzungen auf Knoten im selben Konto und derselben Region wie der Anforderer.

Kann ich den just-in-time Knotenzugriff verwenden, um Zugriff zu beantragen und Sitzungen auf Knoten zu starten, die mit einer Hybridaktivierung registriert sind?

Ja, der just-in-time Knotenzugriff unterstützt das Anfordern von Zugriff auf und das Starten von Sitzungen auf Knoten, die mit einer Hybridaktivierung registriert sind. Der Knoten muss im selben Konto und in derselben Region wie der Anforderer registriert sein.

Diagnose und Abhilfemaßnahmen

Mithilfe der einheitlichen Systems-Manager-Konsole können Sie Probleme in Ihrer gesamten Flotte in einem einzigen Diagnosevorgang identifizieren. In Unternehmen können Sie dann mit einem einzigen Automation-Vorgang versuchen, die Problembehebung für alle oder nur für ausgewählte Ziele durchzuführen. Für eine Organisation können Sie als delegierter Kontoadministrator Ziele für alle Konten und Regionen auswählen. Wenn Sie in einem einzelnen Konto arbeiten, können Sie Ziele in einer einzelnen Region gleichzeitig auswählen.

Systems Manager kann verschiedene Arten von Bereitstellungsfehlern sowie fehlerhafte Konfigurationen diagnostizieren und Ihnen helfen, diese zu beheben. Systems Manager kann auch Amazon Elastic Compute Cloud (Amazon EC2)-Instances in Ihrem Konto oder Ihrer Organisation identifizieren, die Systems Manager nicht als verwalteten Knoten behandeln kann. Der EC2-Instance-Diagnoseprozess kann Probleme im Zusammenhang mit Fehlkonfigurationen für eine Virtual Private Cloud (VPC), in einer Domain Name Service (DNS)-Einstellung oder in einer Amazon Elastic Compute Cloud (Amazon EC2)-Sicherheitsgruppe identifizieren.

> **Note**
>
> Systems Manager unterstützt sowohl EC2-Instances als auch andere Maschinentypen in einer Hybrid- und Multi-Cloud-Umgebung als verwaltete Knoten. Um ein verwalteter Knoten zu sein, muss AWS Systems Manager-Agent (SSM Agent) auf der Maschine installiert sein und Systems Manager muss berechtigt sein, Aktionen auf der Maschine auszuführen.
> Für EC2-Instances kann diese Berechtigung auf Kontoebene mithilfe einer AWS Identity and Access Management (IAM)-Rolle oder auf Instance-Ebene mithilfe eines Instance-Profils bereitgestellt werden. Weitere Informationen finden Sie unter Konfigurieren von erforderlichen Instance-Berechtigungen für Systems Manager.
> Für Nicht-EC2-Maschinen wird diese Berechtigung mithilfe einer IAM-Servicerolle bereitgestellt. Weitere Informationen finden Sie unter Erstellen Sie die für Systems Manager in Hybrid- und Multi-Cloud-Umgebungen erforderliche IAM-Servicerolle.

Bevor Sie beginnen

Um die Diagnose- und Problembehebungsfunktion zur Erkennung nicht verwalteter EC2-Instances verwenden zu können, müssen Sie zunächst Ihre Organisation oder Ihr Konto in die einheitliche Systems-Manager-Konsole einbinden. Während dieses Vorgangs müssen Sie die Option wählen, IAM-Rollen und verwaltete Richtlinien zu erstellen, die für diese Operationen erforderlich sind. Weitere Informationen finden Sie unter Einrichten der einheitlichen Systems-Manager-Konsole für eine Organisation.

Mithilfe der folgenden Themen können Sie bestimmte häufig auftretende Typen fehlgeschlagener Bereitstellungen, schwankender Konfigurationen und nicht verwalteter EC2-Instances identifizieren und beheben.

Themen

- Diagnose und Behebung fehlgeschlagener Bereitstellungen
- Diagnose und Behebung von Konfigurationsabweichungen
- Diagnose und Behebung nicht verwalteter Amazon-EC2-Instances in Systems Manager
- Arten von Runbook-Aktionen mit Auswirkungen auf die Behebung
- Ausführungsfortschritt und Verlauf von Behebungen in Systems Manager anzeigen

Diagnose und Behebung fehlgeschlagener Bereitstellungen

Systems Manager kann die folgenden Arten von fehlgeschlagenen Bereitstellungen diagnostizieren und Ihnen dann bei der Behebung helfen:

- Kerneinrichtung für Mitgliedskonten von Organisationen
- Kerneinrichtung für delegiertes Administratorkonto
- Kerneinrichtung für Ihr Konto

Gehen Sie wie folgt vor, um zu versuchen, diese Art von Problemen zu beheben.

Um fehlgeschlagene Bereitstellungen zu diagnostizieren und zu beheben

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Diagnose und Behebung aus.

3. Wählen Sie die Registerkarte Bereitstellungsprobleme.
4. Sehen Sie sich im Abschnitt Fehlgeschlagene Bereitstellungen die Liste der Ergebnisse für fehlgeschlagene Bereitstellungen an.
5. Wählen Sie in der Spalte Einrichtungsschritt den Namen eines Ergebnisses aus, um weitere Informationen zu dem Problem zu erhalten. Zum Beispiel: Kerneinrichtung für Mitgliedskonten von Organisationen.
6. Auf der Detailseite für diese fehlgeschlagene Bereitstellung können Sie eine Liste der Konten und die Anzahl der einzelnen Regionen einsehen, in denen Bereitstellungsfehler aufgetreten sind.
7. Wählen Sie eine Konto-ID aus, um Informationen über die Gründe für Fehler in diesem Konto anzuzeigen.
8. Prüfen Sie im Bereich Fehlerhafte Regionen die unter Statusgrund angegebenen Informationen. Diese Informationen können einen Grund für die fehlgeschlagene Bereitstellung angeben und so Aufschluss darüber geben, welche Konfigurationsänderungen vorgenommen werden müssen.
9. Wenn Sie die Bereitstellung erneut versuchen möchten, ohne Änderungen an der Konfiguration vorzunehmen, wählen Sie Erneut bereitstellen.

Diagnose und Behebung von Konfigurationsabweichungen

Systems Manager kann die folgenden Typen veränderter Konfigurationen diagnostizieren und Ihnen dann helfen, sie zu korrigieren:

- Kerneinrichtung für Mitgliedskonten von Organisationen
- Kerneinrichtung für delegiertes Administratorkonto
- Kerneinrichtung für Ihr Konto

Gehen Sie wie folgt vor, um zu versuchen, diese Arten von fehlerhaften Konfigurationen zu korrigieren.

So diagnostizieren und beheben Sie Konfigurationsabweichungen

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Diagnose und Behebung aus.
3. Wählen Sie die Registerkarte Bereitstellungsprobleme.

4. Sehen Sie sich im Abschnitt Konfigurationsabweichungen die Liste der Ergebnisse für fehlgeschlagene Bereitstellungen an.

   –oder–

   Um eine neue Diagnose durchzuführen, wählen Sie Abweichung finden.
5. Wählen Sie in der Spalte Einrichtungsschritt den Namen eines Ergebnisses aus, um weitere Informationen zu dem Problem zu erhalten. Zum Beispiel: Kerneinrichtung für Mitgliedskonten von Organisationen.
6. Auf der Detailseite für diese fehlgeschlagene Bereitstellung können Sie eine Liste der Konten und die Anzahl der einzelnen Regionen einsehen, in denen es zu Konfigurationsabweichungen gekommen ist.
7. Wählen Sie eine Konto-ID aus, um Informationen über den Grund für Konfigurationsabweichungen in diesem Konto anzuzeigen.
8. Im Bereich Ressourcenabweichungen werden in der Spalte Ressource die Namen der Ressourcen aufgeführt, bei denen es zu Abweichungen gekommen ist. In der Spalte Abweichungs-Typ wird angegeben, ob die Ressource geändert oder gelöscht wurde.
9. Um die beabsichtigte Konfiguration erneut bereitzustellen, wählen Sie Erneut bereitstellen.

Diagnose und Behebung nicht verwalteter Amazon-EC2-Instances in Systems Manager

Um Sie bei der Verwaltung Ihrer Amazon Elastic Compute Cloud (Amazon EC2)-Instances mit Systems Manager zu unterstützen, können Sie die vereinheitlichte Systems-Manager-Konsole verwenden, um Folgendes zu tun:

1. Führen Sie einen manuellen oder geplanten Diagnoseprozess durch, um zu ermitteln, welche EC2-Instances in Ihrem Konto oder Ihrer Organisation derzeit nicht durch Systems Manager verwaltet werden.
2. Identifizieren Sie Netzwerk- oder andere Probleme, die Systems Manager daran hindern, die Verwaltung der Instances zu übernehmen.
3. Führen Sie eine Automation-Ausführung durch, um das Problem automatisch zu beheben, oder greifen Sie auf Informationen zu, die Sie bei der manuellen Behebung des Problems unterstützen.

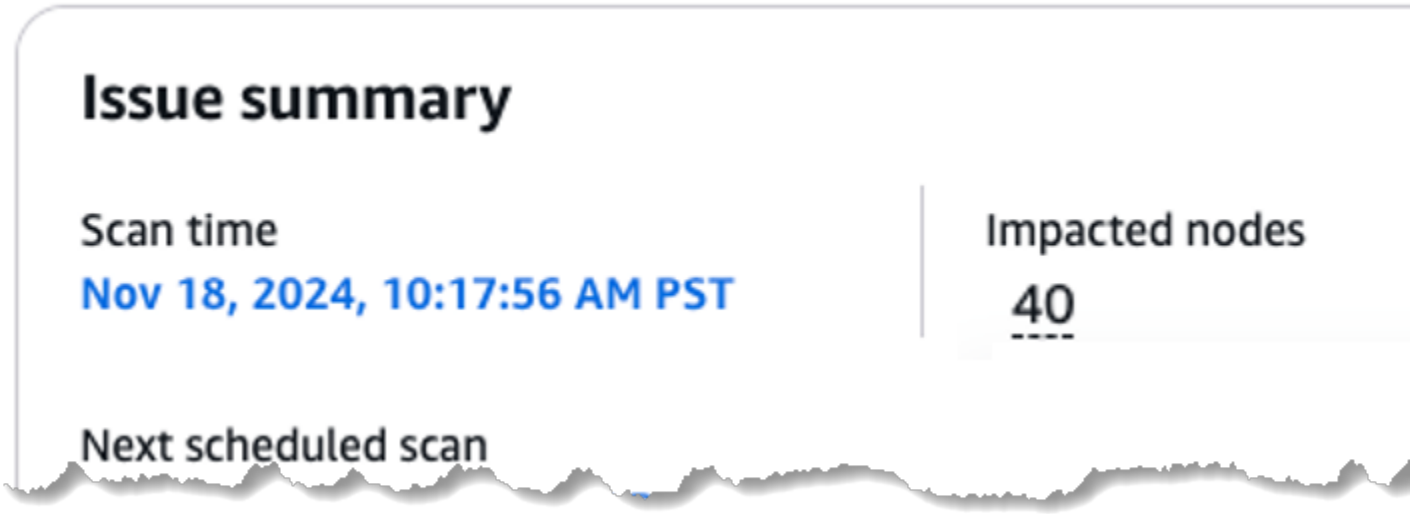
Verwenden Sie die Informationen in den folgenden Themen, um Probleme zu diagnostizieren und zu beheben, die Systems Manager daran hindern, Ihre EC2-Instances zu verwalten.

Wie Systems Manager die betroffenen Knoten für die Liste „Probleme mit nicht verwalteten EC2-Instances“ zählt

Die Anzahl der Knoten, die auf der Registerkarte Probleme mit nicht verwalteten EC2-Instances als nicht verwaltet gemeldet wurden, entspricht der Gesamtzahl der Instances mit einem der folgenden Statuswerte zum Zeitpunkt des Diagnosescans:

- `Running`
- `Stopped`
- `Stopping`

Diese Zahl wird im Bereich Problemübersicht als Betroffene Knoten gemeldet. In der folgenden Abbildung ist die Anzahl der betroffenen Knoten dargestellt, die derzeit nicht von Systems Manager verwaltet werden, `40`.



Im Gegensatz zu dem Bericht über nicht verwaltete EC2-Instances auf der Seite Knoteninformationen überprüfen ist diese Anzahl von EC2-Instances nicht dynamisch. Sie stellt die Ergebnisse dar, die während des letzten gemeldeten Diagnosescans gemacht wurden und als Wert für die Scanzeit angezeigt werden. Wir empfehlen daher, regelmäßig einen Diagnosescan für nicht verwaltete EC2-Instances durchzuführen, um die gemeldete Anzahl der betroffenen Knoten auf dem neuesten Stand zu halten.

Informationen zur Anzahl nicht verwalteter Instances finden Sie auf der Seite Knoten-Erkenntnisse überprüfen unter Was ist eine nicht verwaltete Instanz? im Thema Überprüfen von Erkenntnissen.

Themen

- Kategorien von diagnostizierbaren Problemen mit nicht verwalteten EC2-Instances
- Ausführen einer Diagnose und optionaler Problembehebung für nicht verwaltete EC2-Instances
- Planung eines wiederkehrenden Scans für nicht verwaltete EC2-Instances

Kategorien von diagnostizierbaren Problemen mit nicht verwalteten EC2-Instances

In diesem Thema werden die Hauptkategorien von EC2-Managementproblemen sowie die spezifischen Probleme in jeder Kategorie aufgeführt, bei deren Diagnose und Behebung Systems Manager Ihnen helfen kann. Beachten Sie, dass Systems Manager bei einigen Problemen zwar das Problem identifizieren kann, aber keine automatische Behebung anbietet. In diesen Fällen werden Sie von der Systems-Manager-Konsole zu Informationen weitergeleitet, die Ihnen helfen, ein Problem manuell zu lösen.

Der Diagnoseprozess untersucht jede Gruppe von EC2-Instances gleichzeitig entsprechend der Virtual Private Cloud (VPC), zu der sie gehören.

Problemtypen

- Problemkategorie: Konfiguration von Sicherheitsgruppen und HTTPS-Kommunikation
- Problemkategorie: Konfiguration von DNS- oder DNS-Hostnamen
- Problemkategorie: VPC-Endpunktkonfiguration

Problemkategorie: Konfiguration von Sicherheitsgruppen und HTTPS-Kommunikation

Bei einem Diagnosevorgang wird möglicherweise festgestellt, dass SSM Agent keine Kommunikation mit dem Systems-Manager-Service über HTTPS möglich ist. In diesen Fällen können Sie sich dafür entscheiden, ein Automation-Runbook auszuführen, das versucht, Sicherheitsgruppen zu aktualisieren, die an die Instances angehängt sind.

> **Note**
>
> Gelegentlich kann Systems Manager diese Probleme möglicherweise nicht automatisch beheben, aber Sie können die betroffenen Sicherheitsgruppen manuell bearbeiten.

Unterstützte Problemtypen

- Instance-Sicherheitsgruppe: Ausgehender Datenverkehr ist auf Port 443 nicht zulässig
- **`ssm`**-Sicherheitsgruppe des VPC-Endpunkts: Eingehender Datenverkehr ist auf Port 443 nicht zulässig
- **`ssmmessages`**-Sicherheitsgruppe des VPC-Endpunkts: Eingehender Datenverkehr ist auf Port 443 nicht zulässig
- **`ec2messages`**-Sicherheitsgruppe des VPC-Endpunkts: Eingehender Datenverkehr ist auf Port 443 nicht zulässig

Weitere Informationen finden Sie unter Die Eingangsregeln für Endpunkt-Sicherheitsgruppen überprüfen im Thema Fehlerbehebung für SSM Agent.

Problemkategorie: Konfiguration von DNS- oder DNS-Hostnamen

Bei einem Diagnosevorgang wird möglicherweise festgestellt, dass Domain Name System (DNS) oder DNS-Hostnamen für die VPC nicht richtig konfiguriert sind. In diesen Fällen können Sie ein Automation-Runbook ausführen, das versucht, die Attribute `enableDnsSupport` und `enableDnsHostnames` der betroffenen VPC zu aktivieren.

Unterstützte Problemtypen

- Die DNS-Unterstützung ist in einer VPC deaktiviert.
- Ein DNS-Hostname ist in einer VPC deaktiviert.

Weitere Informationen finden Sie unter Ihre VPC-DNS-bezogenen Attribute überprüfen im Thema Fehlerbehebung für SSM Agent.

Problemkategorie: VPC-Endpunktkonfiguration

Bei einem Diagnosevorgang wird möglicherweise festgestellt, dass VPC-Endpunkte nicht richtig für die VPC konfiguriert sind.

Wenn VPC-Endpoints, die von SSM Agent erforderlich sind, nicht existieren, versucht Systems Manager, ein Automation-Runbook auszuführen, um die VPC-Endpoints zu erstellen und sie einem Subnetz in jeder relevanten regionalen Availability Zone (AZ) zuzuordnen. Wenn die erforderlichen Endpunkte in VPC vorhanden sind, aber keinem Subnetz zugeordnet sind, in dem das Problem auftritt, ordnet das Runbook die VPC-Endpunkte dem betroffenen Subnetz zu.

> **Note**
>
> Systems Manager unterstützt nicht die Behebung aller falsch konfigurierten VPC-Endpunktprobleme. In diesen Fällen leitet Sie Systems Manager zu manuellen Abhilfemaßnahmen weiter, anstatt ein Automation-Runbook auszuführen.

Unterstützte Problemtypen

- Es wurde kein `ssm.region.amazonaws.com`-Endpunkt für PrivateLink gefunden.
- Es wurde kein `ssmmessages.region.amazonaws.com`-Endpunkt für PrivateLink gefunden.
- Es wurde kein `ec2messages.region.amazonaws.com`-Endpunkt für PrivateLink gefunden.

Diagnostizierbare Problemtypen

Systems Manager kann die folgenden Problemtypen diagnostizieren, aber derzeit ist kein Runbook zur Behebung dieser Probleme verfügbar. Sie können Ihre Konfiguration für diese Probleme manuell bearbeiten.

- Das Subnetz einer Instance ist nicht mit einem `ssm.region.amazonaws.com`-Endpunkt verbunden.
- Das Subnetz einer Instance ist nicht mit einem `ssmmessages.region.amazonaws.com`-Endpunkt verbunden.
- Das Subnetz einer Instance ist nicht mit einem `ec2messages.region.amazonaws.com`-Endpunkt verbunden.

Weitere Informationen finden Sie unter Ihre VPC-Konfiguration überprüfen im Thema Fehlerbehebung für SSM Agent.

Ausführen einer Diagnose und optionaler Problembehebung für nicht verwaltete EC2-Instances

Verwenden Sie das folgende Verfahren, um die netzwerk- und VPC-bezogenen Probleme zu diagnostizieren, die Systems Manager möglicherweise daran hindern, Ihre EC2-Instances zu verwalten.

Mit der Diagnose können Probleme der folgenden Typen erkannt und gruppiert werden:

- Probleme mit der Netzwerkkonfiguration – Arten von Netzwerkproblemen, die die Kommunikation von EC2-Instances mit dem Systems-Manager-Service in der Cloud verhindern könnten. Für diese Probleme sind möglicherweise Behebungsmaßnahmen verfügbar. Weitere Informationen zu Netzwerkkonfigurationsproblemen finden Sie unter Kategorien von diagnostizierbaren Problemen mit nicht verwalteten EC2-Instances.
- Unbekannte Probleme – Eine Liste mit Ergebnissen für Fälle, in denen der Diagnosevorgang nicht ermitteln konnte, warum EC2-Instances nicht mit dem Systems-Manager-Service in der Cloud kommunizieren können.

Um eine Diagnose und Problembehebung für nicht verwaltete EC2-Instances durchzuführen

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Diagnose und Behebung aus.
3. Wählen Sie die Problemregisterkarte Unverwaltete EC2-Instances aus.
4. Wählen Sie im Abschnitt Problemzusammenfassung die Option Neue Diagnose ausführen aus.

   –oder–

   Wenn Sie zum ersten Mal nicht verwaltete EC2-Probleme diagnostizieren, wählen Sie im Abschnitt Diagnose unverwalteter EC2-Instances die Option Ausführen aus.

   > **Tip**
   >
   > Wählen Sie während der Ausführung der Diagnose Fortschritt anzeigen oder Ausführungen anzeigen aus, um den aktuellen Status der Ausführung zu überwachen. Weitere Informationen finden Sie unter Ausführungsfortschritt und Verlauf von Behebungen in Systems Manager anzeigen.

5. Nachdem die Diagnose abgeschlossen ist, gehen Sie wie folgt vor:

   - Für alle Probleme, die im Abschnitt Unbekannte Probleme gemeldet wurden, klicken Sie auf den Link Weitere Informationen, um Informationen zur Lösung des Problems zu erhalten.
   - Bei Problemen, die im Abschnitt Netzwerkkonfigurationsprobleme gemeldet wurden, fahren Sie mit dem nächsten Schritt fort.

6. Wählen Sie in der Liste der Suchtypen in der Spalte Empfehlungen für ein bestimmtes Problem den Link aus, z. B. 2 Empfehlungen.

7. Wählen Sie im sich öffnenden Bereich Empfehlungen aus den verfügbaren Abhilfemaßnahmen aus:

   - Weitere Informationen – Öffnen Sie ein Thema mit Informationen zur manuellen Lösung eines Problems.
   - Runbook anzeigen – Öffnen Sie einen Bereich mit Informationen zum Automation-Runbook, das Sie ausführen können, um das Problem mit Ihren EC2-Instances zu beheben, sowie Optionen zum Generieren einer Vorschau der Aktionen, die das Runbook ausführen würde. Fahren Sie mit dem nächsten Schritt fort.

8. Führen Sie im Bereich Runbooks Folgendes aus:

   a. Eine Beschreibung des Dokuments finden Sie im Inhalt, der einen Überblick über die Maßnahmen bietet, die das Runbook zur Behebung Ihrer Probleme mit nicht verwalteten EC2-Instances ergreifen kann. Wählen Sie Schritte anzeigen, um eine Vorschau der einzelnen Aktionen anzuzeigen, die das Runbook ausführen würde.

   b. Führen Sie für Targets (Ziele) Folgendes aus:

      - Wenn Sie Problembehebungen für eine Organisation verwalten, geben Sie unter Konten an, ob dieses Runbook auf alle Konten oder nur auf eine Untergruppe der von Ihnen ausgewählten Konten abzielt.
      - Geben Sie für Regionen an, ob dieses Runbook auf alle AWS-Regionen in Ihrem Kontos oder Ihrer Organisation oder nur auf eine Teilmenge der von Ihnen ausgewählten Regionen abzielt.

   c. Lesen Sie sich die Informationen sorgfältig durch, um eine Runbook-Vorschauversion zu erhalten. In diesen Informationen wird erläutert, welchen Umfang und welche Auswirkungen es hätte, wenn Sie das Runbook ausführen würden.

      > **Note**
      >
      > Die Entscheidung, das Runbook auszuführen, würde mit Gebühren verbunden sein. Lesen Sie die Vorschauinformationen sorgfältig durch, bevor Sie entscheiden, ob Sie fortfahren möchten.

      Der Inhalt der Runbook-Vorschau enthält die folgenden Informationen:

      - In wie vielen Regionen würde der Runbook-Vorgang stattfinden.

- (Nur Organisationen) In wie vielen Organisationseinheiten (OUs) der Vorgang ausgeführt werden würde.
- Die Arten von Aktionen, die ergriffen werden würden, und wie viele davon.

  Zu den Aktionstypen gehören Folgende:

  - Mutation: Ein Runbook-Schritt würde Änderungen an den Zielen vornehmen, die Ressourcen erstellen, ändern oder löschen.
  - Nicht mutierend: Ein Runbook-Schritt würde Daten über Ressourcen abrufen, aber keine Änderungen an ihnen vornehmen. Diese Kategorie umfasst im Allgemeinen `Describe*`, `List*`, `Get*` und ähnliche schreibgeschützte API-Aktionen.
  - Unbestimmt: Ein unbestimmter Schritt ruft Ausführungen auf, die von einem anderen Orchestrierungsservice wie AWS Lambda, AWS Step Functions oder AWS Systems Manager Run Command durchgeführt werden. Ein unbestimmter Schritt kann auch eine Drittanbieter-API aufrufen. Systems Manager Automation kennt das Ergebnis der Orchestrierungsprozesse oder API-Ausführungen von Drittanbietern nicht, sodass die Ergebnisse der Schritte unbestimmt sind.

d. An dieser Stelle können Sie eine der folgenden Aktionen auswählen:

- Beenden Sie das Runbook und führen Sie es nicht aus.
- Wählen Sie Ausführen, um das Runbook mit den Optionen auszuführen, die Sie bereits ausgewählt haben.

Wenn Sie den Vorgang ausführen möchten, wählen Sie Fortschritt anzeigen oder Ausführungen anzeigen, um den aktuellen Status der Ausführung zu überwachen. Weitere Informationen finden Sie unter Ausführungsfortschritt und Verlauf von Behebungen in Systems Manager anzeigen.

Planung eines wiederkehrenden Scans für nicht verwaltete EC2-Instances

Sie können einen On-Demand-Scan für Amazon-EC2-Instances in Ihrem Konto oder Ihrer Organisation ausführen, die Systems Manager aufgrund verschiedener Konfigurationsprobleme nicht verwalten kann. Sie können diesen Scan auch so planen, dass er nach einem Zeitplan automatisch ausgeführt wird.

Um einen wiederkehrenden Scan für nicht verwaltete EC2-Instances zu planen

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Diagnose und Behebung aus.
3. Wählen Sie die Problemregisterkarte Unverwaltete EC2-Instances aus.
4. Aktivieren Sie im Abschnitt Unverwaltete EC2-Instances diagnostizieren die Option Wiederkehrende Diagnose planen.
5. Wählen Sie unter Diagnosehäufigkeit aus, ob die Diagnose einmal täglich oder einmal pro Woche ausgeführt werden soll.
6. (Optional) Geben Sie unter Startzeit eine Uhrzeit im 24-Stunden-Format ein, zu der die Diagnose beginnen soll. Geben Sie zum Beispiel für 8:15 Uhr abends **`20:15`** ein.

   Die von Ihnen eingegebene Zeit bezieht sich auf Ihre aktuelle lokale Zeitzone.

   Wenn Sie keine Uhrzeit angeben, wird der Diagnosescan sofort ausgeführt. Systems Manager plant außerdem, dass der Scan in der Zukunft zum aktuellen Zeitpunkt ausgeführt wird. Wenn Sie eine Uhrzeit angeben, wartet Systems Manager darauf, den Diagnosescan zum angegebenen Zeitpunkt auszuführen.
7. Wählen Sie Ausführen. Die Diagnose wird sofort ausgeführt, aber auch nach dem von Ihnen angegebenen Zeitplan.

Arten von Runbook-Aktionen mit Auswirkungen auf die Behebung

Systems Manager kann Diagnosevorgänge ausführen, mit denen bestimmte Arten von fehlgeschlagenen Bereitstellungen und fehlerhaften Konfigurationen sowie bestimmte Arten von Konfigurationsproblemen erkannt werden, die Systems Manager daran hindern, EC2-Instances zu verwalten. Die Ergebnisse der Diagnose können Empfehlungen für Automation-Runbooks beinhalten, die Sie ausführen können, um ein Problem zu beheben. Weitere Informationen zu diesen Diagnosevorgängen finden Sie in den folgenden Themen:

- Diagnose und Behebung fehlgeschlagener Bereitstellungen
- Diagnose und Behebung von Konfigurationsabweichungen
- Diagnose und Behebung nicht verwalteter Amazon-EC2-Instances in Systems Manager

Wenn Systems Manager ein Problem identifiziert, das möglicherweise durch die Ausführung eines Automation-Runbooks auf den betroffenen Ressourcen behoben werden kann, erhalten Sie eine Ausführungsvorschau. Die Ausführungsvorschau enthält Informationen über die Arten von Änderungen, die die Runbook-Ausführung an Ihren Zielen vornehmen würde. Zu diesen Informationen gehört, wie viele der drei Arten von Änderungen in der Diagnose identifiziert wurden.

Diese Änderungstypen lauten wie folgt:

- `Mutating`: Ein Runbook-Schritt würde Änderungen an den Zielen vornehmen, die Ressourcen erstellen, ändern oder löschen.
- `Non-Mutating`: Ein Runbook-Schritt würde Daten über Ressourcen abrufen, aber keine Änderungen an ihnen vornehmen. Diese Kategorie umfasst im Allgemeinen `Describe*`, `List*`, `Get*` und ähnliche schreibgeschützte API-Aktionen.
- `Undetermined`: Ein unbestimmter Schritt ruft Ausführungen auf, die von einem anderen Orchestrierungsservice ausgeführt werden, wie AWS Lambda, AWS Step Functions oder Run Command, ein Tool in AWS Systems Manager. Ein unbestimmter Schritt kann auch eine Drittanbieter-API aufrufen oder ein Python- oder PowerShell-Skript ausführen. Systems Manager Automation kann nicht erkennen, was das Ergebnis der Orchestrierungsprozesse oder API-Ausführungen von Drittanbietern sein würde, und kann sie daher nicht auswerten. Die Ergebnisse dieser Schritte müssten manuell überprüft werden, um ihre Auswirkungen zu ermitteln.

  In der folgenden Tabelle finden Sie Informationen zur Art der Auswirkungen der unterstützten Automation-Aktionen.

Arten der Auswirkungen der unterstützten Behebungsmaßnahmen

In der Tabelle sind die Arten der Auswirkungen (mutierend, nicht mutierend und unbestimmt) verschiedener Aktionen aufgeführt, die in ein Behebungs-Runbook aufgenommen werden können.

Aktion[1]	Art der Auswirkung
aws:approve	Nicht Mutation
aws:assertAwsResourceProperty	Nicht Mutation
aws:branch	Nicht Mutation
aws:changeInstanceState	Mutation

Aktion[1]	Art der Auswirkung
aws:copyImage	Mutation
aws:createImage	Mutation
aws:createStack	Mutation
aws:createTags	Mutation
aws:deleteImage	Mutation
aws:deleteStack	Mutation
aws:executeAutomation	Unbestimmt
aws:executeAwsApi	Unbestimmt
aws:executeScript	Unbestimmt
aws:executeStateMachine	Unbestimmt
aws:invokeLambdaFunction	Unbestimmt
aws:invokeWebhook	Unbestimmt
aws:loop	Variiert. Hängt von den Aktionen in der Schleife ab.
aws:pause	Nicht Mutation
aws:runCommand	Unbestimmt
aws:runInstances	Mutation
aws:sleep	Nicht Mutation
aws:updateVariable	Mutation
aws:waitForAwsResourceProperty	Nicht Mutation

[1]Weitere Informationen zu Automation-Aktionen finden Sie unter Systems Manager Automation Aktionen-Referenz.

Ausführungsfortschritt und Verlauf von Behebungen in Systems Manager anzeigen

Sie können eine Liste aller laufenden und abgeschlossenen Behebungsvorgänge anzeigen, die mit der Funktion Diagnose und Behebung in Systems Manager durchgeführt wurden.

Die Daten in der Liste mit dem Ausführungsverlauf enthalten die folgenden Informationstypen:

- Die Art der Ausführung, `Diagnosis` oder `Remediation`.
- Der Ausführungsstatus, z. B. `Success` oder `Failed`.
- Die Zeiten, zu denen die Ausführung begonnen und beendet wurde.

So können Sie den Ausführungsfortschritt und den Verlauf der Behebungen einsehen

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Diagnose und Behebung aus.
3. Wählen Sie Ausführungen anzeigen aus.

   > **Tip**
   >
   > Während einer Ausführung können Sie auch Fortschritt anzeigen wählen, um die Seite Ausführungsverlauf zu öffnen.

4. (Optional) Geben Sie in das Suchfeld
   ( )
   einen Ausdruck ein, um die Ausführungsliste einzugrenzen, z. B. **EC2** oder **VPC**.
5. (Optional) Um zusätzliche Details zu einer Ausführung anzuzeigen, wählen
   Sie in der Spalte Ausführungsname einen Vorgangsnamen aus, z. B. AWS-DiagnoseUnmanagedEC2NetworkIssues.

   Im Detailbereich können Sie Informationen zu allen Schritten, die während des Vorgangs versucht wurden, sowie zu allen Eingaben und Ausgaben für die Ausführung überprüfen.

Systems-Manager-Einstellungen anpassen

Die Optionen auf den Einstellungsseiten aktivieren und konfigurieren Funktionen in der vereinheitlichten Systems-Manager-Konsole. Die angezeigten Optionen hängen von dem Konto ab, mit dem Sie angemeldet sind, und davon, ob Sie Systems Manager bereits eingerichtet haben oder nicht.

> **Note**
>
> Die Optionen auf der Seite Einstellungen haben keinen Einfluss auf die Systems-Manager-Tools (früher Funktionen genannt).

Kontoeinrichtungseinstellungen

Wenn Systems Manager aktiviert ist und Sie mit einem Konto angemeldet sind, das kein Mitglied von Organizations ist, oder wenn der delegierte Administrator Ihr Organisationskonto nicht zu Systems Manager hinzugefügt hat, wird auf der Kontoeinrichtungsseite die Option Systems Manager deaktivieren angezeigt. Die Deaktivierung von Systems Manager bedeutet, dass Systems Manager nicht das vereinheitlichte Konsolenerlebnis anzeigt. Alle Systems-Manager-Tools funktionieren weiterhin.

Organisationseinstellungen

Auf der Registerkarte Organisationseinstellungen wird im Bereich Heimatregion die Region angezeigt, die bei der Einrichtung als Heimatregion AWS-Region ausgewählt wurde. In Umgebungen mit mehreren Konten und Regionen, in denen verwendet wird AWS Organizations, aggregiert Systems Manager automatisch Knotendaten aus allen Konten und Regionen in der Heimatregion. Wenn Sie Daten auf diese Weise aggregieren, können Sie Knotendaten über Konten und Regionen hinweg an einem einzigen Ort anzeigen.

> **Note**
>
> Wenn Sie die Heimatregion ändern möchten, müssen Sie Systems Manager deaktivieren und erneut aktivieren. Um Systems Manager zu deaktivieren, wählen Sie Deaktivieren.

Im Bereich Organisationseinrichtung werden die AWS Organisationseinheiten angezeigt, die bei der Einrichtung AWS-Regionen ausgewählt wurden. Um zu ändern, welche Organisationseinheiten und Regionen Knotendaten in Systems Manager anzeigen, wählen Sie Bearbeiten. Weitere Informationen zum Festlegen von Systems Manager für Organizations finden Sie unter Einrichten von AWS Systems Manager.

Funktionskonfigurationen

Im Abschnitt Featurekonfigurationen können Sie wichtige Systems-Manager-Funktionen aktivieren und konfigurieren, die das Knotenmanagement in Ihrem Unternehmen verbessern. Diese Features sorgen zusammen für eine automatisierte Verwaltung, Compliance-Überwachung und Wartung Ihrer verwalteten Knoten.

Sie können diese Features bei der Erstinstallation von Systems Manager konfigurieren oder sie später auf der Seite Einstellungen ändern. Jedes Feature kann je nach den Anforderungen Ihres Unternehmens unabhängig aktiviert oder deaktiviert werden.

Standardkonfiguration für die Hostverwaltung

Die Standard-Host-Management-Konfiguration (DHMC) konfiguriert automatisch Amazon Elastic Compute Cloud (Amazon EC2) -Instances in Ihrer Organisation so, dass sie von Systems Manager verwaltet werden. Wenn diese Option aktiviert ist, stellt DHMC sicher, dass neue und bestehende EC2 Instanzen über die erforderlichen AWS Identity and Access Management (IAM-) Berechtigungen und Konfigurationen verfügen, um mit Systems Manager Manager-Diensten zu kommunizieren.

DHCM bietet die folgenden Vorteile:

- Automatische IAM-Rollenzuweisung — Stellt sicher, dass EC2 Instanzen über die erforderlichen IAM-Rollen und -Richtlinien verfügen, um als verwaltete Knoten zu fungieren
- Behebung von Abweichungen – Korrigiert automatisch Konfigurationsabweichungen, wenn Instances ihren Status als verwalteter Knoten verlieren
- Vereinfachtes Onboarding – Reduziert die Anzahl manueller Konfigurationsschritte für neue Instances
- Konsistente Konfiguration — Behält einheitliche Einstellungen für Ihre gesamte Flotte bei EC2

Konfiguration der Häufigkeit von Abweichungskorrekturen

Drift Remediation erkennt und korrigiert automatisch, wenn EC2 Instances ihre verwaltete Knotenkonfiguration verlieren. Sie können konfigurieren, wie oft Systems Manager nach Konfigurationsabweichungen sucht und diese behebt.

Konfigurieren Sie die Standardkonfiguration für die Host-Verwaltung wie folgt:

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Settings (Einstellungen).
3. Suchen Sie im Abschnitt Featurekonfigurationen nach Standard-Host-Management-Konfiguration.
4. Um DHMC zu aktivieren, schalten Sie den Schalter ein.
5. Wählen Sie unter Häufigkeit der Abweichungskorrektur aus, wie oft Systems Manager nach Konfigurationsabweichungen suchen und diese korrigieren soll:
   - Täglich – Abweichungen werden einmal täglich überprüft und behoben
   - Wöchentlich – Abweichungen werden einmal pro Woche überprüft und behoben
   - Monatlich – Abweichungen werden einmal pro Monat überprüft und behoben
6. Wählen Sie Speichern.

> **Note**
>
> Wenn Sie DHMC aktivieren, erstellt Systems Manager die erforderlichen IAM-Rollen und -Richtlinien in Ihrem Konto. Diese Rollen ermöglichen es EC2 Instanzen, mit Systems Manager Manager-Diensten zu kommunizieren. Weitere Informationen über die von DHMC erstellten IAM-Rollen finden Sie unter Verwalten von EC2-Instances mit Systems Manager.

Erfassung von Inventarmetadaten

Bei der Erfassung von Inventarmetadaten werden automatisch detaillierte Informationen über Ihre verwalteten Knoten gesammelt, einschließlich installierter Anwendungen, Netzwerkkonfigurationen, Systemupdates und anderer Systemmetadaten. Diese Informationen helfen Ihnen dabei, die Compliance zu gewährleisten, Sicherheitsanalysen durchzuführen und die Zusammensetzung Ihrer Infrastruktur zu verstehen.

Die Bestandserfassung bietet die folgenden Vorteile:

- Compliance-Überwachung – Verfolgen Sie installierte Software und Konfigurationen für die Erstellung von Compliance-Berichten
- Sicherheitsanalyse – Identifizieren Sie veraltete Software und potenzielle Sicherheitslücken
- Asset Management — Führen Sie eine up-to-date Bestandsaufnahme Ihrer Infrastruktur
- Abfragefunktionen – Verwenden Sie erfasste Daten mit Amazon Q Developer für Abfragen in natürlicher Sprache

Arten der erfassten Inventardaten

Wenn die Erfassung von Inventarmetadaten aktiviert ist, erfasst Systems Manager die folgenden Arten von Informationen von Ihren verwalteten Knoten:

- Anwendungen – Installierte Softwarepakete und Anwendungen
- Netzwerkkonfigurationen – Netzwerkschnittstellen, IP-Adressen und Netzwerkeinstellungen
- Systemupdates – Installierte Patches und verfügbare Updates
- Systemeigenschaften – Hardwarespezifikationen, Betriebssystemdetails und Systemkonfigurationen
- Services – Laufende Services und ihre Konfigurationen

Konfigurieren der Häufigkeit der Bestandserfassung

Sie können konfigurieren, wie häufig Systems Manager Inventar-Metadaten aus Ihren verwalteten Knoten erfasst. Eine häufigere Erfassung bietet mehr up-to-date Informationen, kann jedoch die Nutzung der AWS Dienste erhöhen.

Konfigurieren Sie die Erfassung von Inventar-Metadaten wie folgt:

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Settings (Einstellungen).
3. Suchen Sie im Abschnitt Feature-Konfigurationen nach Erfassung von Inventarmetadaten.
4. Um die Inventarerfassung zu aktivieren, schalten Sie den Schalter ein.
5. Wählen Sie unter Erfassungshäufigkeit aus, wie oft Systems Manager Inventardaten erfassen soll:

- Täglich – Inventardaten werden einmal täglich erfasst
- Wöchentlich – Inventardaten werden einmal pro Woche erfasst
- Monatlich – Inventardaten werden einmal pro Monat erfasst

6. Wählen Sie Speichern.

> **Important**
>
> Für die Inventarerfassung müssen verwaltete Knoten über die erforderlichen Berechtigungen zum Erfassen von Systeminformationen verfügen. Stellen Sie sicher, dass Ihre verwalteten Knoten über die entsprechenden IAM-Rollen und -Richtlinien verfügen. Weitere Informationen zu erforderlichen Berechtigungen finden Sie unter AWS Systems Manager-Bestand.

SSM Agent-Aktualisierungen

Automatische SSM Agent-Updates stellen sicher, dass auf Ihren verwalteten Knoten die neueste Version von SSM Agent ausgeführt wird. Wenn Sie den Agenten behalten, erhalten Sie up-to-date Zugriff auf die neuesten Funktionen, Sicherheitsverbesserungen und Bugfixes.

Automatische SSM Agent-Aktualisierungen bieten die folgenden Vorteile:

- Aktuelle Features – Zugriff auf neue Systems-Manager-Funktionen und Verbesserungen
- Sicherheitsupdates – Automatische Installation von Sicherheitspatches und Fixes
- Verbesserte Zuverlässigkeit – Bugfixes und Stabilitätsverbesserungen
- Geringerer Wartungsaufwand – Manuelle Agenten-Updates sind nicht mehr erforderlich

Konfigurieren von automatischen Agenten-Updates

Sie können konfigurieren, wie oft Systems Manager nach SSM Agent-Updates sucht und diese auf Ihren verwalteten Knoten installiert. Regelmäßige Updates sorgen für optimale Leistung und Sicherheit.

Konfigurieren Sie SSM Agent-Updates wie folgt:

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Settings (Einstellungen).
3. Suchen Sie im Abschnitt Feature-Konfigurationen nach SSM AgentUpdates.
4. Um automatische Updates zu aktivieren, schalten Sie den Schalter ein.
5. Wählen Sie unter Aktualisierungshäufigkeit aus, wie oft Systems Manager nach Agenten-Updates suchen und diese installieren soll:
   - Täglich – Einmal täglich wird nach Updates gesucht
   - Wöchentlich – Einmal pro Woche wird nach Updates gesucht
   - Monatlich – Einmal pro Monat wird nach Updates gesucht
6. Wählen Sie Speichern.

Einstellungen diagnostizieren und korrigieren

Die Einstellungen für Diagnose und Korrektur bestimmen, ob Systems Manager Ihre Knoten automatisch scannt, um sicherzustellen, dass sie mit Systems Manager kommunizieren können. Wenn diese Option aktiviert ist, wird die Funktion automatisch nach einem von Ihnen definierten Zeitplan ausgeführt. Die Funktion identifiziert, welche Knoten keine Verbindung zu Systems Manager herstellen können und warum. Dieses Feature bietet auch empfohlene Runbooks zur Behebung von Netzwerkproblemen und anderen Problemen, die verhindern, dass Knoten als verwaltete Knoten konfiguriert werden.

Planen eines wiederkehrenden Diagnosescans

Systems Manager kann verschiedene Arten von Bereitstellungsfehlern sowie fehlerhafte Konfigurationen diagnostizieren und Ihnen helfen, diese zu beheben. Systems Manager kann auch Amazon Elastic Compute Cloud (Amazon EC2) -Instances in Ihrem Konto oder Ihrer Organisation identifizieren, die Systems Manager nicht als verwalteten Knoten behandeln kann. Der EC2 Instanzdiagnoseprozess kann Probleme im Zusammenhang mit Fehlkonfigurationen für eine Virtual Private Cloud (VPC), in einer Domain Name Service (DNS) -Einstellung oder in einer Amazon Elastic Compute Cloud (Amazon EC2) -Sicherheitsgruppe identifizieren.

Um die Identifizierung von Knoten zu vereinfachen, die keine Verbindung zu Systems Manager herstellen können, können Sie mit dem Feature Wiederkehrende Diagnose planen einen wiederkehrenden Diagnosescan automatisieren. Mithilfe der Scans kann festgestellt werden, welche Knoten keine Verbindung zu Systems Manager herstellen können und warum. Gehen Sie wie folgt vor, um einen wiederkehrenden Diagnosescan Ihrer Knoten zu aktivieren und zu konfigurieren.

So planen Sie einen wiederkehrenden Diagnosescan

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Einstellungen und dann die Registerkarte Diagnose und Behebung aus.
3. Aktivieren Sie die Option Wiederkehrende Diagnose planen.
4. Wählen Sie unter Scanzeitraum aus, wie oft der Scan ausgeführt werden soll.
5. (Optional) Geben Sie unter Startzeit eine Uhrzeit im 24-Stunden-Format ein, zu der die Diagnose beginnen soll. Geben Sie zum Beispiel für 8:15 Uhr abends **`20:15`** ein.

   Die von Ihnen eingegebene Zeit bezieht sich auf Ihre aktuelle lokale Zeitzone.

   Wenn Sie keine Uhrzeit angeben, wird der Diagnosescan sofort ausgeführt. Systems Manager plant außerdem, dass der Scan in der Zukunft zum aktuellen Zeitpunkt ausgeführt wird.
   Wenn Sie eine Uhrzeit angeben, wartet Systems Manager darauf, den Diagnosescan zum angegebenen Zeitpunkt auszuführen.
6. Wählen Sie Speichern.
7. Wenn der Scan abgeschlossen ist, können Sie sich die Details anzeigen lassen, indem Sie im linken Navigationsbereich die Option Diagnose und Behebung auswählen.

Weitere Informationen zur Diagnose- und Problembehebungsfunktion finden Sie unter Diagnose und Abhilfemaßnahmen.

S3-Bucket-Verschlüsselung aktualisieren

Wenn Sie Systems Manager einbinden, erstellt Quick Setup einen Amazon Simple Storage Service (Amazon S3) -Bucket im delegierten Administratorkonto für AWS Organizations Setups. Bei der Einrichtung eines Einzelkontos wird der Bucket in dem Konto gespeichert, das gerade eingerichtet wird. Dieser Bucket wird verwendet, um die Metadaten zu speichern, die bei Diagnosescans generiert wurden.

Weitere Informationen zur Einrichtung der vereinheitlichten Systems-Manager-Konsole finden Sie unter Einrichten von AWS Systems Manager.

Standardmäßig werden Ihre Daten im Bucket mit einem AWS Key Management Service (AWS KMS) -Schlüssel verschlüsselt, der Ihnen AWS gehört und für Sie verwaltet.

Sie können wählen, ob Sie einen anderen AWS KMS Schlüssel für Ihre Bucket-Verschlüsselung verwenden möchten. Als weitere Alternative können Sie die serverseitige Verschlüsselung mit AWS KMS keys (SSE-KMS) unter Verwendung eines vom Kunden verwalteten Schlüssels (CMK) verwenden. Weitere Informationen finden Sie unter Arbeiten mit Amazon-S3-Buckets und Bucket-Richtlinien für Systems Manager.

Um einen anderen AWS KMS Schlüssel für die S3-Bucket-Verschlüsselung zu verwenden

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Einstellungen und dann die Registerkarte Diagnose und Behebung aus.
3. Wählen Sie im Bereich S3-Bucket-Verschlüsselung aktualisieren die Option Bearbeiten aus.
4. Aktivieren Sie das Kontrollkästchen Verschlüsselungseinstellungen anpassen (erweitert).
5. Wählen Sie unter AWS KMS Schlüssel auswählen den Amazon-Ressourcennamen (ARN) des Schlüssels aus oder geben Sie ihn ein.

   > **Tip**
   >
   > Um einen neuen Schlüssel zu erstellen, wählen Sie AWS KMS -Schlüssel erstellen.

6. Wählen Sie Speichern.

Arbeiten mit Amazon-S3-Buckets und Bucket-Richtlinien für Systems Manager

Quick Setup erstellt während des Onboardingprozesses für AWS Systems Manager einen Amazon Simple Storage Service-Bucket (Amazon S3) im delegierten Administratorkonto für Organisations-Einrichtungen. Bei der Einrichtung eines Einzelkontos wird der Bucket in dem Konto gespeichert, das gerade eingerichtet wird.

Sie können Systems Manager verwenden, um Diagnosevorgänge für Ihre Flotte durchzuführen, um Fälle von fehlgeschlagenen Bereitstellungen und fehlerhaften Konfigurationen zu identifizieren. Systems Manager kann auch Fälle erkennen, in denen Konfigurationsprobleme Systems Manager daran hindern, EC2-Instances in Ihrem Konto oder Ihrer Organisation zu verwalten. Die Ergebnisse dieser Diagnosevorgänge werden in diesem Amazon-S3-Bucket gespeichert, der sowohl durch eine Verschlüsselungsmethode als auch durch eine S3-Bucket-Richtlinie geschützt ist. Informationen

zu den Diagnosevorgängen, die Daten in diesen Bucket ausgeben, finden Sie unter Diagnose und Abhilfemaßnahmen.

Ändern der Bucket-Verschlüsselungsmethode

Standardmäßig verwendet der S3-Bucket eine serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3).

Sie können stattdessen eine serverseitige Verschlüsselung mit AWS KMS keys (SSE-KMS) unter Verwendung eines kundseitig verwalteten Schlüssels (CMK) als Alternative zu von Amazon S3 verwalteten Schlüsseln verwenden, wie unter Umstellen auf einen vom AWS KMS-Kunden verwalteten Schlüssel zur Verschlüsselung von S3-Ressourcen erklärt.

Inhalte der Bucketrichtlinien

Die Bucket-Richtlinie verhindert, dass sich Mitgliedskonten in einer Organisation gegenseitig entdecken. Lese- und Schreibberechtigungen für den Bucket sind nur für die Diagnose- und Behebungsrollen zulässig, die für Systems Manager erstellt wurden. Die Inhalte dieser vom System generierten Richtlinien finden Sie unter S3-Bucket-Richtlinien für die vereinheitlichte Systems-Manager-Konsole.

> **Warning**
>
> Wenn Sie die Standard-Bucket-Richtlinie ändern, können Mitgliedskonten in einer Organisation sich möglicherweise gegenseitig erkennen oder Diagnoseergebnisse für Instances in einem anderen Konto lesen. Wir empfehlen, äußerste Vorsicht walten zu lassen, wenn Sie diese Richtlinie ändern.

Themen

- Umstellen auf einen vom AWS KMS-Kunden verwalteten Schlüssel zur Verschlüsselung von S3-Ressourcen
- S3-Bucket-Richtlinien für die vereinheitlichte Systems-Manager-Konsole

Umstellen auf einen vom AWS KMS-Kunden verwalteten Schlüssel zur Verschlüsselung von S3-Ressourcen

Quick Setup erstellt während des Onboarding-Prozesses für die vereinheitlichte Systems-Manager-Konsole einen Amazon Simple Storage Service (Amazon S3)-Bucket im delegierten

Administratorkonto. In diesem Bucket werden die während der Ausführung des Reparatur-Runbooks generierten Diagnoseausgabedaten gespeichert. Standardmäßig verwendet der Bucket eine serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3).

Den Inhalt dieser Richtlinien finden Sie unter S3-Bucket-Richtlinien für die vereinheitlichte Systems-Manager-Konsole.

Als weitere Alternative zu AWS KMS key können Sie die serverseitige Verschlüsselung mit AWS KMS keys (SSE-KMS) unter Verwendung eines kundenseitig verwalteten Schlüssels (CMK) verwenden.

Führen Sie die folgenden Aufgaben aus, um Systems Manager für die Verwendung Ihres CMK zu konfigurieren.

Aufgabe 1: Fügen Sie einem vorhandenen CMK ein Tag hinzu

AWS Systems Manager verwendet Ihr CMK nur, wenn es mit dem folgenden Schlüssel-Wert-Paar gekennzeichnet ist:

- Schlüssel: `SystemsManagerManaged`
- Wert: `true`

Gehen Sie wie folgt vor, um Zugriff auf die Verschlüsselung des S3-Buckets mit Ihrem CMK zu gewähren.

Hinzufügen eines Tags zu Ihrem vorhandenen CMK

1. Öffnen Sie die AWS KMS-Konsole unter https://console.aws.amazon.com/kms.
2. Klicken Sie in linken Navigationsleiste auf Vom Kunden verwaltete Schlüssel.
3. Wählen Sie AWS KMS key aus, mit dem Sie AWS Systems Manager verwenden möchten.
4. Wählen Sie die Registerkarte Tags und dann Bearbeiten aus.
5. Wählen Sie Add tag.
6. Gehen Sie wie folgt vor:
   a. Geben Sie für Tag-Schlüssel **`SystemsManagerManaged`** ein.
   b. Geben Sie für Tag-Wert **`true`** ein.
7. Wählen Sie Speichern aus.

Aufgabe 2: Eine bestehende CMK-Schlüsselrichtlinie verändern

Gehen Sie wie folgt vor, um die KMS-Schlüsselrichtlinie Ihres CMK zu aktualisieren, sodass AWS Systems Manager-Rollen den S3-Bucket in Ihrem Namen verschlüsseln können.

Um eine bestehende CMK-Schlüsselrichtlinie zu ändern

1. Öffnen Sie die AWS KMS-Konsole unter https://console.aws.amazon.com/kms.
2. Klicken Sie in linken Navigationsleiste auf Vom Kunden verwaltete Schlüssel.
3. Wählen Sie AWS KMS key aus, mit dem Sie AWS Systems Manager verwenden möchten.
4. Wählen Sie im Tab Schlüsselrichtlinie die Option Bearbeiten aus.
5. Fügen Sie dem `Statement`-Feld die folgende JSON-Anweisung hinzu und ersetzen Sie die *`Platzhalterwerte`* durch Ihre eigenen Informationen.

   Stellen Sie sicher, dass Sie alle AWS-Konto-IDs, die in Ihrer Organisation in AWS Systems Manager integriert sind, dem `Principal`-Feld hinzufügen.

   Um den richtigen Bucket-Namen in der Amazon-S3-Konsole zu finden, suchen Sie im delegierten Administratorkonto den Bucket im Format `do-not-delete-ssm-`*`operational-account-id-home-region-disambiguator`*.

```
{
     "Sid": "EncryptionForSystemsManagerS3Bucket",
     "Effect": "Allow",
     "Principal": {
         "AWS": [
             "account-id-1",
             "account-id-2",
             ...
         ]
     },
     "Action": ["kms:Decrypt", "kms:GenerateDataKey"],
     "Resource": "*",
     "Condition": {
         "StringEquals": {
             "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::amzn-s3-demo-bucket"
         },
         "StringLike": {
             "kms:ViaService": "s3.*.amazonaws.com"
         },
         "ArnLike": {
```

```
            "aws:PrincipalArn": "arn:aws:iam::*:role/AWS-SSM-*"
        }
    }
}
```

> **Tip**
>
> Alternativ können Sie die CMK-Schlüsselrichtlinie mithilfe des Bedingungsschlüssels aws:PrincipalOrgID aktualisieren, um AWS Systems Manager Zugriff auf Ihr CMK zu gewähren.

Aufgabe 3: Geben Sie den CMK in den Systems-Manager-Einstellungen an

Gehen Sie nach Abschluss der beiden vorherigen Aufgaben wie folgt vor, um die S3-Bucket-Verschlüsselung zu ändern. Durch diese Änderung wird sichergestellt, dass der zugehörige Quick Setup-Konfigurationsprozess Berechtigungen für Systems Manager hinzufügen kann, Ihr CMK zu akzeptieren.

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Settings (Einstellungen).
3. Wählen Sie auf der Registerkarte Diagnose und Behebung im Abschnitt S3-Bucket-Verschlüsselung aktualisieren die Option Bearbeiten aus.
4. Aktivieren Sie das Kontrollkästchen Verschlüsselungseinstellungen anpassen (erweitert).
5. Wählen Sie im Suchfeld ( ) die ID eines vorhandenen Schlüssels aus, oder fügen Sie den ARN eines vorhandenen Schlüssels ein.
6. Wählen Sie Speichern.

S3-Bucket-Richtlinien für die vereinheitlichte Systems-Manager-Konsole

Dieses Thema umfasst die Amazon-S3-Bucket-Richtlinien, die von Systems Manager erstellt wurden, wenn Sie eine Organisation oder ein einzelnes Konto in die vereinheitlichte Systems-Manager-Konsole einbinden.

> **Warning**
>
> Wenn Sie die Standard-Bucket-Richtlinie ändern, können Mitgliedskonten in einer Organisation sich möglicherweise gegenseitig erkennen oder Diagnoseergebnisse für Instances in einem anderen Konto lesen. Wir empfehlen, äußerste Vorsicht walten zu lassen, wenn Sie diese Richtlinie ändern.

Amazon-S3-Bucket-Richtlinie für eine Organisation

Der Diagnose-Bucket wird mit der folgenden Standard-Bucket-Richtlinie erstellt, wenn eine Organisation in Systems Manager integriert wird.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "DenyHTTPRequests",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ],
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "false"
                }
            }
        },
        {
            "Sid": "DenyNonSigV4Requests",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
```

```
            ],
            "Condition": {
                "StringNotEquals": {
                    "s3:SignatureVersion": "AWS4-HMAC-SHA256"
                }
            }
        },
        {
            "Sid": "AllowAccessLog",
            "Effect": "Allow",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/access-logs/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "000000000000"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::amzn-s3-demo-bucket"
                }
            }
        },
        {
            "Sid": "AllowCrossAccountRead",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/actions/*/
${aws:PrincipalAccount}/*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": "organization-id"
                }
            }
        },
        {
            "Sid": "AllowCrossAccountWrite",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:PutObject",
                "s3:DeleteObject"
```

```
            ],
            "Resource": "arn:aws:s3:::bucket-name/actions/*/
${aws:PrincipalAccount}/*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": "organization-id"
                },
                "ArnLike": {
                    "aws:PrincipalArn": [
                        "arn:aws:iam::*:role/AWS-SSM-
DiagnosisExecutionRole-operational-account-id-home-region",
                        "arn:aws:iam::*:role/AWS-SSM-
DiagnosisAdminRole-operational-account-id-home-region",
                        "arn:aws:iam::*:role/AWS-SSM-
RemediationExecutionRole-operational-account-id-home-region",
                        "arn:aws:iam::*:role/AWS-SSM-
RemediationAdminRole-operational-account-id-home-region"
                    ]
                }
            }
        },
        {
            "Sid": "AllowCrossAccountListUnderAccountOwnPrefix",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": "organization-id"
                },
                "StringLike": {
                    "s3:prefix": "*/${aws:PrincipalAccount}/*"
                }
            }
        },
        {
            "Sid": "AllowCrossAccountGetConfigWithinOrganization",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetEncryptionConfiguration",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
            "Condition": {
                "StringEquals": {
```

```
                    "aws:PrincipalOrgID": "organization-id"
                }
            }
        }
    ]
}
```

Amazon-S3-Bucket-Richtlinie für ein einzelnes Konto

Der Diagnose-Bucket wird mit der folgenden Standard-Bucket-Richtlinie erstellt, wenn ein einzelnes Konto in Systems Manager integriert wird.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenyHTTPRequests",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "false"
        }
      }
    },
    {
      "Sid": "DenyNonSigV4Requests",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
      "Condition": {
```

```
        "StringNotEquals": {
          "s3:SignatureVersion": "AWS4-HMAC-SHA256"
        }
      }
    }
  ]
}
```

AWS Systems Manager Tools verwenden

Systems Manager gruppiert Tools in vier Kategorien. In der folgenden Dokumentation werden die verschiedenen Tools von AWS Systems Manager sowie deren Einrichtung und Verwendung beschrieben. Wählen Sie die Registerkarten unter jeder Kategorie aus, um mehr über jedes Tool zu erfahren.

Knoten-Tools

Ein verwalteter Knoten ist jede Maschine, die für die Verwendung mit Systems Manager in Hybrid- und Multi-Cloud-Umgebungen konfiguriert ist.

Compliance

Mit Compliance können Sie Ihre Flotte verwalteter Knoten auf Patch-Compliance und Konfigurations-Inkonsistenzen hin scannen. Sie können Daten aus mehreren Bereichen sammeln AWS-Konten und AWS-Regionen aggregieren und dann nach bestimmten Ressourcen suchen, die nicht den Vorschriften entsprechen. Standardmäßig werden von Compliance die Compliance-Daten zu Patch Manager-Patches und State Manager-Zuordnungen angezeigt. Sie können auch den Service anpassen und Ihre eigenen Compliance-Typen auf Grundlage Ihrer IT- oder Business-Anforderungen erstellen.

Distributor

Distributor ermöglicht das Erstellen von Paketen und deren Bereitstellung auf verwalteten Knoten. Mit Distributor können Sie Ihre eigene Software paketieren oder nach AWS bereitgestellten Agent-Softwarepaketen suchen, z. B. für die Installation auf verwalteten Systems AmazonCloudWatchAgentManager Manager-Knoten. Nachdem Sie ein Paket zum ersten Mal installiert haben, können Sie Distributor verwenden, um eine neue Paketversion zu deinstallieren und neu zu installieren oder um eine direkte Aktualisierung durchzuführen, bei der neue oder geänderte Dateien hinzufügt werden. Distributor veröffentlicht Ressourcen (z. B. Softwarepakete) auf von Systems Manager verwalteten Knoten.

Fleet Manager

Fleet Manager ist eine einheitliche Benutzeroberfläche (UI), mit der Sie Ihre Knoten remote verwalten können. Mit Fleet Manager können Sie sich den Zustand und den Leistungsstatus Ihrer gesamten Flotte von einer Konsole aus ansehen. Sie können auch Daten aus einzelnen Geräten

und Instances sammeln, um allgemeine Problembehandlungs- und Verwaltungsaufgaben über die Konsole auszuführen. Dazu gehören das Anzeigen von Verzeichnis- und Dateiinhalten, die Windows-Registry-Verwaltung, die Betriebssystembenutzerverwaltung und vieles mehr.

Hybrid Activations

Um in Ihrer Hybrid- und Multicloud-Umgebung, die keine EC2 Computer sind, als verwaltete Knoten einzurichten, erstellen Sie eine Hybrid-Aktivierung. Nach Abschluss der Aktivierung erhalten Sie einen Aktivierungscode und eine ID. Diese Kombination aus Code und ID funktioniert wie eine Amazon Elastic Compute Cloud (Amazon EC2) -Zugriffs-ID und ein geheimer Schlüssel, um von Ihren verwalteten Instances aus einen sicheren Zugriff auf den Systems Manager Manager-Service zu ermöglichen.

Sie können auch eine Aktivierung für Edge-Geräte erstellen, wenn Sie diese mithilfe von Systems Manager verwalten möchten.

Inventory

Inventory (Bestand) automatisiert die Erfassung des Software-Bestands auf Ihren verwalteten Instances. Sie können mit Inventory Metadaten über Anwendungen, Dateien, Komponenten, Patches und vieles mehr erfassen.

Patchmanager

Verwenden Sie Patch Manager zum Automatisieren des Patch-Vorgangs für Ihre verwalteten Knoten, sowohl mit Sicherheits-Updates als auch anderen Arten von Updates. Sie können Patch Manager verwenden, um Patches sowohl für Betriebssysteme als auch für Anwendungen durchzuführen. (Unter Windows Server ist der Anwendungssupport auf Updates für Microsoft-Anwendungen beschränkt.)

Mit diesem Tool können Sie verwaltete Knoten auf fehlende Patches scannen und diese mit Tags einzeln oder auf große Gruppen von verwalteten Knoten anwenden. Patch Manager nutzt Patch-Baselines, die Regeln für die automatische Genehmigung von Patches innerhalb einer festgelegten Anzahl von Tagen nach ihrer Veröffentlichung enthalten können, sowie Listen von genehmigten und abgelehnten Patches. Sie können sicherheitsrelevante Patches regelmäßig installieren, indem Sie das Einspielen von Patches als Systems-Manager-Wartungsfenster-Aufgabe planen, oder Sie können Ihre verwalteten Knoten jederzeit bei Bedarf patchen.

Für Linux-Betriebssysteme können Sie als Teil Ihrer Patch-Baseline die Repositorys definieren, die für Patching-Operationen verwendet werden sollen. Auf diese Weise können Sie sicherstellen, dass Updates nur aus vertrauenswürdigen Repositorys installiert werden, unabhängig davon,

welche Repositorys auf dem verwalteten Knoten konfiguriert sind. Für Linux haben Sie auch die Möglichkeit, ein beliebiges Paket auf dem verwalteten Knoten zu aktualisieren, nicht nur diejenigen, die als Sicherheits-Updates für Betriebssysteme eingestuft sind. Sie können Patchberichte auch generieren, die an einen S3-Bucket Ihrer Wahl gesendet werden. Für einen einzelnen verwalteten Knoten enthalten Berichte Details aller Patches für die Maschine. Für einen Bericht über alle verwaltete Knoten wird nur eine Zusammenfassung der fehlenden Patches bereitgestellt.

Run Command

Mit Run Command können Sie die Konfiguration Ihrer verwalteten Instances in großem Umfang remote und sicher verwalten. Verwenden Sie Run Command zum Ausführen von On-Demand-Änderungen, wie das Aktualisieren von Anwendungen oder das Ausführen von Linux-Shell-Skripts und Windows PowerShell-Befehlen auf einem Zielsatz mit Dutzenden oder Hunderten von verwalteten Knoten.

Session Manager

Verwenden Sie diese Option Session Manager, um Ihre Edge-Geräte und Amazon Elastic Compute Cloud (Amazon EC2) -Instances über eine interaktive browserbasierte Shell mit einem Klick oder über die zu verwalten. AWS CLISession Managerbietet sicheres und überprüfbares Edge-Geräte- und Instanzmanagement, ohne dass eingehende Ports geöffnet, Bastion-Hosts verwaltet oder SSH-Schlüssel verwaltet werden müssen. Session Managerermöglicht es Ihnen außerdem, Unternehmensrichtlinien einzuhalten, die einen kontrollierten Zugriff auf Edge-Geräte und -Instanzen, strenge Sicherheitspraktiken und vollständig überprüfbare Protokolle mit Edge-Geräte- und Instanzzugriffsdetails vorschreiben, und bietet Endbenutzern gleichzeitig einen einfachen plattformübergreifenden Zugriff auf Ihre Edge-Geräte und -Instanzen mit nur einem Klick. EC2 Um Session Manager zu verwenden, müssen Sie das Advanced-Instances-Kontingent aktivieren. Weitere Informationen finden Sie unter Aktivieren des Kontingents für erweiterte Instances.

State Manager

State Manager ermöglicht die Automatisierung des Prozesses, mit dem Ihre verwalteten Knoten in einem definierten Zustand gehalten werden. Mit State Manager können Sie bei Ihren verwalteten Knoten sicherstellen, dass ihnen per Bootstrapping bestimmte Software beim Startup angefügt wird, dass sie mit einer Windows-Domain verbunden werden (nur Windows Server-Knoten) oder dass sie mit bestimmten Software-Updates gepatcht werden.

Tools für das Änderungsmanagement

Automation

Mit Automation werden häufig durchzuführende Wartungs- und Bereitstellungsaufgaben automatisiert. Sie können Automation verwenden, um Amazon Machine Images (AMIs) zu erstellen und zu aktualisieren, Treiber- und Agent-Updates anzuwenden, Passwörter auf Windows Server-Instances zurückzusetzen, SSH-Schlüssel auf Linux-Instances zurückzusetzen oder OS-Patches oder Anwendungs-Updates anzuwenden.

Kalender ändern

Change Calendar hilft Ihnen, Datums- und Uhrzeitbereiche einzurichten, in denen von Ihnen angegebene Aktionen (z. B. in Systems-Manager-Automatisierungs-Runbooks) in Ihrem AWS-Konto ausgeführt bzw. nicht ausgeführt werden können. In Change Calendar werden diese Bereiche Ereignisse genannt. Wenn Sie einen Change Calendar-Eintrag erstellen, erstellen Sie ein Systems Manager-Dokument vom Typ `ChangeCalendar`. In Change Calendar wird das Dokument als iCalendar 2.0-Daten im Klartextformat gespeichert. Ereignisse, die Sie dem Change Calendar-Eintrag hinzufügen, werden Teil des Dokuments. Sie können Ereignisse manuell in der Change Calendar-Schnittstelle hinzufügen oder Ereignisse aus einem unterstützten Kalender eines Drittanbieters mithilfe einer `.ics`-Datei importieren.

Change Manager

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Change Manager ist ein Change-Management-Framework für Unternehmen zum Anfordern, Genehmigen, Implementieren und Melden von Betriebsänderungen an Ihrer Anwendungskonfiguration und Infrastruktur. Wenn Sie ein einziges delegiertes Administratorkonto verwenden AWS Organizations, können Sie Änderungen an mehreren AWS-Konten in mehreren AWS-Regionen verwalten. Alternativ können Sie mit einem lokalen Konto Änderungen für

einen einzigen AWS-Konto verwalten. Verwenden Sie Change Manager zum Verwalten von Änderungen an sowohl AWS -Ressourcen als auch On-Premises-Ressourcen.

Documents

Ein Systems Manager-Dokument (SSM-Dokument) definiert die Aktionen, die Systems Manager ausführt. Zu den SSM-Dokumenttypen gehören Command-Dokumente, die von State Manager und Run Command verwendet werden, und Automation-Runbooks, die von Systems Manager Automation verwendet werden. Systems Manager umfasst Dutzende vorkonfigurierter Dokumente, die Sie verwenden können, indem Sie zur Laufzeit Parameter angeben. Die Dokumente können im JSON- oder YAML-Format vorliegen und enthalten die von Ihnen angegeben Schritte und Parameter.

Wartungsfenster

Mit Maintenance Windows können Sie wiederkehrende Zeitpläne für verwaltete Instances einrichten, um Verwaltungsaufgaben wie etwa die Installation von Patches und Updates ohne Unterbrechung der geschäftskritischen Vorgänge auszuführen.

Schnelleinrichtung

Wird verwendet Quick Setup, um häufig verwendete Funktionen AWS-Services und Funktionen mit empfohlenen Best Practices zu konfigurieren. Sie können es einzeln AWS-Konto oder Quick Setup in mehreren verwenden AWS-Konten und AWS-Regionen durch Integration mit AWS Organizations. Quick Setupvereinfacht die Einrichtung von Diensten, einschließlich Systems Manager, durch die Automatisierung häufiger oder empfohlener Aufgaben. Zu diesen Aufgaben gehören beispielsweise die Erstellung der erforderlichen Instanzprofilrollen AWS Identity and Access Management (IAM) und die Einrichtung betrieblicher Best Practices wie regelmäßige Patchscans und Inventarerfassung.

Anwendungstools

AppConfig

Mit AppConfig können Sie Anwendungskonfigurationen und Feature-Flags erstellen, verwalten und bereitstellen. AppConfig unterstützt kontrollierte Bereitstellungen für Anwendungen jeder Größe. Sie können es AppConfig mit Anwendungen verwenden, die auf EC2 Amazon-Instances, AWS Lambda Containern, mobilen Anwendungen oder Edge-Geräten gehostet werden. Um Fehler bei der Bereitstellung von Anwendungskonfigurationen zu vermeiden, enthält AppConfig Validierungen. Eine Validierung stellt durch eine syntaktische oder semantische Prüfung sicher,

dass die Konfiguration, die Sie bereitstellen möchten, wie beabsichtigt funktioniert. Während einer Konfigurationsbereitstellung überwacht AppConfig die Anwendung, um sicherzustellen, dass die Bereitstellung erfolgreich ist. Falls im System ein Fehler auftritt oder die Bereitstellung einen Alarm auslöst, setzt AppConfig die Änderung zurück, um die Auswirkungen auf die Benutzer Ihrer Anwendung zu minimieren.

Application Manager

Application Managerhilft DevOps Technikern, Probleme mit ihren AWS Ressourcen im Kontext ihrer Anwendungen und Cluster zu untersuchen und zu beheben. Bei Application Manager einer Anwendung handelt es sich um eine logische Gruppe von AWS Ressourcen, die Sie als Einheit betreiben möchten. Diese logische Gruppe kann verschiedene Versionen einer Anwendung, Eigentumsgrenzen für Betreiber oder Entwicklerumgebungen darstellen, um nur einige zu nennen. Application ManagerDie Unterstützung für Container-Cluster umfasst sowohl Amazon Elastic Kubernetes Service (Amazon EKS) als auch Amazon Elastic Container Service (Amazon ECS) - Cluster. Application Managerfasst Betriebsinformationen aus mehreren Tools AWS-Services und Systems Manager Manager-Tools in einem einzigen AWS-Managementkonsole zusammen.

Parameter Store

Parameter Store ermöglicht eine sichere, hierarchische Speicherung für die Konfigurationsdatenverwaltung und das Verschlüsselungsmanagement. Sie können Daten wie Passwörter, Datenbankzeichenfolgen, Amazon Elastic Compute Cloud (Amazon EC2) -Instance IDs und Amazon Machine Image (AMI) IDs und Lizenzcodes als Parameterwerte speichern. Sie können Werte als Klartext oder als verschlüsselte Daten speichern. Anschließend können Sie die Werte anhand des eindeutigen Namens, den Sie beim Erstellen des Parameters angegeben haben, referenzieren.

Tools für den Betrieb

CloudWatch Dashboards

CloudWatch Amazon-Dashboards sind anpassbare Seiten in der CloudWatch Konsole, mit denen Sie Ihre Ressourcen in einer einzigen Ansicht überwachen können, auch die Ressourcen, die über verschiedene Regionen verteilt sind. Sie können CloudWatch-Dashboards verwenden, um benutzerdefinierte Ansichten der Metriken und Alarme für Ihre AWS -Ressourcen zu erstellen.

Explorer

Explorerist ein anpassbares Operations-Dashboard, das Informationen über Ihre AWS Ressourcen enthält. Explorerzeigt eine aggregierte Ansicht der Betriebsdaten (OpsData) für Sie AWS-Konten und Across AWS-Regionen an. OpsData Enthält Metadaten zu Ihren EC2 Amazon-Instances, Details zur Patch-Konformität und betriebliche Arbeitsaufgaben (OpsItems). Explorer Explorerbietet einen Überblick darüber, wie sie auf Ihre Geschäftsbereiche oder Anwendungen verteilt OpsItems sind, wie sie sich im Laufe der Zeit entwickeln und wie sie sich je nach Kategorie unterscheiden. Sie können Informationen in Explorer gruppieren und filtern, um sich auf die Elemente zu konzentrieren, die für Sie relevant sind und eine Aktion erfordern. Wenn Sie Probleme mit hoher Priorität erkennen, können Sie OpsCenter verwenden, ein Tool von Systems Manager, um Automation-Runbooks auszuführen und diese Probleme zu beheben.

Incident Manager

Incident Manager ist eine Incident-Management-Konsole, die Benutzern hilft, Vorfälle, die sich auf ihre AWS gehosteten Anwendungen auswirken, zu minimieren und diese zu beheben.

Incident Manager beschleunigt die Behebung von Vorfällen, indem es die zuständigen Mitarbeiter über die Auswirkungen informiert, relevante Daten zur Fehlerbehebung hervorhebt und Tools für die Zusammenarbeit bereitstellt, um die Dienste wieder zum Laufen zu bringen. Incident Manager automatisiert auch Antwortpläne und ermöglicht die Eskalation des Responder-Teams.

OpsCenter

OpsCenterbietet einen zentralen Ort, an dem Betriebsingenieure und IT Fachkräfte betriebliche Arbeitsaufgaben (OpsItems) im Zusammenhang mit AWS Ressourcen einsehen, untersuchen und lösen können. OpsCenterwurde entwickelt, um die durchschnittliche Zeit bis zur Lösung von Problemen zu reduzieren, die sich auf AWS Ressourcen auswirken. Dieses Systems Manager-Tool aggregiert und standardisiert OpsItems über Services hinweg und bietet gleichzeitig kontextbezogene Untersuchungsdaten über jedes OpsItem, ähnliche OpsItems und verwandte Ressourcen. OpsCenter bietet außerdem Systems Manager-Automatisierungsdokumente (Runbooks), die Sie verwenden können, um Probleme schnell zu lösen. Sie können durchsuchbare, benutzerdefinierte Daten für jedes OpsItem angeben. Sie können automatisch generierte Zusammenfassungsberichte über OpsItems nach Status und Quelle anzeigen.

Knotenaufgabe mit Systems Manager ausführen

Verwenden Sie dieses Tutorial, für Ihre Ersten Schritte mit AWS Systems Manager. Sie erfahren, wie Sie eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance starten, die von Systems Manager verwaltet wird, und wie Sie eine Verbindung mit der verwalteten Instance herstellen.

Da Systems Manager eine Sammlung mehrerer Tools ist, kann der gesamte Service nicht durch eine einzelne Anleitung oder ein einzelnes Tutorial vorgestellt werden. Dieses Tutorial enthält eine Einführung in einige der Tools.

Voraussetzungen

Bevor Sie beginnen, sollten Sie sicherstellen, dass Sie die in Verwalten von EC2-Instances mit Systems Manager beschriebenen Schritte ausgeführt haben.

Starten einer Instance mit einer AMI mit vorinstalliertem SSM Agent

Sie können eine Amazon-EC2-Instance über die AWS-Managementkonsole starten, wie in der folgenden Vorgehensweise beschrieben. Dieses Tutorial soll Sie dabei unterstützen, Ihre erste verwaltete Instance schnell zu starten. Es deckt daher nicht alle möglichen Optionen ab.

So starten Sie eine Instance

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie über das Dashboard der EC2-Konsole in der Ansicht Launch instance (Starten der Instance) die Option Launch instance (Instance starten) aus und klicken Sie unter den angezeigten Optionen auf Launch instance (Instance starten).
3. Geben Sie für Name und Tags unter Name einen beschreibenden Namen für Ihre Instance ein.
4. Führen Sie unter Anwendungs- und Betriebssystem-Images (Amazon Machine Image) die folgenden Schritte aus:
   a. Wählen Sie die Registerkarte Schnellstart und dann Amazon Linux. Dies ist das Betriebssystem für Ihre Instance.
   b. Wählen Sie für Amazon Machine Image (AMI) eine HVM-Version von Amazon Linux 2 aus.
5. Für Instance-Typ können Sie aus der Liste Instance-Typ die Hardware-Konfiguration für Ihre Instance auswählen. Wählen Sie den Instance-Typ `t2.micro` aus (Standardeinstellung). Der `t2.micro`-Instance-Typ ist für das kostenlose Kontingent von AWS berechtigt. In AWS-

Regionen, in denen `t2.micro` nicht verfügbar ist, können Sie eine `t3.micro`-Instance im Rahmen des kostenlosen Kontingents verwenden. Weitere Informationen finden Sie unter Kostenloses Kontingent für AWS.

6. Wählen Sie unter Schlüsselpaar (Anmeldung) für Schlüsselpaar-Name ein Schlüsselpaar aus.
7. Wählen Sie für Netzwerkeinstellungen die Option Bearbeiten aus. Beachten Sie bei Name der Sicherheitsgruppe, dass der Assistent eine Sicherheitsgruppe für Sie erstellt und ausgewählt hat. Sie können diese Sicherheitsgruppe verwenden oder alternativ eine zuvor erstellte Sicherheitsgruppe mit den folgenden Schritten auswählen:
   a. Wählen Sie Select an existing security group (Eine bestehende Sicherheitsgruppe auswählen) aus.
   b. Wählen Sie unter Common security groups (Gemeinsame Sicherheitsgruppen) Ihre Sicherheitsgruppe in der Liste mit den vorhandenen Sicherheitsgruppen aus.
8. Wenn Sie die Standardkonfiguration für die Host-Verwaltung nicht verwenden, erweitern Sie den Abschnitt Erweiterte Details und wählen Sie für das IAM-Instance-Profil das Instance-Profil aus, das Sie bei der Einrichtung in Konfigurieren von erforderlichen Instance-Berechtigungen für Systems Manager erstellt haben.
9. Behalten Sie die Standardauswahl für die anderen Konfigurationseinstellungen für Ihre Instance bei.
10. Überprüfen Sie eine Zusammenfassung Ihrer Instance-Konfiguration im Bereich Zusammenfassung. Sobald Sie bereit sind, wählen Sie Instance starten aus.
11. Eine Bestätigungsseite informiert Sie darüber, dass Ihre Instance gestartet wird. Wählen Sie View all Instances (Alle Instances anzeigen) aus, um die Bestätigungsseite zu schließen und zur Konsole zurückzukehren.
12. Auf dem Bildschirm Instances können Sie den Status des Starts anzeigen. Es dauert einige Zeit, bis die Instance startet.
13. Es kann ein paar Minuten dauern, bis die Instance als verwaltet angezeigt wird und Sie eine Verbindung damit herstellen können. Um zu überprüfen, ob Ihre Instance die Statusprüfungen bestanden hat, zeigen Sie diese Informationen in der Spalte Statusprüfung an.

Eine Verbindung zu Ihrer verwalteten Instance mithilfe von Systems Manager herstellen

So stellen Sie eine Verbindung zu Ihrer verwalteten Instance her

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben der Instance aus, mit der Sie sich verbinden möchten.
4. Wählen Sie im Menü Knotenaktionen die Option Terminalsitzung starten aus.
5. Wählen Sie Verbinden aus.

Bereinigen Ihrer Instance

Wenn Sie die Arbeit mit der verwalteten Instance, die Sie für dieses Tutorial erstellt haben, abgeschlossen haben, beenden Sie sie. Durch das Beenden einer Instance wird diese effektiv gelöscht.

So beenden Sie Ihre Instance

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich Instances aus. Wählen Sie in der Liste mit den Instances die gewünschte Instance aus.
3. Wählen Sie Instance state (Instance-Status), Terminate instance (Instance beenden).
4. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Beenden aus.

   Amazon EC2 fährt Ihre Instance herunter und beendet sie. Nachdem Ihre Instance beendet wurde, bleibt sie noch kurz auf der Konsole sichtbar, danach wird der Eintrag automatisch gelöscht. Sie können die beendete Instance nicht selbst aus der Konsolenanzeige entfernen.

AWS Systems Manager Knoten-Tools

AWS Systems Manager bietet die folgenden Tools für den Zugriff, die Verwaltung und Konfiguration Ihrer verwalteten Knoten. Ein verwalteter Knoten ist eine Maschine, die für die Verwendung mit Systems Manager in einer Hybrid- und Multi-Cloud-Umgebung konfiguriert ist.

Themen

- AWS Systems Manager-Compliance
- AWS Systems Manager Distributor
- AWS Systems Manager Fleet Manager
- AWS Systems Manager Hybride Aktivierungen
- AWS Systems Manager-Bestand
- AWS Systems Manager Patch Manager
- AWS Systems Manager Run Command
- AWS Systems Manager Session Manager
- AWS Systems Manager State Manager

AWS Systems Manager-Compliance

Sie können Compliance, ein Tool in, verwenden AWS Systems Manager, um Ihre Flotte verwalteter Knoten auf Patch-Compliance und Konfigurationsinkonsistenzen zu überprüfen. Sie können Daten aus mehreren Regionen sammeln und aggregieren AWS-Konten und dann nach bestimmten Ressourcen suchen, die nicht den Vorschriften entsprechen. Standardmäßig zeigt Compliance aktuelle Compliance-Daten zum Patchen in Patch Manager und Assoziationen in State Manager an. (Patch Manager und State Manager sind ebenfalls beide Tools in AWS Systems Manager.) Um mit Compliance zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im linken Navigationsbereich Compliance.

Daten zur Patch-Konformität von Patch Manager können an gesendet werden AWS Security Hub CSPM. Security Hub CSPM bietet Ihnen einen umfassenden Überblick über Ihre Sicherheitswarnungen mit hoher Priorität und den Compliance-Status. Er überwacht auch den Patching-Status Ihrer Flotte. Weitere Informationen finden Sie unter Integration von Patch Manager in AWS Security Hub CSPM.

Compliance bietet die folgenden zusätzlichen Vorteile und Funktionen:

- Zeigen Sie den Compliance-Verlauf und die Änderungsnachverfolgung für Patch Manager-Patching-Daten und State Manager-Zuordnungen mithilfe von AWS Config an.
- Passen Sie Compliance an, um Ihre eigenen Compliance-Typen auf Grundlage Ihrer IT- oder Business-Anforderungen zu erstellen.

- Beheben Sie ProblemeRun Command, indem Sie ein anderes Tool in AWS Systems Manager oder Amazon EventBridge verwenden. State Manager
- Portieren Sie Daten nach Amazon Athena und Amazon Quick Suite, um flottenweite Berichte zu erstellen.

EventBridge Unterstützung

Dieses Systems Manager Manager-Tool wird in den EventBridge Amazon-Regeln als Ereignistyp unterstützt. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge und Referenz: EventBridge Amazon-Ereignistypen und -muster für Systems Manager.

Chef InSpec-Integration

Systems Manager lässt sich in integrieren Chef InSpec. InSpec ist ein Open-Source-Runtime-Framework, mit dem Sie menschenlesbare Profile auf GitHub Amazon Simple Storage Service (Amazon S3) erstellen können. Anschließend können Sie Systems Manager verwenden, um Compliance-Scans auszuführen und konforme bzw. nicht konforme Knoten anzuzeigen. Weitere Informationen finden Sie unter Verwenden von Chef InSpec-Profilen mit Systems Manager Compliance.

Preisgestaltung

Compliance wird ohne Zusatzkosten angeboten. Sie zahlen nur für die AWS Ressourcen, die Sie nutzen.

Inhalt

- Erste Schritte mit Compliance
- Konfigurieren von Berechtigungen für die Compliance
- Erstellen einer Ressource Data Sync für Compliance
- Erfahren Sie mehr über Compliance
- Löschen einer Ressource Data Sync für Compliance
- Beheben von Compliance-Problemen mithilfe von EventBridge
- Weisen Sie benutzerdefinierte Compliance-Metadaten zu mithilfe der AWS CLI

Erste Schritte mit Compliance

Führen Sie die folgenden Schritte durch, um Compliance, ein Tool in AWS Systems Manager, zu verwenden.

Aufgabe	Weitere Informationen
Compliance funktioniert mit Patch-Daten in Patch Manager und Zuordnungen in State Manager. (Patch Manager und State Manager sind ebenfalls Tools in AWS Systems Manager.) Compliance funktioniert auch mit benutzerdefinierten Kompatibilitätstypen auf verwalteten Knoten, die mit Systems Manager verwaltet werden. Überprüfen Sie, ob Sie die Einrichtungsanforderungen für Ihre Instances der Amazon Elastic Compute Cloud (Amazon EC2) und Nicht-EC2-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung erfüllt haben.	Einrichten der einheitlichen Systems-Manager-Konsole für eine Organisation
Aktualisieren Sie die AWS Identity and Access Management (IAM)-Rolle, die Ihre verwaltet en Knoten verwenden, um die Compliance-Berechtigungen einzuschränken.	Konfigurieren von Berechtigungen für die Compliance
Wenn Sie die Patch-Compliance überwachen möchten, überprüfen Sie die Konfiguration des Patch Manager. Sie müssen Patch-Vorgänge mit dem Patch Manager durchführen, bevor die Patch-Compliance-Daten von Compliance angezeigt werden können.	AWS Systems Manager Patch Manager
Wenn Sie die Zuordnungs-Compliance überwachen möchten, überprüfen Sie, dass Sie State Manager-Zuordnungen erstellt haben. Sie müssen Zuordnungen erstellen, bevor	AWS Systems Manager State Manager

Aufgabe	Weitere Informationen
die Daten zur Zuordnungs-Compliance von Compliance angezeigt werden können.	
(Optional) Konfigurieren Sie das System, um den Compliance-Verlauf und die Änderungs nachverfolgung anzuzeigen.	Anzeigen von Compliance-Konfigurationsverlauf und Änderungsnachverfolgung
(Optional) Erstellen Sie benutzerdefinierte Compliance-Typen.	Weisen Sie benutzerdefinierte Compliance-Metadaten zu mithilfe der AWS CLI
(Optional) Erstellen Sie eine Resource Data Sync zur Aggregation aller Compliance-Daten in einem Amazon Simple Storage Service (Amazon S3)-Bucket.	Erstellen einer Ressource Data Sync für Compliance

Konfigurieren von Berechtigungen für die Compliance

Aus Sicherheitsgründen empfehlen wir, dass Sie die von Ihren verwalteten Knoten verwendete AWS Identity and Access Management (IAM-) Rolle mit den folgenden Berechtigungen aktualisieren, um die Fähigkeit des Knotens einzuschränken, die PutComplianceItemsAPI-Aktion zu verwenden. Diese API-Aktion registriert einen Compliance-Typ und andere Compliance-Details auf einer bestimmten Ressource, z. B. einer EC2 Amazon-Instance oder einem verwalteten Knoten.

Wenn es sich bei Ihrem Knoten um eine EC2 Amazon-Instance handelt, müssen Sie das von der Instance verwendete IAM-Instance-Profil mit den folgenden Berechtigungen aktualisieren. Weitere Informationen zu Instanzprofilen für von Systems Manager verwaltete EC2 Instanzen finden Sie unterKonfigurieren von erforderlichen Instance-Berechtigungen für Systems Manager. Für andere Typen von verwalteten Knoten sollten Sie die vom Knoten verwendete IAM-Rolle mit den folgenden Berechtigungen aktualisieren. Weitere Informationen finden Sie unter Aktualisieren von Berechtigungen für eine Rolle im IAM-Benutzerhandbuch.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutComplianceItems"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:SourceInstanceARN": "${ec2:SourceInstanceARN}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutComplianceItems"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ssm:SourceInstanceARN": "${ssm:SourceInstanceARN}"
                }
            }
        }
    ]
}
```

Erstellen einer Ressource Data Sync für Compliance

Sie können die Funktion zur Synchronisierung von Ressourcendaten verwenden AWS Systems Manager , um Compliance-Daten von all Ihren verwalteten Knoten an einen Amazon Simple Storage Service (Amazon S3) -Ziel-Bucket zu senden. Wenn Sie die Synchronisierung erstellen, können Sie verwaltete Knoten aus mehreren AWS-Konten AWS-Regionen, und Ihrer Hybrid- und Multi-Cloud-Umgebung angeben. Resource Data Sync aktualisiert die Daten dann automatisch, sobald neue Compliance-Daten erfasst werden. Da alle Compliance-Daten in einem Ziel-S3-Bucket gespeichert sind, können Sie Dienste wie Amazon Athena und Amazon Quick Suite verwenden, um die aggregierten Daten abzufragen und zu analysieren. Resource Data Sync muss einmalig für Compliance konfiguriert werden.

Führen Sie die folgenden Schritte aus, um mit der AWS-Managementkonsole eine Ressource Data Sync für Compliance zu erstellen.

So erstellen und konfigurieren Sie einen S3-Bucket für die Synchronisierung von Ressourcendaten (Konsole)

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Erstellen Sie ein Bucket zum Speichern der zusammengefassten Compliance-Daten. Weitere Informationen finden Sie unter Erstellen eines Buckets im Benutzerhandbuch zu Amazon Simple Storage Service. Notieren Sie sich den Namen des Buckets und den AWS-Region Ort, an dem Sie ihn erstellt haben.
3. Öffnen Sie den Bucket, wählen Sie die Registerkarte Permissions (Berechtigungen) und anschließend die Option Bucket Policy (Bucket-Richtlinie).
4. Kopieren Sie die folgende Bucket-Richtlinie in den Richtlinien-Editor. Ersetzen Sie amzn-s3-demo-bucket und *Account-ID* durch den Namen des S3-Buckets, den Sie erstellt haben, und eine gültige ID. AWS-Konto Optional können Sie es *Bucket-Prefix* durch den Namen eines Amazon S3 S3-Präfix (Unterverzeichnis) ersetzen. Wenn Sie kein Präfix erstellt haben, entfernen Sie*Bucket-Prefix*/aus dem ARN in der Richtlinie.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "SSMBucketPermissionsCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
        },
        {
            "Sid": " SSMBucketDelivery",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket/Bucket-Prefix/*/
accountid=111122223333/*"],
```

```
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
```

So erstellen Sie eine Ressourcendaten-Synchronisierung

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie Account management (Kontoverwaltung), Resource Data Syncs und dann Create resource data sync (Resource Data Sync erstellen)
4. Geben Sie im Feld Sync name einen Namen für die Synchronisierungskonfiguration ein.
5. Geben Sie im Feld Bucket name (Bucket-Name) den Namen des zu Beginn dieses Vorgangs erstellten Amazon S3-Buckets an.
6. (Optional) Geben Sie im Feld Bucket-Präfix den Namen eines S3-Bucket-Präfixes (Unterverzeichnis) an.
7. Wählen Sie im Feld Bucket-Region die Option Diese Region aus, wenn sich der erstellte S3-Bucket in der aktuellen AWS-Region befindet. Wenn sich der Bucket in einer anderen Region befindet AWS-Region, wählen Sie „Andere Region“ und geben Sie den Namen der Region ein.

   > **Note**
   >
   > Wenn sich die Synchronisierung und der Ziel-S3-Bucket in verschiedenen Regionen befinden, müssen Sie möglicherweise Gebühren für die Datenübertragung zahlen. Weitere Informationen finden Sie unter Amazon S3 – Preise.

8. Wählen Sie Erstellen aus.

Erfahren Sie mehr über Compliance

Compliance, ein Tool in AWS Systems Manager, sammelt und meldet Daten über den Status von Patches in Patch Manager Patching und Assoziationen in. State Manager (Patch Managerund beide Tools State Manager sind auch dabei.) AWS Systems Manager Compliance berichtet auch zu benutzerdefinierten Compliance-Typen, die Sie für Ihre verwalteten Knoten angegeben haben. Dieser Abschnitt enthält Details über jeden dieser Compliance-Typen sowie Informationen zum Anzeigen von Systems Manager-Compliance-Daten. Dieser Abschnitt enthält auch Informationen zum Anzeigen des Compliance-Verlaufs und der Änderungsnachverfolgung.

> **Note**
>
> Systems Manager lässt sich in integrieren Chef InSpec. InSpec ist ein Open-Source-Runtime-Framework, mit dem Sie menschenlesbare Profile auf GitHub Amazon Simple Storage Service (Amazon S3) erstellen können. Anschließend können Sie Systems Manager verwenden, um Compliance-Scans auszuführen und konforme und nicht konforme Instances anzuzeigen. Weitere Informationen finden Sie unter Verwenden von Chef InSpec-Profilen mit Systems Manager Compliance.

Info zu Patch Compliance

Nachdem Sie mit Patch Manager in Ihren Instances installiert haben, stehen Ihnen die Compliance-Statusinformationen sofort in der Konsole oder in der Antwort auf AWS Command Line Interface (AWS CLI)-Befehle oder entsprechenden Systems Manager-API-Vorgängen zur Verfügung.

Weitere Informationen über Patch-Compliance-Statuswerte finden Sie unter Statuswerte der Patch-Compliance.

Informationen zu State Manager-Zuordnungs-Compliance

Nachdem Sie eine oder mehrere State Manager Zuordnungen erstellt haben, stehen Ihnen Informationen zum Compliance-Status sofort in der Konsole oder als Reaktion auf AWS CLI Befehle oder entsprechende Systems Manager Manager-API-Operationen zur Verfügung. Für Zuordnungen zeigt Compliance den Status `Compliant` oder `Non-compliant` und den der Zuordnung zugewiesenen Schweregrad an, z. B. `Critical` oder `Medium`.

Wenn State Manager eine Zuordnung auf einem verwalteten Knoten ausführt, wird eine Compliance-Aggregation ausgelöst, die den Compliance-Status für alle Zuordnungen auf diesem Knoten aktualisiert. Der `ExecutionTime`-Wert in Compliance-Berichten gibt an, wann der Compliance-

Status von Systems Manager erfasst wurde, und nicht, wann die Zuordnung auf dem verwalteten Knoten ausgeführt wurde. Das bedeutet, dass mehrere Verknüpfungen möglicherweise identische `ExecutionTime`-Werte anzeigen, auch wenn sie zu unterschiedlichen Zeiten ausgeführt wurden. Die tatsächlichen Ausführungszeiten für Verknüpfungen können Sie mithilfe des AWS CLI Befehls describe-association-execution-targetsoder anhand der Ausführungsdetails in der Konsole im Verlauf der Zuordnungsausführung ermitteln.

Informationen zu benutzerdefinierter Compliance

Einem verwalteten Knoten können Compliance-Metadaten zugewiesen werden. Diese Metadaten können anschließend mit anderen Compliance-Daten für Compliance-Berichte zusammengefasst werden. Beispiel: Ihr Unternehmen führt die Versionen 2.0, 3.0 und 4.0 von Software X auf Ihren verwalteten Knoten aus. Das Unternehmen möchte Version 4.0 zum Standard machen. Das bedeutet, dass Instances mit Versionen 2.0 und 3.0 nicht konform sind. Mithilfe des PutComplianceItemsAPI-Vorgangs können Sie explizit angeben, auf welchen verwalteten Knoten ältere Versionen von Software X ausgeführt werden. Sie können Konformitätsmetadaten nur mithilfe von AWS CLI AWS Tools for Windows PowerShell, oder zuweisen SDKs. Mit dem folgenden CLI-Beispielbefehl werden einer verwalteten Instance Compliance-Metadaten zugewiesen und der Compliance-Typ im benötigten Format angegeben `Custom:`. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen.

Linux & macOS

```
aws ssm put-compliance-items \
    --resource-id i-1234567890abcdef0 \
    --resource-type ManagedInstance \
    --compliance-type Custom:SoftwareXCheck \
    --execution-summary ExecutionTime=AnyStringToDenoteTimeOrDate \
    --items
 Id=Version2.0,Title=SoftwareXVersion,Severity=CRITICAL,Status=NON_COMPLIANT
```

Windows

```
aws ssm put-compliance-items ^
    --resource-id i-1234567890abcdef0 ^
    --resource-type ManagedInstance ^
    --compliance-type Custom:SoftwareXCheck ^
    --execution-summary ExecutionTime=AnyStringToDenoteTimeOrDate ^
    --items
 Id=Version2.0,Title=SoftwareXVersion,Severity=CRITICAL,Status=NON_COMPLIANT
```

> **Note**
>
> Der `ResourceType`-Parameter unterstützt nur `ManagedInstance`. Wenn Sie einem AWS IoT Greengrass -Core-Gerät benutzerdefinierte Compliance hinzufügen, müssen Sie einen `ResourceType` von `ManagedInstance` angeben.

Compliance-Manager können daraufhin Zusammenfassungen anzeigen oder Berichte über nicht konforme verwaltete Knoten erstellen. Sie können einem verwalteten Knoten maximal 10 verschiedene benutzerdefinierte Compliance-Typen zuweisen.

Ein Beispiel für die Erstellung eines benutzerdefinierten Compliance-Typs und zum Anzeigen von Compliance-Daten finden Sie unter Weisen Sie benutzerdefinierte Compliance-Metadaten zu mithilfe der AWS CLI.

Anzeigen aktueller Compliance-Daten

In diesem Abschnitt wird beschrieben, wie Sie Compliance-Daten in der Systems Manager-Konsole und mithilfe der AWS CLI anzeigen. Weitere Informationen zum Anzeigen des Patch- und Zuordnungs-Compliance-Verlaufs und der Änderungsnachverfolgung finden Sie unter Anzeigen von Compliance-Konfigurationsverlauf und Änderungsnachverfolgung.

Themen

- Anzeigen aktueller Compliance-Daten (Konsole)
- Anzeigen aktueller Compliance-Daten (AWS CLI)

Anzeigen aktueller Compliance-Daten (Konsole)

Verwenden Sie die folgenden Verfahren, um Compliance-Daten in der Systems Manager-Konsole anzuzeigen.

So zeigen Sie aktuelle Compliance-Berichte in der Systems Manager-Konsole an

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im linken Navigationsbereich Compliance.
3. Wählen Sie im Abschnitt Compliance dashboard filtering (Compliance-Dashboard-Filtrierung) eine Option zum Filtern von Compliance-Daten aus. Der Abschnitt Compliance resources

summary (Zusammenfassung der Compliance-Ressourcen) zeigt die Anzahl der Compliance-Daten basierend auf dem von Ihnen ausgewählten Filter an.

4. Um weitere detaillierte Informationen zu einer Ressource zu erhalten, scrollen Sie nach unten zum Bereich Details overview for resources (Detailübersicht für Ressourcen) und wählen Sie die ID eines verwalteten Knotens.
5. Wählen Sie auf der Detailseite Instance ID (Instance-ID) oder Name die Registerkarte Configuration compliance (Konfigurations-Compliance), um den detaillierten Bericht zur Konfigurations-Compliance anzuzeigen.

> **Note**
>
> Weitere Informationen zum Beheben von Compliance-Problemen finden Sie unter Beheben von Compliance-Problemen mithilfe von EventBridge.

Anzeigen aktueller Compliance-Daten (AWS CLI)

Mithilfe der folgenden AWS CLI Befehle können Sie Zusammenfassungen der Kompatibilitätsdaten für Patches, Verknüpfungen und benutzerdefinierte Kompatibilitätstypen im in der AWS CLI anzeigen.

list-compliance-summaries

Gibt eine Übersichtszahl der konformen und nicht konformen Zuordnungs-Statusarten entsprechend der angegebenen Filter zurück. (API:) ListComplianceSummaries

list-resource-compliance-summaries

Gibt eine Übersichtszahl auf Ressourcenebene zurück. Die Übersicht umfasst Informationen über konforme und nicht konforme Statusarten und die detaillierte Anzahl des Schweregrads von Compliance-Elementen entsprechend den festgelegten Filterkriterien. (API: ListResourceComplianceSummaries)

Sie können zusätzliche Compliance-Daten für das Einspielen von Patches mit den folgenden AWS CLI -Befehlen anzeigen.

describe-patch-group-state

Gibt allgemeine zusammengefasste Patch-Compliance-Statusarten für eine Patch-Gruppe zurück. (API: DescribePatchGroupState)

describe-instance-patch-states-for-patch-group

Gibt den allgemeinen Patch-Status für die Instances in der angegebenen Patch-Gruppe zurück. (API: DescribeInstancePatchStatesForPatchGroup)

> **Note**
>
> Eine Veranschaulichung der Konfiguration von Patches und der Anzeige von Informationen zur Patch-Konformität mithilfe von finden Sie unterTutorial: Patchen Sie eine Serverumgebung mit dem AWS CLI. AWS CLI

Anzeigen von Compliance-Konfigurationsverlauf und Änderungsnachverfolgung

Standardmäßig werden von Systems-Manager-Compliance die aktuellen Patch-Vorgänge und Zuordnungs-Compliance-Daten für Ihre verwalteten Knoten angezeigt. Sie können den Verlauf der Einhaltung von Patches und Zuordnungen sowie die Änderungsnachverfolgung anzeigen, indem Sie AWS Config AWS Config bietet einen detaillierten Überblick über die Konfiguration der AWS Ressourcen in Ihrem AWS-Konto. Dazu gehört auch, wie die Ressourcen jeweils zueinander in Beziehung stehen und wie sie in der Vergangenheit konfiguriert wurden, damit Sie sehen können, wie sich die Konfigurationen und Beziehungen im Laufe der Zeit verändern. Zum Anzeigen von Patch-Einspielungen, des Zuordnungs-Compliance-Verlaufs und der Änderungsnachverfolgung müssen Sie die folgenden Ressourcen in AWS Config aktivieren:

- `SSM:PatchCompliance`
- `SSM:AssociationCompliance`

Weitere Informationen dazu, wie Sie diese spezifischen Ressourcen in AWS Config auswählen und konfigurieren, finden Sie unter Selecting Which Resources AWS Config Records im AWS Config - Entwicklerleitfaden.

> **Note**
>
> Informationen zur AWS Config Preisgestaltung finden Sie unter Preise.

Löschen einer Ressource Data Sync für Compliance

Wenn Sie AWS Systems Manager Compliance nicht mehr verwenden wollen, um Compliance-Daten anzuzeigen, empfehlen wir außerdem, Resource Data Syncs zu löschen, die für die Erfassung von Compliance-Daten verwendet werden

Löschen eines Compliance Resource Data Sync

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Klicken Sie auf Account management (Kontenverwaltung), Resource data syncs.
4. Wählen Sie eine Synchronisierung aus der Liste aus.

   > **Important**
   >
   > Stellen Sie sicher, dass Sie die für Compliance verwendete Synchronisierung auswählen. Systems Manager unterstützt die Ressourcendaten-Synchronisierung für mehrere Tools. Wenn Sie die falsche Synchronisierung wählen, können Sie die Datenaggregation für Systems Manager Explorer oder Systems Manager Inventory unterbrechen.

5. Wählen Sie Löschen aus.
6. Löschen Sie den Amazon Simple Storage Service (Amazon S3)-Bucket, in dem die Daten gespeichert wurden. Weitere Informationen zum Löschen eines S3-Buckets finden Sie unter Löschen eines Buckets.

Beheben von Compliance-Problemen mithilfe von EventBridge

Mit Run Command, einem Tool in AWS Systems Manager, lassen sich Probleme bei der Patch- und Zuordnungs-Compliance schnell beheben. Sie können Instance- oder AWS IoT Greengrass-Core-Geräte-IDs oder Tags anvisieren und das `AWS-RunPatchBaseline`-Dokument oder das `AWS-RefreshAssociation`-Dokument ausführen. Wenn das Compliance-Problem nicht durch die Aktualisierung der Zuordnung oder eine erneute Ausführung der Patch-Baseline behoben wird, müssen Sie Ihre Zuordnungen, Patch-Baselines oder Instance-Konfigurationen untersuchen, um herauszufinden, warum die Run Command-Operationen das Problem nicht behoben haben.

Weitere Informationen zu Patch-Vorgängen finden Sie unter AWS Systems Manager Patch Manager und SSM-Befehlsdokument zum Patchen: `AWS-RunPatchBaseline`.

Weitere Informationen zu Zuordnungen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager.

Weitere Informationen zum Ausführen eines Befehls finden Sie unter AWS Systems Manager Run Command.

Compliance als Ziel eines EventBridge-Ereignisses angeben

Sie können Amazon EventBridge auch so konfigurieren, dass als Reaktion auf Systems Manager Compliance-Ereignisse eine Aktion ausgeführt wird. Zum Beispiel, wenn kritische Patch-Updates auf mindestens einem verwalteten Knoten nicht installiert werden oder eine Zuordnung zur Installation einer Antiviren-Software nicht ausgeführt wird, können Sie EventBridge so konfigurieren, dass das `AWS-RunPatchBaseline`-Dokument oder das `AWS-RefreshAssocation`-Dokument ausgeführt wird, wenn das Compliance-Ereignis eintritt.

Führen Sie die folgenden Schritte aus, um Compliance als Ziel eines EventBridge-Ereignisses festzulegen.

Konfigurieren von Compliance als Ziel eines EventBridge-Ereignisses (Konsole)

1. Öffnen Sie die Amazon EventBridge-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Regeln aus.
3. Wählen Sie Regel erstellen aus.
4. Geben Sie einen Namen und eine Beschreibung für die Regel ein.

   Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben AWS-Region und auf demselben Event Bus haben.
5. Wählen Sie für Event Bus den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel auf übereinstimmende Ereignisse reagiert, die von Ihrem eigenen AWS-Konto stammen, wählen Sie Standard aus. Wenn ein AWS-Service in Ihrem Konto ein Ereignis ausgibt, wird es stets an den Standard-Event-Bus Ihres Kontos weitergeleitet.
6. Bei Regeltyp wählen Sie Regel mit einem Ereignismuster aus.
7. Wählen Sie Weiter aus.
8. Wählen Sie für Ereignisquelle AWS-Ereignisse oder EventBridge-Partnerereignisse.
9. Wählen Sie im Abschnitt Ereignismuster die Option Ereignismusterformular aus.
10. Als Event source (Ereignisquelle) wählen Sie AWS-Services aus.

11. Wählen Sie für AWS-Service, die Option Systems Manager aus.
12. Wählen Sie für Event Type (Ereignistyp) Configuration Compliance.
13. Für Specific detail type(s) (Spezifische(r) Detail-Typ(en)), wählen Sie Configuration Compliance State Change (Konfiguration-Compliance-Statusänderung).
14. Wählen Sie Weiter aus.
15. Bei Zieltypen wählen Sie AWS-Service aus.
16. Für Select a target (Ziel auswählen), wählen Sie Systems Manager Run Command.
17. Wählen Sie in der Liste Document (Dokument) ein Systems Manager-Dokument (SSM-Dokument) aus, das Sie ausführen möchten, wenn das Ziel aufgerufen wird. Wählen Sie beispielsweise `AWS-RunPatchBaseline` als ein nicht konformes Patch-Ereignis oder `AWS-RefreshAssociation` als ein nicht konformes Zuweisungsereignis aus.
18. Geben Sie Informationen für die verbleibenden Felder und Parameter an.

> **Note**
>
> Erforderliche Felder und Parameter sind mit einem Sternchen (*) neben den Namen gekennzeichnet. Um ein Ziel zu erstellen, müssen Sie bei jedem erforderlichen Parameter oder Feld einen Wert angeben. Andernfalls erstellt das System die Regel, führt diese aber nicht aus.

19. Wählen Sie Weiter aus.
20. (Optional) Geben Sie ein oder mehrere Tags für die Regel ein. Weitere Informationen finden Sie unter Tagging Your Amazon EventBridge Resources (Taggen Ihrer Amazon EventBridge Resources) im Amazon EventBridge-Benutzerhandbuch.
21. Wählen Sie Weiter aus.
22. Überprüfen Sie die Details der Regel und wählen Sie dann Regel erstellen aus.

Weisen Sie benutzerdefinierte Compliance-Metadaten zu mithilfe der AWS CLI

Das folgende Verfahren führt Sie durch den Prozess, bei dem Sie mithilfe von AWS Command Line Interface (AWS CLI) den AWS Systems Manager PutComplianceItemsAPI-Vorgang aufrufen, um einer Ressource benutzerdefinierte Compliance-Metadaten zuzuweisen. Sie können mit dieser API-Operation zudem einem verwalteten Knoten manuell Patch- oder Zuordnungs-Compliance-Metadaten zuweisen, wie im folgenden Walkthrough dargestellt. Weitere Informationen zur Verwendung benutzerdefinierter Compliance finden Sie unter Informationen zu benutzerdefinierter Compliance.

So weisen Sie einer verwalteten Instance benutzerdefinierte Compliance-Metadaten zu (AWS CLI)

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um einem verwalteten Knoten benutzerdefinierte Compliance-Metadaten zuweisen. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen. Der `ResourceType`-Parameter unterstützt nur einen Wert von `ManagedInstance`. Geben Sie diesen Wert auch dann an, wenn Sie einem verwalteten AWS IoT Greengrass Kerngerät benutzerdefinierte Compliance-Metadaten zuweisen.

   Linux & macOS

   ```
   aws ssm put-compliance-items \
       --resource-id instance_ID \
       --resource-type ManagedInstance \
       --compliance-type Custom:user-defined_string \
       --execution-summary ExecutionTime=user-defined_time_and/or_date_value \
       --items Id=user-defined_ID,Title=user-
   defined_title,Severity=one_or_more_comma-separated_severities:CRITICAL, MAJOR,
    MINOR,INFORMATIONAL, or UNSPECIFIED,Status=COMPLIANT or NON_COMPLIANT
   ```

   Windows

   ```
   aws ssm put-compliance-items ^
       --resource-id instance_ID ^
       --resource-type ManagedInstance ^
       --compliance-type Custom:user-defined_string ^
       --execution-summary ExecutionTime=user-defined_time_and/or_date_value ^
       --items Id=user-defined_ID,Title=user-
   defined_title,Severity=one_or_more_comma-separated_severities:CRITICAL, MAJOR,
    MINOR,INFORMATIONAL, or UNSPECIFIED,Status=COMPLIANT or NON_COMPLIANT
   ```

3. Wiederholen Sie den vorherigen Schritt, um einem oder mehreren Knoten weitere benutzerdefinierte Compliance-Metadaten zuzuweisen. Mit folgenden Befehlen können Sie verwalteten Knoten die Patch- oder Zuordnungs-Compliance-Metadaten auch manuell zuweisen:

   Zuordnungs-Compliance-Metadaten

Linux & macOS

```
aws ssm put-compliance-items \
    --resource-id instance_ID \
    --resource-type ManagedInstance \
    --compliance-type Association \
    --execution-summary ExecutionTime=user-defined_time_and/or_date_value \
    --items Id=user-defined_ID,Title=user-
defined_title,Severity=one_or_more_comma-separated_severities:CRITICAL, MAJOR,
 MINOR,INFORMATIONAL, or UNSPECIFIED,Status=COMPLIANT or NON_COMPLIANT
```

Windows

```
aws ssm put-compliance-items ^
    --resource-id instance_ID ^
    --resource-type ManagedInstance ^
    --compliance-type Association ^
    --execution-summary ExecutionTime=user-defined_time_and/or_date_value ^
    --items Id=user-defined_ID,Title=user-
defined_title,Severity=one_or_more_comma-separated_severities:CRITICAL, MAJOR,
 MINOR,INFORMATIONAL, or UNSPECIFIED,Status=COMPLIANT or NON_COMPLIANT
```

Patch Compliance-Metadaten

Linux & macOS

```
aws ssm put-compliance-items \
    --resource-id instance_ID \
    --resource-type ManagedInstance \
    --compliance-type Patch \
    --execution-summary ExecutionTime=user-defined_time_and/
or_date_value,ExecutionId=user-defined_ID,ExecutionType=Command  \
    --items Id=for_example, KB12345,Title=user-
defined_title,Severity=one_or_more_comma-separated_severities:CRITICAL,
 MAJOR, MINOR,INFORMATIONAL, or UNSPECIFIED,Status=COMPLIANT or
 NON_COMPLIANT,Details="{PatchGroup=name_of_group,PatchSeverity=the_patch_severity,
 for example, CRITICAL}"
```

Windows

```
aws ssm put-compliance-items ^
    --resource-id instance_ID ^
    --resource-type ManagedInstance ^
    --compliance-type Patch ^
    --execution-summary ExecutionTime=user-defined_time_and/
or_date_value,ExecutionId=user-defined_ID,ExecutionType=Command  ^
    --items Id=for_example, KB12345,Title=user-
defined_title,Severity=one_or_more_comma-separated_severities:CRITICAL,
 MAJOR, MINOR,INFORMATIONAL, or UNSPECIFIED,Status=COMPLIANT or
 NON_COMPLIANT,Details="{PatchGroup=name_of_group,PatchSeverity=the_patch_severity,
 for example, CRITICAL}"
```

4. Führen Sie den folgenden Befehl aus, um eine Liste der Compliance-Elemente für einen bestimmten verwalteten Knoten anzuzeigen. Mithilfe von Filtern können Sie Details zu bestimmten Compliance-Daten anzeigen.

   Linux & macOS

   ```
   aws ssm list-compliance-items \
       --resource-ids instance_ID \
       --resource-types ManagedInstance \
       --filters one_or_more_filters
   ```

   Windows

   ```
   aws ssm list-compliance-items ^
       --resource-ids instance_ID ^
       --resource-types ManagedInstance ^
       --filters one_or_more_filters
   ```

   Die folgenden Beispiele zeigen, wie Sie diesen Befehl mit Filtern verwenden.

   Linux & macOS

   ```
   aws ssm list-compliance-items \
       --resource-ids i-02573cafcfEXAMPLE \
       --resource-type ManagedInstance \
   ```

```
    --filters Key=DocumentName,Values=AWS-RunPowerShellScript
 Key=Status,Values=NON_COMPLIANT,Type=NotEqual
 Key=Id,Values=cee20ae7-6388-488e-8be1-a88ccEXAMPLE
 Key=Severity,Values=UNSPECIFIED
```

Windows

```
aws ssm list-compliance-items ^
    --resource-ids i-02573cafcfEXAMPLE ^
    --resource-type ManagedInstance ^
    --filters Key=DocumentName,Values=AWS-RunPowerShellScript
 Key=Status,Values=NON_COMPLIANT,Type=NotEqual
 Key=Id,Values=cee20ae7-6388-488e-8be1-a88ccEXAMPLE
 Key=Severity,Values=UNSPECIFIED
```

Linux & macOS

```
aws ssm list-resource-compliance-summaries \
    --filters Key=OverallSeverity,Values=UNSPECIFIED
```

Windows

```
aws ssm list-resource-compliance-summaries ^
    --filters Key=OverallSeverity,Values=UNSPECIFIED
```

Linux & macOS

```
aws ssm list-resource-compliance-summaries \
    --filters Key=OverallSeverity,Values=UNSPECIFIED
 Key=ComplianceType,Values=Association Key=InstanceId,Values=i-02573cafcfEXAMPLE
```

Windows

```
aws ssm list-resource-compliance-summaries ^
    --filters Key=OverallSeverity,Values=UNSPECIFIED
 Key=ComplianceType,Values=Association Key=InstanceId,Values=i-02573cafcfEXAMPLE
```

5. Führen Sie den folgenden Befehl aus, um eine Übersicht der Compliance-Statusarten anzuzeigen. Mithilfe von Filtern können Sie Details zu bestimmten Compliance-Daten anzeigen.

```
aws ssm list-resource-compliance-summaries --filters One or more filters.
```

Die folgenden Beispiele zeigen, wie Sie diesen Befehl mit Filtern verwenden.

Linux & macOS

```
aws ssm list-resource-compliance-summaries \
    --filters Key=ExecutionType,Values=Command
```

Windows

```
aws ssm list-resource-compliance-summaries ^
    --filters Key=ExecutionType,Values=Command
```

Linux & macOS

```
aws ssm list-resource-compliance-summaries \
    --filters Key=AWS:InstanceInformation.PlatformType,Values=Windows
 Key=OverallSeverity,Values=CRITICAL
```

Windows

```
aws ssm list-resource-compliance-summaries ^
    --filters Key=AWS:InstanceInformation.PlatformType,Values=Windows
 Key=OverallSeverity,Values=CRITICAL
```

6. Mit dem folgenden Befehl zeigen Sie eine Übersichtszahl der konformen und nicht konformen Ressourcen für einen Compliance-Typ an. Mithilfe von Filtern können Sie Details zu bestimmten Compliance-Daten anzeigen.

```
aws ssm list-compliance-summaries --filters One or more filters.
```

Die folgenden Beispiele zeigen, wie Sie diesen Befehl mit Filtern verwenden.

Linux & macOS

```
aws ssm list-compliance-summaries \
```

```
    --filters Key=AWS:InstanceInformation.PlatformType,Values=Windows
 Key=PatchGroup,Values=TestGroup
```

Windows

```
aws ssm list-compliance-summaries ^
    --filters Key=AWS:InstanceInformation.PlatformType,Values=Windows
 Key=PatchGroup,Values=TestGroup
```

Linux & macOS

```
aws ssm list-compliance-summaries \
    --filters Key=AWS:InstanceInformation.PlatformType,Values=Windows
 Key=ExecutionId,Values=4adf0526-6aed-4694-97a5-14522EXAMPLE
```

Windows

```
aws ssm list-compliance-summaries ^
    --filters Key=AWS:InstanceInformation.PlatformType,Values=Windows
 Key=ExecutionId,Values=4adf0526-6aed-4694-97a5-14522EXAMPLE
```

AWS Systems Manager Distributor

Distributor, ein Tool in AWS Systems Manager, hilft Ihnen dabei, Software zu verpacken und auf AWS Systems Manager verwalteten Knoten zu veröffentlichen. Sie können Ihre eigene Software verpacken und veröffentlichen oder Distributor zum Suchen und Veröffentlichen AWS bereitgestellter Agentensoftwarepakete (z. AmazonCloudWatchAgentB.) oder Pakete von Drittanbietern wie Trend Micro verwenden.Durch die Veröffentlichung eines Pakets werden bestimmte Versionen des Paketdokuments auf verwalteten Knoten angekündigt, die Sie anhand von Knoten IDs AWS-Konto IDs, Tags oder einem identifizieren. AWS-Region Um mit Distributor zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Distributor aus.

Nachdem Sie in Distributor ein Paket erstellt haben, können Sie das Paket auf eine der folgenden Weisen installieren:

- Einmalig mithilfe von AWS Systems Manager Run Command
- Anhand eines Zeitplans mithilfe von AWS Systems Manager State Manager

> **Important**
>
> Pakete, die von Drittanbietern vertrieben werden, werden nicht vom Anbieter des Pakets verwaltet AWS und von diesem veröffentlicht. Wir empfehlen Ihnen, zusätzliche Sorgfaltsprüfungen durchzuführen, um die Einhaltung Ihrer internen Sicherheitskontrollen sicherzustellen. Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Dies wird als Modell der geteilten Verantwortung beschrieben. Weitere Informationen hierzu finden Sie in Modell der geteilten Verantwortung.

Welche Vorteile bietet Distributor meiner Organisation?

Distributor bietet die folgenden Vorteile:

- Ein Paket, viele Plattformen

  Wenn Sie ein Paket in Distributor erstellen, erstellt das System ein AWS Systems Manager - Dokument (SSM-Dokument). Sie können ZIP-Dateien an dieses Dokument anfügen. Wenn Sie Distributor ausführen, verarbeitet das System die Anweisungen im SSM-Dokument und installiert das Softwarepaket in der ZIP-Datei auf den angegebenen Zielen. Distributor unterstützt mehrere Betriebssysteme, darunter Windows, Ubuntu Server, Debian Server und Red Hat Enterprise Linux. Weitere Informationen zu unterstützten Plattformen finden Sie unter Unterstützte Paketplattformen und -architekturen.

- Kontrolle über den Paketzugriff über mehrere Gruppen verwalteter Instances hinweg

  Sie können Run Command oder State Manager verwenden, um zu steuern, welche Ihrer verwalteten Knoten ein Paket erhalten und welche Version dieses Paket haben soll. Run Command und State Manager sind Tools in AWS Systems Manager. Verwaltete Knoten können nach Instanz oder Gerät IDs, AWS-Konto Nummern, Tags oder gruppiert AWS-Regionen werden. Mithilfe von State Manager-Zuordnungen können Sie verschiedene Versionen eines Pakets für verschiedene Gruppen von Instances bereitstellen.

- Viele AWS Agentenpakete sind enthalten und sofort einsatzbereit

  Distributorenthält viele AWS Agentenpakete, die Sie sofort auf verwalteten Knoten bereitstellen können. Suchen Sie auf der Distributor `Packages`-Listenseite nach Paketen, die von `Amazon` veröffentlicht werden. Beispiele hierfür sind `AmazonCloudWatchAgent` und `AWSPVDriver`.

- Automatische Bereitstellung

Um Ihre Umgebung auf dem aktuellen Stand zu halten, verwenden Sie State Manager, um Pakete für die automatische Bereitstellung auf ausgesuchten verwalteten Knoten zu planen, wenn diese Maschinen zum ersten Mal gestartet werden.

An wen richtet sich Distributor?

- Jeder AWS Kunde, der neue Softwarepakete erstellen oder bestehende Softwarepakete, einschließlich AWS veröffentlichter Pakete, auf mehreren von Systems Manager verwalteten Knoten gleichzeitig bereitstellen möchte.
- Softwareentwickler, die Softwarepakete erstellen.
- Administratoren, die dafür verantwortlich sind, die von Systems Manager verwalteten Knoten mit den meisten up-to-date Softwarepaketen auf dem neuesten Stand zu halten.

Über welche Features verfügt Distributor?

- Bereitstellung von Paketen auf Windows- ebenso wie auf Linux-Instances

  Mit Distributor können Sie Softwarepakete auf Amazon Elastic Compute Cloud (Amazon EC2) - Instances und AWS IoT Greengrass Kerngeräten für Linux und bereitstellenWindows Server. Eine Liste der unter den jeweiligen Betriebssystemen unterstützten Instance-Typen finden Sie unter the section called “Unterstützte Paketplattformen und -architekturen”.

  > **Note**
  >
  > Distributor wird auf dem macOS-Betriebssystem unterstützt.

- Einmalige Bereitstellung von Paketen oder nach automatisiertem Zeitplan

  Sie können wählen, ob die Pakete einmalig, nach einem regelmäßigen Zeitplan oder immer dann, wenn die Standardpaketversion auf eine andere umgestellt wird, aktualisiert werden sollen.

- Vollständige Neuinstallation von Paketen oder Durchführen von direkten Aktualisierungen

  Um eine neue Paketversion zu installieren, können Sie die aktuelle Version vollständig deinstallieren und stattdessen eine neue Version installieren oder die aktuelle Version nur entsprechend einem von Ihnen bereitgestellten Aktualisierungsskript mit neuen und aktualisierten Komponenten aktualisieren. Ihre Paketanwendung ist während einer Neuinstallation nicht

verfügbar, kann aber während einer direkten Aktualisierung weiterhin verfügbar bleiben. Direkte Aktualisierungen sind besonders nützlich für Anwendungen zur Sicherheitsüberwachung oder andere Szenarien, in denen Sie Anwendungsausfälle vermeiden müssen.

- Zugriff auf Distributor Funktionen über Konsole PowerShell, CLI und SDK

  Sie können mit Distributor der Systems Manager Manager-Konsole AWS Command Line Interface (AWS CLI) oder dem AWS SDK Ihrer Wahl arbeiten. AWS -Tools für PowerShell

- IAM-Zugriffskontrolle

  Mithilfe von AWS Identity and Access Management (IAM-) Richtlinien können Sie steuern, welche Mitglieder Ihrer Organisation Pakete oder Paketversionen erstellen, aktualisieren, bereitstellen oder löschen können. Beispiel: Sie möchten einem Administrator Berechtigungen zum Bereitstellen von Paketen gewähren, nicht jedoch zum Ändern von Paketen oder zum Erstellen neuer Paketversionen.

- Support für Protokollierungs- und Prüfungsfunktionen

  AWS-Konto Durch die Integration mit anderen AWS-Services können Sie Distributor Benutzeraktionen in Ihrem System prüfen und protokollieren. Weitere Informationen finden Sie unter Prüfen und Protokollieren von Distributor-Aktivitäten.

Was ist ein Paket in Distributor?

Ein Paket ist eine Sammlung installierbarer Software oder Komponenten. Beispiele hierfür sind:

- Eine ZIP-Datei mit Software pro Ziel-Betriebssystemplattform. Jede ZIP-Datei muss Folgendes enthalten:
  - Ein install und ein uninstall Skript. Windows Serverbasierte verwaltete Knoten benötigen PowerShell Skripten (Skripten mit dem Namen `install.ps1` und`uninstall.ps1`). Linux-basierte verwaltete Knoten benötigen Shell-Skripten (Skripten mit dem Namen `install.sh` und`uninstall.sh`). AWS Systems Manager SSM Agentliest die Anweisungen in den uninstall UND-Skripten install und führt sie aus.
  - Eine ausführbare Datei. Diese muss SSM Agent finden, um das Paket auf den anvisierten verwalteten Knoten installieren zu können.
- Eine Manifestdatei im JSON-Format, die den Paketinhalt beschreibt. Das Manifest ist nicht in der ZIP-Datei enthalten, aber im selben Amazon Simple Storage Service (Amazon S3)-Bucket gespeichert wie die ZIP-Dateien, aus denen das Paket besteht. Das Manifest identifiziert die Paketversion und ordnet die ZIP-Dateien im Paket den Attributen des anvisierten verwalteten

Knotens zu (z. B. die Version oder Architektur des Betriebssystems). Informationen zum Erstellen des Manifests finden Sie unter Schritt 2: Erstellen des JSON-Paketmanifests.

Wenn Sie Einfache Paketerstellung in der Distributor-Konsole wählen, generiert Distributor die Installations- und Deinstallationsskripts, sowie die Datei-Hashes und das Manifest des JSON-Pakets, basierend auf dem Dateinamen der ausführbaren Software sowie den Zielplattformen und -Architekturen.

Unterstützte Paketplattformen und -architekturen

Sie können Distributor verwenden, um Pakete auf den folgenden Plattformen verwalteter Knoten von Systems Manager zu veröffentlichen. Ein Versionswert muss genau mit der Version des Betriebssystems-Amazon Machine Image (AMI) übereinstimmen. Weitere Informationen zum Ermitteln dieser Version finden Sie in Schritt 4 unter Schritt 2: Erstellen des JSON-Paketmanifests.

> **Note**
>
> Systems Manager unterstützt nicht alle der folgenden Betriebssysteme für AWS IoT Greengrass Kerngeräte. Weitere Informationen finden Sie im AWS IoT Greengrass Version 2 Entwicklerhandbuch unter Einrichten von AWS IoT Greengrass Kerngeräten.

Plattform	Codewert in der Manifestdatei	Unterstützte Architekturen
AlmaLinux	`almalinux`	x86_64 ARM64
Amazon Linux 2 und Amazon Linux 2023	`amazon`	x86_64 oder x86 ARM64(Amazon Linux 2 und AL2 023, A1-Instance-Typen)
Debian Server	`debian`	x86_64 oder x86
openSUSE	`opensuse`	x86_64
openSUSE Leap	`opensuseleap`	x86_64

Plattform	Codewert in der Manifestdatei	Unterstützte Architekturen
Oracle Linux	`oracle`	x86_64
Red Hat Enterprise Linux (RHEL)	`redhat`	x86_64 ARM64 (RHEL 7.6 und neuer, A1-Instance-Typen)
Rocky Linux	`rocky`	x86_64 ARM64
SUSE Linux Enterprise Server (SLES)	`suse`	x86_64
Ubuntu Server	`ubuntu`	x86_64 oder x86 ARM64 (Ubuntu Server 16 und neuer, A1-Instance-Typen)
Windows Server	`windows`	x86_64

Themen

- Einrichten von Distributor
- Mit Distributor-Paketen arbeiten
- Prüfen und Protokollieren von Distributor-Aktivitäten
- Fehlerbehebung für AWS Systems ManagerDistributor

Einrichten von Distributor

Bevor Sie Distributor, ein Tool in AWS Systems Manager, zum Erstellen, Verwalten und Bereitstellen von Softwarepaketen verwenden, führen Sie die folgenden Schritte aus.

Erfüllen von Distributor-Voraussetzungen

Bevor Sie Distributor verwenden, ein Tool in AWS Systems Manager, stellen Sie sicher, dass Ihre Umgebung den folgenden Anforderungen entspricht:

Distributor-Voraussetzungen

Anforderung	Beschreibung
SSM Agent	Auf den verwalteten Knoten, auf denen Pakete bereitgestellt bzw. aus denen Pakete entfernt werden sollen, muss AWS Systems Manager SSM Agent Version 2.3.274.0 oder höher installiert sein. Informationen zum Installieren oder Aktualisieren von SSM Agent finden Sie unter Arbeiten mit SSM Agent.
AWS CLI	(Optional) Um die AWS Command Line Interface (AWS CLI) anstelle der Systems Manager-Konsole zum Erstellen und Verwalten von Paketen zu verwenden, installieren Sie die neueste Version der AWS CLI auf Ihrem lokalen Computer. Weitere Informationen zum Installieren oder Upgraden der CLI finden Sie unter Installieren der AWS Command Line Interface um AWS Command Line Interface-Benutzerhandbuch.
AWS -Tools für PowerShell	(Optional) Um die Tools for PowerShell anstelle der Systems Manager-Konsole zum Erstellen und Verwalten von Paketen zu verwenden, installieren Sie die neueste Version der Tools for PowerShell auf Ihrem lokalen Computer. Weitere Informationen zum Installieren oder Aktualisieren der Tools für PowerShell finden Sie unter Einrichten der AWS Tools

Anforderung	Beschreibung
	for Windows PowerShell oder AWS Tools for PowerShell Core im AWS -Tools für PowerShel l-Benutzerhandbuch.

> **Note**
>
> Systems Manager unterstützt nicht die Verteilung von Paketen an von Oracle Linux verwaltete Knoten mit Distributor.

Überprüfen oder Erstellen eines IAM-Instance-Profils mit Distributor-Berechtigungen

Standardmäßig hat AWS Systems Manager keine Berechtigung zur Ausführung von Aktionen auf Ihren Instances. Sie müssen mit einem AWS Identity and Access Management-(IAM)-Instance-Profil Zugriff gewähren. Ein Instance-Profil ist ein Container, der Informationen zur IAM-Rolle beim Start an eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance übergibt. Diese Anforderung gilt für Berechtigungen für alle Systems-Manager-Tools, nicht nur Distributor.

> **Note**
>
> Wenn Sie Ihre Edge-Geräte für die Ausführung von AWS IoT Greengrass-Core-Software und SSM Agent konfigurieren, geben Sie eine IAM-Servicerolle an, die es dem Systems Manager ermöglicht, Aktionen darauf auszuführen. Sie müssen keine verwalteten Edge-Geräte mit einem Instance-Profil konfigurieren.

Wenn Sie bereits andere Systems-Manager-Tools verwenden, z. B. Run Command und State Manager, ist ein Instance-Profil mit den erforderlichen Berechtigungen für Distributor bereits Ihren Instances zugeordnet. Die einfachste Möglichkeit, um sicherzustellen, dass Sie über Berechtigungen zur Durchführung von Distributor-Aufgaben verfügen, besteht darin, Ihrem Instance-Profil die Richtlinie AmazonSSMManagedInstanceCore anzufügen. Weitere Informationen finden Sie unter Erforderliche Instance-Berechtigungen für Systems Manager konfigurieren.

Kontrolle des Benutzerzugriffs auf Pakete

Mit AWS Identity and Access Management-(IAM)-Richtlinien können Sie steuern, wer Pakete erstellen, bereitstellen und verwalten kann. Sie können auch steuern, welche Run Command- und State Manager-API-Operationen auf verwalteten Knoten ausgeführt werden können. Wie Distributor sind Run Command und State Manager Tools in AWS Systems Manager.

ARN-Format

Benutzerdefinierte Pakete sind Dokument-Amazon Resource Names (ARNs) zugeordnet und haben das folgende Format.

```
arn:aws:ssm:region:account-id:document/document-name
```

Im Folgenden wird ein Beispiel gezeigt.

```
arn:aws:ssm:us-west-1:123456789012:document/ExampleDocumentName
```

Sie können zwei von AWS bereitgestellte IAM-Standardrichtlinien verwenden, eine für Endbenutzer und eine für Administratoren, um Berechtigungen für Distributor-Aktivitäten zu erteilen. Sie können auch benutzerdefinierte IAM-Richtlinien erstellen, die an Ihre Berechtigungsanforderungen angepasst sind.

Weitere Informationen zur Verwendung von Variablen in IAM-Richtlinien finden Sie unter IAM-Richtlinienelemente: Variablen.

Informationen zum Erstellen von Richtlinien und zum Anfügen an Benutzer oder Gruppen finden Sie unter Erstellen von IAM-Richtlinien und Hinzufügen und Entfernen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Erstellen oder Auswählen eines Amazon-S3-Buckets Distributor

Wenn Sie ein Paket erstellen, indem Sie die in der AWS Systems Manager-Konsole den Workflow Einfach auswählen, wählen Sie einen vorhandenen Amazon Simple Storage Service (Amazon S3)-Bucket aus, an den Distributor Ihre Software hochlädt. Distributor ist ein Tool in AWS Systems Manager Wenn Sie den Workflow Advanced (Erweitert) auswählen, müssen Sie Ihre Software oder Komponenten als ZIP-Dateien in einen Amazon-S3-Bucket hochladen, bevor Sie beginnen. Unabhängig davon, ob Sie ein Paket mit dem Workflow Simple (Einfach) oder Advanced (Erweitert) in der Konsole erstellen, oder aber ob Sie die API verwenden, Sie benötigen einen Amazon-S3-Bucket,

bevor Sie mit der Erstellung Ihres Paket beginnen. Bei der Paketerstellung kopiert Distributor Ihre installierbare Software und die Komponenten aus diesem Bucket in einen internen Systems Manager-Speicher. Da die Komponenten in einen internen Speicher kopiert werden, können Sie Ihren Amazon-S3-Bucket löschen oder wiederverwenden, wenn die Paketerstellung abgeschlossen ist.

Weitere Informationen zur Erstellung eines Buckets finden Sie unter Erstellen eines Buckets im Amazon Simple Storage Service Getting Started Guide. Weitere Informationen zum Ausführen eines AWS CLI-Befehls zum Erstellen eines Buckets finden Sie unter mb in der AWS CLI-Befehlsreferenz.

Mit Distributor-Paketen arbeiten

Sie können die AWS Systems Manager-Konsole, die AWS-Befehlszeilen-Tools (AWS CLI und AWS -Tools für PowerShell) und die AWS-SDKs zum Hinzufügen, Verwalten oder Bereitstellen von Paketen in Distributor verwenden. Distributor ist ein Tool in AWS Systems Manager. Vor dem Hinzufügen eines Pakets zu Distributor:

- Erstellen und zippen Sie die zu installierbaren Komponenten.
- (Optional) Erstellen Sie eine JSON-Manifestdatei für das Paket. Dies ist nicht erforderlich, um den Prozess der einfachen Paketerstellung in der Distributor-Konsole zu nutzen. Bei der einfachen Paketerstellung wird die JSON-Manifestdatei automatisch generiert.

  Sie können zum Erstellen der Manifestdatei die AWS Systems Manager-Konsole oder einen Text- oder JSON-Editor verwenden.
- Halten Sie einen Amazon Simple Storage Service (Amazon S3)-Bucket bereit, um Ihre installierbaren Komponenten oder Software zu speichern. Wenn Sie die den Advanced (Erweitert)-Workflow zur Paketerstellung verwenden, laden Sie Ihre Komponenten an den Amazon-S3-Bucket herunter, bevor Sie beginnen.

> **Note**
>
> Sie können diesen Bucket nach Abschluss der Erstellung Ihres Pakets löschen oder anderweitig verwenden, weil Distributor im Rahmen des Paketerstellungsprozesses die Paketinhalte in einen internen Systems Manager-Bucket verschiebt.

Von AWS veröffentlichte Pakete sind bereits verpackt und können sofort bereitgestellt werden. Informationen zum Bereitstellen eines von AWS veröffentlichten Pakets an verwalteten Knoten finden Sie unter Installieren oder Aktualisieren von Distributor-Paketen.

Sie können Distributor-Pakete zwischen AWS-Konten tauschen. Wenn Sie ein Paket verwenden, das von einem anderen Konto in AWS CLI-Befehlen geteilt wird, verwenden Sie das Paket Amazon Resource Name (ARN) anstelle des Paketnamens.

Themen

- Pakete anzeigen in Distributor
- Erstellen eines Pakets in Distributor
- Bearbeiten von Distributor-Paketberechtigungen in der Konsole
- Bearbeiten von Distributor-Paket-Tags in der Konsole
- Hinzufügen einer Version zu einem Distributor-Paket
- Installieren oder Aktualisieren von Distributor-Paketen
- Deinstallieren eines Distributor-Pakets
- Löschen eines Distributor-Pakets

Pakete anzeigen in Distributor

Um Pakete anzuzeigen, die für die Installation verfügbar sind, können Sie die AWS Systems Manager-Konsole oder Ihr bevorzugtes AWS-Befehlszeilen-Tool verwenden. Distributor ist ein Tool in AWS Systems Manager. Um auf Distributor zuzugreifen, öffnen Sie die AWS Systems Manager-Konsole und wählen Sie im linken Navigationsbereich Distributor aus. Sie werden alle Pakete sehen, die Ihnen zur Verfügung stehen.

Im folgenden Abschnitt wird beschrieben, wie Sie Ihre Distributor-Pakete mithilfe Ihrer bevorzugten Befehlszeilen-Tools anzeigen.

Anzeigen des Pakets per Befehlszeile

Dieser Abschnitt enthält Informationen dazu, wie Sie Ihr bevorzugtes Befehlszeilen-Tool verwenden können, um Distributor-Pakete mit den bereitgestellten Befehlen anzuzeigen.

Linux & macOS

Anzeigen von Paketen mit AWS CLI unter Linux

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, mit der Ausnahme freigegebener Pakete.

```
aws ssm list-documents \
```

```
    --filters Key=DocumentType,Values=Package
```

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, die Amazon gehören.

```
aws ssm list-documents \
    --filters Key=DocumentType,Values=Package Key=Owner,Values=Amazon
```

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, die Drittanbietern gehören.

```
aws ssm list-documents \
    --filters Key=DocumentType,Values=Package Key=Owner,Values=ThirdParty
```

Windows

Anzeigen von Paketen mit AWS CLI unter Windows

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, mit der Ausnahme freigegebener Pakete.

```
aws ssm list-documents ^
    --filters Key=DocumentType,Values=Package
```

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, die Amazon gehören.

```
aws ssm list-documents ^
    --filters Key=DocumentType,Values=Package Key=Owner,Values=Amazon
```

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, die Drittanbietern gehören.

```
aws ssm list-documents ^
    --filters Key=DocumentType,Values=Package Key=Owner,Values=ThirdParty
```

PowerShell

Anzeigen von Packages mit Tools for PowerShell

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, mit der Ausnahme freigegebener Pakete.

```
$filter = New-Object Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
```

```
$filter.Key = "DocumentType"
$filter.Values = "Package"

Get-SSMDocumentList `
    -Filters @($filter)
```

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, die Amazon gehören.

```
$typeFilter = New-Object
 Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$typeFilter.Key = "DocumentType"
$typeFilter.Values = "Package"

$ownerFilter = New-Object
 Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$ownerFilter.Key = "Owner"
$ownerFilter.Values = "Amazon"

Get-SSMDocumentList `
    -Filters @($typeFilter,$ownerFilter)
```

- Führen Sie den folgenden Befehl aus, um alle Pakete anzuzeigen, die Drittanbietern gehören.

```
$typeFilter = New-Object
 Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$typeFilter.Key = "DocumentType"
$typeFilter.Values = "Package"

$ownerFilter = New-Object
 Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$ownerFilter.Key = "Owner"
$ownerFilter.Values = "ThirdParty"

Get-SSMDocumentList `
    -Filters @($typeFilter,$ownerFilter)
```

Erstellen eines Pakets in Distributor

Um ein Paket zu erstellen, bereiten Sie Ihre installierbare Software oder Komponenten vor, immer eine Datei pro Betriebssystem-Plattform. Zur Erstellung eines Pakets ist mindestens eine Datei erforderlich.

Manchmal verwenden unterschiedliche Plattformen dieselbe Datei. Alle Ihrem Paket hinzugefügten Dateien müssen jedoch im Abschnitt `Files` des Manifests aufgelistet sein. Wenn Sie ein Paket in der Konsole über den einfachen Workflow erstellen, wird das Manifest automatisch generiert. Die maximale Anzahl von Dateien, die Sie einem einzelnen Dokument anfügen können, beträgt 20. Die maximale Größe der einzelnen Dateien beträgt 1 GB. Weitere Informationen zu unterstützten Plattformen finden Sie unter Unterstützte Paketplattformen und -architekturen.

Wenn Sie ein neues Paket erstellen, erstellt das System ein neues SSM-Dokument. Mit diesem Dokument können Sie das Paket an verwaltete Knoten bereitstellen.

Nur zu Demonstrationszwecken steht Ihnen ein Beispielpaket, ExamplePackage.zip, zum Herunterladen von unserer Website zur Verfügung. Das Beispielpaket enthält ein vollständiges JSON-Manifest und drei .zip-Dateien mit Installationsprogrammen für Version 7.0.0. PowerShell Die Installations- und Deinstallationsskripts enthalten keine gültigen Befehle. Wenn Sie ein Paket im Workflow Advanced (Erweitert) erstellen, müssen Sie alle installierbaren Softwaredateien und Skripts in einer ZIP-Datei komprimieren, beim Workflow Simple (Einfach) ist es jedoch nicht notwendig, installierbare Komponenten zu zippen.

Themen

- Erstellen Sie ein Paket mithilfe des einfachen Workflows
- Erstellen Sie ein Paket mithilfe des erweiterten Workflows

Erstellen Sie ein Paket mithilfe des einfachen Workflows

In diesem Abschnitt wird beschrieben, wie Sie ein Paket in Distributor erstellen, wenn Sie das Paket über den Einfach-Workflow in der Distributor-Konsole erstellen. Distributor ist ein Tool in AWS Systems Manager. Um ein Paket zu erstellen, bereiten Sie Ihre zu installierenden Komponenten vor, eine Datei pro Betriebssystemplattform. Zur Erstellung eines Pakets ist mindestens eine Datei erforderlich. Bei der einfachen Paketerstellung werden die Installations- und Deinstallationsskripts generiert, sowie die Datei-Hashes und eine Manifest-Datei im JSON-Format. Der Simple (Einfach)-Workflow übernimmt das Hochladen und Komprimieren Ihrer installierbaren Dateien sowie das Erstellen eines neuen Pakets und des zugehörigen SSM-Dokuments. Weitere Informationen zu unterstützten Plattformen finden Sie unter Unterstützte Paketplattformen und -architekturen.

Wenn Sie die Methode „Simple (Einfach)“ verwenden, um ein Paket zu erstellen, erstellt Distributor die `install`- und `uninstall`-Skripts für Sie. Wenn Sie jedoch ein Paket für ein direktes Update erstellen, müssen Sie Ihren eigenen `update`-Skript-Inhalt in der Registerkarte Update script

bereitstellen.. Wenn Sie Eingabebefehle für ein `update`-Skript hinzufügen, schließt Distributor dieses Skript zusammen mit den `install`- und `uninstall`-Skripts in das für Sie erstellte ZIP-Paket ein.

> **Note**
>
> Verwenden Sie die Aktualisierungsoption `In-place` zum Hinzufügen neuer oder aktualisierter Dateien einer vorhandenen Paketinstallation, ohne die zugehörige Anwendung offline zu schalten.

Erstellen Sie ein Paket mithilfe des einfachen Workflows

1. Öffnen Sie die Konsole unter. AWS Systems Manager https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Distributor-Startseite die Option Create package (Paket erstellen) aus und wählen Sie dann die Option Simple (Einfach).
4. Geben Sie auf der Seite Create package (Paket erstellen) einen Namen für Ihr Paket ein. Paketnamen können Buchstaben, Zahlen, Punkte, Bindestriche und Unterstriche enthalten. Der Name sollte allgemein genug sein, um auf alle Versionen der Paketanhänge angewendet werden zu können, jedoch spezifisch genug, um den Zweck des Pakets zu identifizieren.
5. (Optional) Geben Sie unter Version name (Versionsname) einen Versionsnamen ein. Versionsnamen dürfen maximal 512 Zeichen lang sein und dürfen keine Sonderzeichen enthalten.
6. Wählen Sie unter Location (Speicherort) einen Bucket aus. Verwenden Sie dazu den Namen und das Präfix des Buckets oder die Bucket-URL.
7. Wählen Sie unter Upload software (Software hochladen) die Option Add software (Software hinzufügen) aus und navigieren Sie dann zu installierbaren Softwaredateien mit den Erweiterungen `.rpm`, `.msi`, oder `.deb`. Wenn der Dateiname Leerzeichen enthält, schlägt der Upload fehl. Sie können mehrere Softwaredateien in einer einzigen Aktion hochladen.
8. Überprüfen Sie unter Für Target platform (Ziel-Plattform für jede Plattform, ob das Ziel-Betriebssystem für die installierbare Datei korrekt ist. Wenn das angezeigte Betriebssystem nicht korrekt ist, wählen Sie das richtige Betriebssystem aus der Dropdown-Liste aus.

   Beim Workflow Simple (Einfach) zur Paketerstellung sind zusätzliche Schritte erforderlich, wenn Distributor nur eine Datei für mehrere Betriebssysteme verwenden soll, da installierbare Dateien

nur einmal hochgeladen werden. Wenn Sie zum Beispiel eine installierbare Softwaredatei mit dem Namen `Logtool_v1.1.1.rpm` hochladen, müssen Sie einige Standardeinstellungen für den Einfach-Workflow ändern, um unter Amazon-Linux- und Ubuntu Server-Betriebssystemen dieselbe Software als Ziel festzulegen. Führen Sie bei Verwendung mehrerer Zielplattformen einen der folgenden Schritte aus.

- Verwenden Sie stattdessen den Workflow Advanced (Erweitert), zippen Sie jede installierbare Datei, bevor Sie beginnen, und richten Sie das Manifest manuell so ein, dass eine installierbare Datei für mehrere Betriebssystemplattformen oder -versionen verwendet werden kann. Weitere Informationen finden Sie unter Erstellen Sie ein Paket mithilfe des erweiterten Workflows.
- Bearbeiten Sie die Manifestdatei im Workflow Simple (Einfach) so, dass Ihre ZIP-Datei für mehrere Betriebssystemplattformen oder -versionen verwendet wird. Weitere Informationen zu diesem Verfahren finden Sie am Ende von Schritt 4 in Schritt 2: Erstellen des JSON-Paketmanifests.

9. Stellen Sie unter Platform version (Plattformversion)sicher, dass als Betriebssystem-Plattformversion **_any**, eine Hauptversionsnummer, gefolgt von einem Platzhalter (7.*), angezeigt wird, oder genau die spezifische Betriebssystemversion, die Sie als Plattformversion für Ihre Softwareinstallation verwenden möchten. Weitere Informationen zur Angabe der Betriebssystem-Plattformversion finden Sie unter Schritt 4 in Schritt 2: Erstellen des JSON-Paketmanifests.
10. Wählen Sie unter Architecture (Architektur) für jeden installierbare Datei die richtige Prozessorarchitektur aus der Dropdown-Liste aus. Weitere Informationen zu unterstützten Prozessorarchitekturen finden Sie unter Unterstützte Paketplattformen und -architekturen.
11. (Optional) Erweitern Sie Scripts (Skripts) und überprüfen Sie die Skripts, die Distributor für Ihre installierbare Software generiert hat.
12. (Optional) Um ein Aktualisierungsskript für direkte Aktualisierungen bereitzustellen, erweitern Sie Scripts (Skripts), wählen Sie die Registerkarte Update script (Aktualisierungsskript) aus und geben Sie die Befehle für das Aktualisierungsskript ein.

    Systems Manager generiert keine Aktualisierungsskripts für Sie.
13. Zum Hinzufügen weiterer installierbaren Softwaredateien wählen Sie Add Software (Software hinzufügen). Andernfalls fahren Sie mit dem nächsten Schritt fort.
14. (Optional) Erweitern Sie Manifest und überprüfen Sie das JSON-Manifest des Pakets, das Distributor für Ihre installierbare Software generiert hat. Wenn Sie Informationen über Ihre Software geändert haben, nachdem Sie mit dieser Prozedur begonnen haben, beispielsweise die

Plattformversion oder die Zielplattform, wählen Sie Generate Manifest (Manifest erzeugen), um das Paketmanifest zu aktualisieren.

Sie können das Manifest manuell bearbeiten, wenn Sie möchten, dass für eine installierbare Software mehr als ein Betriebssystem als Ziel festgelegt wird, wie in Schritt 8 beschrieben. Weitere Informationen zum Bearbeiten des Manifests finden Sie unter Schritt 2: Erstellen des JSON-Paketmanifests.

15. Wählen Sie Create package (Paket erstellen) aus.

Warten Sie, bis Distributor das Hochladen Ihrer Software und das Erstellen Ihres Pakets abgeschlossen hat. Distributor zeigt den Uploadstatus für jede installierbare Datei einzeln an. Abhängig von der Anzahl und Größe der Pakete, die Sie hinzufügen, kann dies einige Minuten dauern. Distributor leitet Sie automatisch auf die Seite Package details (Paketdetails) für das neue Paket weiter, Sie können diese Seite aber jederzeit selbst öffnen, nachdem die Software hochgeladen ist. Auf der Seite Package details (Paketdetails) werden erst dann alle Informationen zu Ihrem Paket angezeigt, wenn Distributor den Paketerstellungsprozess abgeschlossen hat. Um den Upload- und Paketerstellungsprozess abzubrechen, wählen Sie Cancel (Abbrechen).

Wenn Distributor keine installierbaren Softwaredateien hochladen kann, wird eine Fehlermeldung Upload failed (Upload fehlgeschlagen) angezeigt. Um den Uploadversuch zu wiederholen, wählen Sie Retry Upload (Uploadversuch wiederholen). Weitere Informationen zur Fehlerbehebung bei der Paketerstellung finden Sie unter Fehlerbehebung für AWS Systems ManagerDistributor.

Erstellen Sie ein Paket mithilfe des erweiterten Workflows

In diesem Abschnitt erfahren Sie, wie fortgeschrittene Benutzer ein Paket in Distributor erstellen können, nachdem sie die installierbaren Komponenten mit den Skripts zur Installation und Deinstallation, sowie eine JSON-Manifestdatei als ZIP-Archiv an einen Amazon S3-Bucket hochgeladen haben.

Um ein Paket zu erstellen, bereiten Sie Ihre ZIP-Dateien mit den zu installierenden Komponenten vor (eine ZIP-Datei pro Betriebssystemplattform). Zur Erstellung eines Pakets ist mindestens eine ZIP-Datei erforderlich. Erstellen Sie als Nächstes ein JSON-Manifest. Das Manifest enthält Verweise auf Ihre Paketcodedateien. Wenn Sie die erforderlichen Codedateien zu einem Ordner oder Verzeichnis hinzugefügt haben und das Manifest mit den korrekten Werten ausgefüllt ist, laden Sie Ihr Paket an einen S3-Bucket hoch.

Ein Beispielpaket, ExamplePackage.zip, steht für Sie zum Herunterladen von unserer Website zur Verfügung. Das Beispielpaket enthält ein fertiges JSON-Manifest und drei ZIP-Dateien.

Themen

- Schritt 1: Erstellen der ZIP-Dateien
- Schritt 2: Erstellen des JSON-Paketmanifests
- Schritt 3: Hochladen von Paket und Manifest zu einem Amazon S3-Bucket
- Schritt 4: Hinzufügen eines Pakets zu Distributor

Schritt 1: Erstellen der ZIP-Dateien

Die Grundlage Ihres Pakets ist mindestens eine ZIP-Datei mit Softwaredateien oder zu installierenden Komponenten. Ein Paket enthält eine ZIP-Datei pro Betriebssystem, das Sie unterstützen möchten, es sei denn, eine ZIP-Datei kann auf mehreren Betriebssystemen installiert werden. Beispielsweise können unterstützte Versionen von Red Hat Enterprise Linux- und Amazon-Linux-Instances in der Regel dieselben ausführbaren RPM-Dateien ausführen. Daher müssen Sie Ihrem Paket nur eine ZIP-Datei anfügen, um beide Betriebssysteme zu unterstützen.

Erforderliche Dateien

Die folgenden Elemente müssen in jeder ZIP-Datei enthalten sein:

- Ein install und ein uninstall Skript. Windows Serverbasierte verwaltete Knoten benötigen PowerShell Skripten (Skripten mit dem Namen `install.ps1` und`uninstall.ps1`). Linux-basierte verwaltete Knoten benötigen Shell-Skripts (Skripts mit dem Namen `install.sh` und `uninstall.sh`). SSM Agent führt die Anweisungen in den install- und uninstall-Skripts aus.

  Ihre Installationsskripts können beispielsweise ein Installationsprogramm ausführen (z. B. eine RPM- oder MSI-Datei), Dateien kopieren oder Konfigurationseinstellungen festlegen.
- Eine ausführbare Datei, Installationsprogrammpakete (.rpm, .deb, .msi usw.), weitere Skripts oder Konfigurationsdateien.

Optionale Dateien

Die folgenden Elemente können optional in jeder ZIP-Datei enthalten sein:

- Ein update-Skript. Die Angabe eines Aktualisierungsskripts ermöglicht es Ihnen, die Option `In-place update` zum Installieren eines Pakets zu verwenden. Wenn Sie einer vorhandenen Paketinstallation neue oder aktualisierte Dateien hinzufügen möchten, wird die Paketanwendung bei dieser `In-place update` Option nicht offline geschaltet, während das Update ausgeführt

wird. Windows Serverbasierte verwaltete Knoten benötigen ein PowerShell Skript (mit dem Namen`update.ps1`). Linux-basierte verwaltete Knoten benötigen ein Shell-Skript (Skript mit dem Namen `update.sh`). SSM Agent führt die Anweisungen im update-Skript aus.

Weitere Informationen zum Installieren oder Aktualisieren von Paketen finden Sie unter Installieren oder Aktualisieren von Distributor-Paketen.

Für Beispiele für ZIP-Dateien, einschließlich Beispiel install - und uninstall Skripten, laden Sie das Beispielpaket (ExamplePackage.zip) herunter.

Schritt 2: Erstellen des JSON-Paketmanifests

Nachdem Sie die zu installierenden Dateien vorbereitet und gezippt haben, erstellen Sie ein JSON-Manifest. Im Folgenden finden Sie eine Vorlage. Die einzelnen Teile der Manifestvorlage werden im Verfahren in diesem Abschnitt beschrieben. Sie können einen JSON-Editor verwenden, um dieses Manifest in einer eigenen Datei zu erstellen. Alternativ können Sie das Manifest in der AWS Systems Manager Konsole erstellen, wenn Sie ein Paket erstellen.

```
{
  "schemaVersion": "2.0",
  "version": "your-version",
  "publisher": "optional-publisher-name",
  "packages": {
    "platform": {
      "platform-version": {
        "architecture": {
          "file": ".zip-file-name-1.zip"
        }
      }
    },
    "another-platform": {
      "platform-version": {
        "architecture": {
          "file": ".zip-file-name-2.zip"
        }
      }
    },
    "another-platform": {
      "platform-version": {
        "architecture": {
          "file": ".zip-file-name-3.zip"
        }
```

```
      }
    }
  },
  "files": {
    ".zip-file-name-1.zip": {
      "checksums": {
        "sha256": "checksum"
      }
    },
    ".zip-file-name-2.zip": {
      "checksums": {
        "sha256": "checksum"
      }
    }
  }
}
```

So erstellen Sie ein JSON-Paket-Manifest

1. Fügen Sie Ihrem Manifest die Schemaversion hinzu. In dieser Version ist die Schemaversion stets `2.0`.

   ```
   { "schemaVersion": "2.0",
   ```

2. Fügen Sie Ihrem Manifest eine benutzerdefinierte Paketversion hinzu. Dies ist auch der Wert in Version name (Versionsname), den Sie angeben, wenn Sie Ihr Paket zu Distributor hinzufügen. Er wird Teil des AWS Systems Manager -Dokuments, das Distributor erstellt, wenn Sie Ihr Paket hinzufügen. Sie stellen diesen Wert auch als Eingabewert im Dokument `AWS-ConfigureAWSPackage` bereit, um eine andere als die aktuelle Version des Pakets zu installieren. Ein `version`-Wert kann Buchstaben, Zahlen, Unterstriche, Bindestriche und Punkte enthalten. Er darf jedoch höchstens 128 Zeichen enthalten. Sie sollten eine von Menschen lesbare Paketversion verwenden, um bei Bereitstellungen die Angabe der genauen Paketversionen für Sie und andere Administratoren einfacher zu machen. Im Folgenden wird ein -Beispiel gezeigt.

   ```
   "version": "1.0.1",
   ```

3. (Optional) Fügen Sie den Namen des Publishers hinzu. Im Folgenden wird ein -Beispiel gezeigt.

   ```
   "publisher": "MyOrganization",
   ```

4. Fügen Sie Pakete hinzu. Der Abschnitt `"packages"` beschreibt die von den ZIP-Dateien in Ihrem Paket unterstützten Plattformen, Versionen und Architekturen. Weitere Informationen finden Sie unter Unterstützte Paketplattformen und -architekturen.

   Das *`platform-version`* kann der Platzhalterwert sein,`_any`. Sie verwenden den Platzhalterwert, um anzugeben, dass eine ZIP-Datei eine beliebige Version der Plattform unterstützt. Sie können auch eine Hauptversion und gefolgt von einem Platzhalter angeben, sodass alle Nebenversionen unterstützt werden, z. B. 7.*. Wenn Sie einen *`platform-version`* Wert für eine bestimmte Betriebssystemversion angeben möchten, stellen Sie sicher, dass er genau der Release-Version des Betriebssystems entspricht, auf AMI das Sie abzielen. Im Folgenden werden Ressourcen empfohlen, mit denen Sie den richtigen Wert für das Betriebssystem ermitteln können.

   - Auf einem Windows Server-basierten verwalteten Knoten ist die Version in Form von Windows Management Instrumentation (WMI)-Daten verfügbar. Sie können den folgenden Befehl im Prompt ausführen, um Versionsinformationen zu erhalten. Anschließend müssen Sie die Ergebnisse nach `version` durchsuchen.

     ```
     wmic OS get /format:list
     ```

   - Auf einem Linux-basierten verwalteten Knoten erhalten Sie die Version, indem Sie zunächst nach der Betriebssystemversion scannen (der folgende Befehl). Suchen Sie den Wert von `VERSION_ID`.

     ```
     cat /etc/os-release
     ```

     Wenn die ausgegebene Zeichenfolge nicht die benötigten Informationen enthält, führen Sie den folgenden Befehl aus, um die LSB-Versionsinformationen aus der Datei `/etc/lsb-release` abzurufen, und suchen den Wert von `DISTRIB_RELEASE`.

     ```
     lsb_release -a
     ```

     Wenn diese Methoden nicht zum Erfolg führen, finden Sie die Version in der Regel anhand der verwendeten Distribution. In Debian Server können Sie beispielsweise die Datei `/etc/debian_version` und in Red Hat Enterprise Linux die Datei `/etc/redhat-release` scannen.

```
hostnamectl
```

```
"packages": {
    "platform": {
      "platform-version": {
        "architecture": {
          "file": ".zip-file-name-1.zip"
        }
      }
    },
    "another-platform": {
      "platform-version": {
        "architecture": {
          "file": ".zip-file-name-2.zip"
        }
      }
    },
    "another-platform": {
      "platform-version": {
        "architecture": {
          "file": ".zip-file-name-3.zip"
        }
      }
    }
  }
```

Im Folgenden wird ein -Beispiel gezeigt. In diesem Beispiel ist die Betriebssystemplattform `amazon`, die unterstützte Version `2016.09`, die Architektur `x86_64` und die ZIP-Datei, die diese Plattform unterstützt, `test.zip`.

```
{
    "amazon": {
        "2016.09": {
            "x86_64": {
                "file": "test.zip"
            }
        }
    }
},
```

Sie können mit dem Platzhalterwert (`_any`) angeben, dass das Paket alle Versionen des übergeordneten Elements unterstützt. Um beispielsweise anzugeben, dass das Paket in jeder Version von Amazon Linuxs unterstützt wird, sollte Ihre Paketanweisung ähnlich wie folgt aussehen. Sie können den Platzhalter `_any` auf Versions- oder Architekturebene verwenden, um alle Versionen einer Plattform, alle Architekturen in einer Version oder alle Versionen und alle Architekturen einer Plattform zu unterstützen.

```
{
    "amazon": {
        "_any": {
            "x86_64": {
                "file": "test.zip"
            }
        }
    }
},
```

Das folgende Beispiel fügt `_any` hinzu, um zu zeigen, dass das erste Paket `data1.zip` für alle Architekturen von Amazon Linux 2016.09 unterstützt wird. Das zweite Paket (`data2.zip`) wird für alle Versionen von Amazon Linux unterstützt, jedoch nur für verwaltete Knoten mit `x86_64`-Architektur. Sowohl die Version mit `2023.8` als auch die Version mit `_any` sind Einträge unter `amazon`. Es handelt sich um eine einzige Plattform (Amazon Linux), aber verschiedene unterstützte Versionen und Architekturen sowie zugehörige ZIP-Dateien.

```
{
    "amazon": {
        "2023.8": {
            "_any": {
                "file": "data1.zip"
            }
        },
        "_any": {
            "x86_64": {
                "file": "data2.zip"
            }
        }
    }
}
```

Sie können im Abschnitt `"packages"` des Manifests mehrmals auf eine ZIP-Datei verweisen, wenn diese mehrere Plattformen unterstützt. Wenn Sie beispielsweise eine ZIP-Datei haben, die sowohl Red Hat Enterprise Linux 8.x-Versionen als auch Amazon Linux unterstützt, haben Sie zwei Einträge in dem `"packages"` Abschnitt, die auf dieselbe .zip-Datei verweisen, wie im folgenden Beispiel gezeigt.

```
{
    "amazon": {
        "2023.8.20250715 ": {
            "x86_64": {
                "file": "test.zip"
            }
        }
    },
    "redhat": {
        "8.*": {
            "x86_64": {
                "file": "test.zip"
            }
        }
    }
},
```

5. Fügen Sie die Liste mit den zu diesem Paket gehörenden ZIP-Dateien aus Schritt 4 hinzu. Für jeden Dateieintrag sind der Dateiname und die Prüfsumme des `sha256`-Hash-Werts erforderlich. Die Prüfsummenwerte im Manifest müssen mit dem `sha256`-Hash-Wert in den gezippten Ressourcen übereinstimmen, um zu verhindern, dass die Paketinstallation fehlschlägt.

   Um die korrekte Prüfsumme aus den zu installierenden Dateien zu erhalten, können Sie die folgenden Befehle ausführen. In Linux führen Sie `shasum -a 256` *`file-name`*`.zip` oder `openssl dgst -sha256` *`file-name`*`.zip` aus. Führen Sie unter Windows das Cmdlet in aus. `Get-FileHash -Path` *`path-to-.zip-file`* PowerShell

   Der Abschnitt `"files"` des Manifests enthält einen Verweis auf jede ZIP-Datei in Ihrem Paket.

```
"files": {
        "test-agent-x86.deb.zip": {
            "checksums": {
                "sha256":
 "EXAMPLE2706223c7616ca9fb28863a233b38e5a23a8c326bb4ae241dcEXAMPLE"
```

```
            }
        },
        "test-agent-x86_64.deb.zip": {
            "checksums": {
                "sha256":
"EXAMPLE572a745844618c491045f25ee6aae8a66307ea9bff0e9d1052EXAMPLE"
            }
        },
        "test-agent-x86_64.nano.zip": {
            "checksums": {
                "sha256":
"EXAMPLE63ccb86e830b63dfef46995af6b32b3c52ce72241b5e80c995EXAMPLE"
            }
        },
        "test-agent-rhel8-x86.nano.zip": {
            "checksums": {
                "sha256":
"EXAMPLE13df60aa3219bf117638167e5bae0a55467e947a363fff0a51EXAMPLE"
            }
        },
        "test-agent-x86.msi.zip": {
            "checksums": {
                "sha256":
"EXAMPLE12a4abb10315aa6b8a7384cc9b5ca8ad8e9ced8ef1bf0e5478EXAMPLE"
            }
        },
        "test-agent-x86_64.msi.zip": {
            "checksums": {
                "sha256":
"EXAMPLE63ccb86e830b63dfef46995af6b32b3c52ce72241b5e80c995EXAMPLE"
            }
        },
        "test-agent-rhel8-x86.rpm.zip": {
            "checksums": {
                "sha256":
"EXAMPLE13df60aa3219bf117638167e5bae0a55467e947a363fff0a51EXAMPLE"
            }
        }
    }
```

6. Nachdem Sie die Paketinformationen hinzugefügt haben, speichern und schließen Sie die Manifestdatei.

Im Folgenden finden Sie ein Beispiel für ein fertiges Manifest. In diesem Beispiel haben Sie eine ZIP-Datei `NewPackage_LINUX.zip`, die mehrere Plattformen unterstützt, jedoch nur einmal im Abschnitt `"files"` referenziert wird.

```
{
    "schemaVersion": "2.0",
    "version": "1.7.1",
    "publisher": "Amazon Web Services",
    "packages": {
        "windows": {
            "_any": {
                "x86_64": {
                    "file": "NewPackage_WINDOWS.zip"
                }
            }
        },
        "amazon": {
            "_any": {
                "x86_64": {
                    "file": "NewPackage_LINUX.zip"
                }
            }
        },
        "ubuntu": {
            "_any": {
                "x86_64": {
                    "file": "NewPackage_LINUX.zip"
                }
            }
        }
    },
    "files": {
        "NewPackage_WINDOWS.zip": {
            "checksums": {
                "sha256":
 "EXAMPLEc2c706013cf8c68163459678f7f6daa9489cd3f91d52799331EXAMPLE"
            }
        },
        "NewPackage_LINUX.zip": {
            "checksums": {
                "sha256":
 "EXAMPLE2b8b9ed71e86f39f5946e837df0d38aacdd38955b4b18ffa6fEXAMPLE"
            }
```

```
        }
    }
}
```

Beispiel für ein Paket

Ein Beispielpaket, ExamplePackage.zip, steht Ihnen auf unserer Website zum Herunterladen zur Verfügung. Das Beispielpaket enthält ein fertiges JSON-Manifest und drei ZIP-Dateien.

Schritt 3: Hochladen von Paket und Manifest zu einem Amazon S3-Bucket

Bereiten Sie Ihr Paket vor, indem Sie alle ZIP-Dateien in einen Ordner oder ein Verzeichnis kopieren oder verschieben. Ein Paket ist nur gültig, wenn es das von Ihnen in Schritt 2: Erstellen des JSON-Paketmanifests erstellte Manifest und alle ZIP-Dateien enthält, die in der Dateiliste des Manifests angegeben sind.

So laden Sie das Paket und das Manifest in Amazon S3 hoch

1. Kopieren oder verschieben Sie alle von Ihnen im Manifest angegebenen ZIP-Archivdateien in einen Ordner oder ein Verzeichnis. Komprimieren Sie nicht den Ordner oder das Verzeichnis, in das/den Sie die ZIP-Archivdateien und die Manifestdatei verschieben.
2. Erstellen Sie einen Bucket, oder wählen Sie einen vorhandenen Bucket aus. Weitere Informationen finden Sie unter Create a Bucket (Bucket erstellen) im Amazon Simple Storage Service Getting Started Guide (Amazon Simple Storage Service Erste-Schritte-Leitfaden). Weitere Informationen darüber, wie Sie einen AWS CLI Befehl ausführen, um einen Bucket zu erstellen, finden Sie mbin der AWS CLI Befehlsreferenz.
3. Laden Sie den Ordner oder das Verzeichnis zum Bucket hoch. Anleitungen finden Sie unter Hinzufügen eines Objekts zu einem Bucket im Erste Schritte-Handbuch zu Amazon Simple Storage Service. Wenn Sie Ihr JSON-Manifest in die AWS Systems Manager Konsole einfügen möchten, laden Sie das Manifest nicht hoch. Weitere Informationen zum Ausführen eines AWS CLI Befehls zum Hochladen von Dateien in einen Bucket finden Sie mvin der AWS CLI Befehlsreferenz.
4. Wählen Sie auf der Startseite des Buckets den von Ihnen hochgeladenen Ordner oder das Verzeichnis aus. Wenn Sie Ihre Dateien in einen Unterordner in einem Bucket hochgeladen haben, stellen Sie sicher, dass Sie sich den Namen des Unterordner notieren (wird auch als Präfix bezeichnet). Sie benötigen das Präfix, um Ihr Paket zu Distributor hinzuzufügen.

Schritt 4: Hinzufügen eines Pakets zu Distributor

Sie können die AWS Systems Manager Konsole, die AWS Befehlszeilentools (AWS CLI und AWS Tools für PowerShell) verwenden oder AWS SDKs ein neues Paket hinzufügenDistributor. Wenn Sie ein Paket hinzufügen, fügen Sie ein neues SSM-Dokument hinzu. Mit dem Dokument können Sie das Paket an verwaltete Knoten bereitstellen.

Themen

- Fügen Sie ein Paket mit der Konsole hinzu
- Fügen Sie ein Paket hinzu, indem Sie AWS CLI

Fügen Sie ein Paket mit der Konsole hinzu

Sie können die AWS Systems Manager Konsole verwenden, um ein Paket zu erstellen. Halten Sie den Namen des Buckets bereit, auf den Sie in Ihr Paket in Schritt 3: Hochladen von Paket und Manifest zu einem Amazon S3-Bucket hochgeladen haben.

So fügen Sie Distributor ein Paket hinzu (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählenen Sie auf der Distributor-Startseite die Option Create package (Paket erstellen) und klicken Sie dann auf Advanced (Erweitert).
4. Geben Sie auf der Seite Create package (Paket erstellen) einen Namen für Ihr Paket ein. Paketnamen können Buchstaben, Zahlen, Punkte, Bindestriche und Unterstriche enthalten. Der Name sollte allgemein genug sein, um auf alle Versionen der Paketanhänge angewendet werden zu können, jedoch spezifisch genug, um den Zweck des Pakets zu identifizieren.
5. Geben Sie unter Version name (Versionsname) den exakten Wert des Eintrags `version` in Ihrer Manifestdatei ein.
6. Wählen Sie unter S3 bucket name (S3-Bucketname) den Namen des Buckets aus, in den Sie Ihre ZIP-Dateien und das Manifest in the section called “Schritt 3: Hochladen von Paket und Manifest zu einem Amazon S3-Bucket” hochgeladen haben.
7. Geben Sie unter S3 key prefix (S3-Schlüsselpräfix) den Unterordner des Buckets ein, in dem Ihre ZIP-Dateien und das Manifest gespeichert sind.

8. Wählen Sie unter Manifest die Option Extract from package (Aus Paket extrahieren) aus, um ein Manifest zu verwenden, das Sie mit Ihren ZIP-Dateien in den Amazon S3-Bucket hochgeladen haben.

   (Optional) Wenn Sie Ihr JSON-Manifest nicht in den S3-Bucket hochgeladen haben, in dem Ihre ZIP-Dateien gespeichert sind, wählen Sie New Manifest (Neues Manifest) aus. Sie können das gesamte Manifest in dem JSON-Editor erstellen oder in ihn hineinkopieren. Weitere Informationen zum Erstellen des JSON-Manifests finden Sie unter Schritt 2: Erstellen des JSON-Paketmanifests.
9. Wenn das Manifest fertiggestellt ist, wählen Sie Create package (Paket erstellen).
10. Warten Sie, bis Distributor die Erstellung des Pakets aus den ZIP-Dateien und dem Manifest abgeschlossen hat. Abhängig von der Anzahl und Größe der Pakete, die Sie hinzufügen, kann dies einige Minuten dauern. Distributor leitet Sie automatisch auf die Seite Package details (Paketdetails) für das neue Paket weiter, Sie können diese Seite aber jederzeit selbst öffnen, nachdem die Software hochgeladen ist. Auf der Seite Package details (Paketdetails) werden erst dann alle Informationen zu Ihrem Paket angezeigt, wenn Distributor den Paketerstellungsprozess abgeschlossen hat. Um den Upload- und Paketerstellungsprozess abzubrechen, wählen Sie Cancel (Abbrechen).

Fügen Sie ein Paket hinzu, indem Sie AWS CLI

Sie können den verwenden AWS CLI , um ein Paket zu erstellen. Halten Sie die URL für den Bucket bereit, zu dem Sie Ihr Paket in Schritt 3: Hochladen von Paket und Manifest zu einem Amazon S3-Bucket hochgeladen haben.

Um ein Paket zu Amazon S3 hinzuzufügen, verwenden Sie AWS CLI

1. Um das zum Erstellen eines Pakets AWS CLI zu verwenden, führen Sie den folgenden Befehl aus und *`package-name`* ersetzen Sie ihn durch den Namen Ihres Pakets und *`path-to-manifest-file`* durch den Dateipfad für Ihre JSON-Manifestdatei. amzn-s3-demo-bucket ist die URL des Amazon S3-Buckets, in dem das gesamte Paket gespeichert ist. Wenn Sie den Befehl create-document in Distributor ausführen, geben Sie den `Package`-Wert für `--document-type` an.

   Wenn Sie Ihre Manifestdatei nicht dem Amazon S3-Bucket hinzugefügt haben, ist der `--content`-Parameterwert der Dateipfad zur JSON-Manifestdatei.

   ```
   aws ssm create-document \
   ```

```
    --name "package-name" \
    --content file://path-to-manifest-file \
    --attachments Key="SourceUrl",Values="amzn-s3-demo-bucket" \
    --version-name version-value-from-manifest \
    --document-type Package
```

Im Folgenden wird ein -Beispiel gezeigt.

```
aws ssm create-document \
    --name "ExamplePackage" \
    --content file://path-to-manifest-file \
    --attachments Key="SourceUrl",Values="https://s3.amazonaws.com/amzn-s3-demo-
bucket/ExamplePackage" \
    --version-name 1.0.1 \
    --document-type Package
```

2. Vergewissern Sie sich, dass Ihr Paket hinzugefügt wurde, und zeigen Sie das Paketmanifest an, indem Sie den folgenden Befehl ausführen und ihn durch den Namen Ihres Pakets ersetzen*package-name*. Um eine spezifische Version des Dokuments (nicht identisch mit der Version eines Pakets) zu erhalten, können Sie den Parameter `--document-version` hinzufügen.

```
aws ssm get-document \
    --name "package-name"
```

Informationen zu anderen Optionen, die Sie mit dem Befehl create-document verwenden können, finden Sie unter create-document im Abschnitt AWS Systems Manager der AWS CLI -Command Reference. Informationen zu anderen Optionen, die Sie mit dem Befehl get-document verwenden können, finden Sie unter get-document.

Bearbeiten von Distributor-Paketberechtigungen in der Konsole

Nachdem Sie Distributor, einem Tool in AWS Systems Manager, ein Paket hinzugefügt haben, können Sie die Berechtigungen des Pakets in der Systems-Manager-Konsole bearbeiten. Sie können den Berechtigungen eines Pakets weitere AWS-Konten hinzufügen. Pakete können nur für andere Konten in derselben AWS-Region freigegeben werden. Die Freigabe über Regionsgrenzen hinweg wird nicht unterstützt. Standardmäßig sind Pakete auf Private (Privat) festgelegt. Das bedeutet, dass nur Benutzer mit Zugriff auf das AWS-Konto des Paketerstellers Paketinformationen anzeigen und

das Paket aktualisieren oder löschen können. Wenn Private (Privat)-Berechtigungen akzeptabel sind, können Sie dieses Verfahren überspringen.

> **Note**
>
> Sie können die Berechtigungen von Paketen aktualisieren, die mit 20 oder weniger Konten freigegeben werden.

Bearbeiten von Paketberechtigungen in der Konsole

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Seite Packages (Pakete) das Paket aus, für das Sie Berechtigungen bearbeiten möchten.
4. Wählen Sie auf der Registerkarte Package details (Paketdetails) die Option Edit permissions (Berechtigungen bearbeiten) aus, um Berechtigungen zu ändern.
5. Wählen Sie unter Edit permissions (Berechtigungen bearbeiten) die Option Shared with specific accounts (Für bestimmte Konten freigegeben) aus.
6. Fügen Sie in Shared with specific accounts (Für bestimmte Konten freigegeben) nacheinander AWS-Konto hinzu. Wenn Sie fertig sind, wählen Sie Speichern.

Bearbeiten von Distributor-Paket-Tags in der Konsole

Nachdem Sie Distributor, einem Tool in AWS Systems Manager, ein Paket hinzugefügt haben, können Sie die Tags des Pakets in der Systems-Manager-Konsole bearbeiten. Diese Tags werden auf das Paket angewendet. Sie haben keine Verbindung zu Tags in dem verwalteten Knoten, auf dem Sie das Paket bereitstellen möchten. Tags unterscheiden nach Groß- und Kleinschreibung. Es handelt sich um Schlüssel-Wert-Paare, die Ihnen helfen können, Ihre Pakete nach für Ihre Organisation relevanten Kriterien zu gruppieren und zu filtern. Wenn Sie keine Tags hinzufügen möchten, können Sie Ihr Paket installieren oder eine neue Version hinzufügen.

So bearbeiten Sie Paket-Tags in der Konsole

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Seite Packages (Pakete) das Paket aus, für das Sie Tags bearbeiten möchten.
4. Wählen Sie auf der Registerkarte Package details (Paketdetails) in Tags (Tags) die Option Edit (Bearbeiten) aus.
5. Geben Sie unter Add tags (Tags hinzufügen) einen Schlüssel oder ein Schlüssel-Wert-Paar für das Tag ein. Klicken Sie anschließend auf Add (Hinzufügen). Wiederholen Sie dies, wenn Sie weitere Tags hinzufügen möchten. Um Tags zu löschen, wählen Sie unten im Fenster für das Tag X aus.
6. Wenn Sie Ihrem Paket keine weiteren Tags mehr hinzufügen möchten, wählen Sie Save (Speichern) aus.

Hinzufügen einer Version zu einem Distributor-Paket

Um eine Paketversion hinzuzufügen, erstellen Sie ein Paket und verwenden Sie es dann, Distributor um eine Paketversion hinzuzufügen, indem Sie dem AWS Systems Manager (SSM-) Dokument, das bereits für ältere Versionen vorhanden ist, einen Eintrag hinzufügen. Distributorist ein Tool in AWS Systems Manager. Um Zeit zu sparen, aktualisieren Sie das Manifest für eine ältere Version des Pakets, ändern den Wert des Eintrags `version` im Manifest (z. B. von `Test_1.0` in `Test_2.0`) und speichern das Manifest als Manifest für die neue Version. Bei dem einfachen Add Version (Version hinzufügen)-Workflow in der Distributor-Konsole wird das Manifest automatisch aktualisiert.

Eine neue Paketversion kann:

- Mindestens eine der installierbaren Dateien ersetzen, die der aktuellen Version angefügt sind.
- Neue installierbare Dateien hinzufügen, um zusätzliche Plattformen zu unterstützen
- Dateien löschen, um die Unterstützung für bestimmte Plattformen zu beenden

Eine neuere Version kann denselben Amazon Simple Storage Service (Amazon S3)-Bucket verwenden, muss jedoch eine URL mit einem anderen Dateinamen am Ende besitzen. Sie können die Systems Manager-Konsole oder das AWS Command Line Interface (AWS CLI) verwenden, um die neue Version hinzuzufügen. Beim Hochladen einer installierbaren Datei mit demselben Namen wie eine vorhandene installierbare Datei in dem Amazon S3-Bucket wird die vorhandene Datei überschrieben. Es werden keine Dateien aus der älteren Version in die neue Version hineinkopiert. Sie müssen installierbare Dateien aus der älteren Version erneut hochladen, damit sie in die neue Version aufgenommen werden. Nachdem Distributor Ihre neue Paketversion erstellt hat, können Sie

den Amazon S3-Bucket löschen oder wiederverwenden, da Distributor Ihre Software im Rahmen der Versioning in einen internen Systems Manager-Bucket kopiert.

> **Note**
>
> Jedes Paket ist auf maximal 25 Versionen beschränkt. Sie können Versionen löschen, die nicht mehr benötigt werden.

Themen

- Hinzufügen einer Paketversion mit der Konsole
- Hinzufügen einer Paketversion mit dem AWS CLI

Hinzufügen einer Paketversion mit der Konsole

Führen Sie vor der Ausführung der folgenden Schritte die Anweisungen unter Erstellen eines Pakets in Distributor aus, um ein neues Paket für die Version zu erstellen. Verwenden Sie anschließend die Systems Manager-Konsole, um Distributor eine neue Paketversion hinzuzufügen.

Hinzufügen einer Paketversion mithilfe des einfachen Workflows

Um eine Paketversion mithilfe des einfachen Workflows hinzuzufügen, bereiten Sie aktualisierte installierbare Dateien vor oder fügen Sie installierbare Dateien hinzu, um weitere Plattformen und Architekturen zu unterstützen. Verwenden Sie anschließend Distributor zum Hochladen neuer und aktualisierter installierbarer Dateien und fügen Sie eine Paketversion hinzu. Die vereinfachte Add version (Version hinzufügen)-Workflow in der Distributor Konsole aktualisiert die Manifestdatei und das zugehörige SSM-Dokument automatisch.

Hinzufügen einer Paketversion mithilfe des einfachen Workflows

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Distributor-Startseite das Paket aus, dem Sie eine weitere Version hinzufügen möchten.
4. Wählen Sie auf der Seite Add version (Version hinzufügen) die Option Simple (Einfach).

5. Geben Sie unter Version name (Versionsname) einen Versionsnamen ein. Der Versionsname für die neue Version muss sich von der älteren Version unterscheiden. Versionsnamen dürfen maximal 512 Zeichen lang sein und dürfen keine Sonderzeichen enthalten.
6. Wählen Sie für S3 bucket name (S3-Bucketname), einen vorhandenen S3-Bucket aus der Liste aus. Dabei kann es sich um den Bucket handeln, den Sie zum Speichern installierbarer Dateien für ältere Versionen verwendet haben, aber die installierbaren Dateinamen müssen unterschiedlich sein, damit das Überschreiben vorhandener installierbaren Dateien in dem Bucket vermieden wird.
7. Geben Sie unter S3 key prefix (S3-Schlüsselpräfix) den Unterordner des Buckets ein, in dem Ihre installierbaren Komponenten gespeichert sind.
8. Navigieren Sie unter Upload Software (Software hochladen) zu den installierbaren Softwaredateien, die Sie für die neue Version anfügen möchten. Installierbare Versionen von vorhandenen Dateien werden nicht automatisch in eine neue Version herüberkopiert. Sie müssen alle installierbaren Dateien aus älteren Versionen des Pakets erneut hochladen, wenn Sie diese in die neue Version übernehmen möchten. Sie können mehrere Softwaredateien in einer einzigen Aktion hochladen.
9. Überprüfen Sie unter Für Target platform (Ziel-Plattform für jede Plattform, ob das Ziel-Betriebssystem für die installierbare Datei korrekt ist. Wenn das angezeigte Betriebssystem nicht korrekt ist, wählen Sie das richtige Betriebssystem aus der Dropdown-Liste aus.

   Bei dem Workflow Simple (Einfach) zur Versioning sind zusätzliche Schritte erforderlich, wenn nur eine Datei für mehrere Betriebssysteme als Ziel verwendet werden soll, da installierbare Dateien nur einmal hochgeladen werden. Wenn Sie zum Beispiel eine installierbare Softwaredatei mit dem Namen `Logtool_v1.1.1.rpm` hochladen, müssen Sie einige Standardeinstellungen für den Einfach-Workflow ändern, um Distributor anzuweisen, für dieselbe Software sowohl das Amazon-Linux- als auch das Ubuntu Server-Betriebssystem als Ziel festzulegen. Um dieses Problem zu beheben, können Sie eine der folgenden Aktionen ausführen.

   - Verwenden Sie stattdessen den Workflow Advanced (Erweitert) zur Versioning, zippen Sie jede installierbare Datei, bevor Sie beginnen, und richten Sie das Manifest manuell so ein, dass eine installierbare Datei für mehrere Betriebssystemplattformen oder -versionen verwendet werden kann. Weitere Informationen finden Sie unter Hinzufügen einer Paketversion mithilfe des erweiterten Workflows.
   - Bearbeiten Sie die Manifestdatei im Workflow Simple (Einfach) so, dass Ihre ZIP-Datei für mehrere Betriebssystemplattformen oder -versionen verwendet wird. Weitere Informationen

zu diesem Verfahren finden Sie am Ende von Schritt 4 in Schritt 2: Erstellen des JSON-Paketmanifests.

10. Vergewissern Sie sich, dass es sich bei der angezeigten Plattformversion des Betriebssystems entweder **_any** um eine Hauptversion mit einem Platzhalter (8.*) oder um die genaue Betriebssystemversion handelt, für die Ihre Software gelten soll. Weitere Informationen zum Festlegen einer Plattformversion finden Sie unter Schritt 4 in Schritt 2: Erstellen des JSON-Paketmanifests.
11. Wählen Sie unter Architecture (Architektur) für jeden installierbare Datei die richtige Prozessorarchitektur aus der Dropdown-Liste aus. Weitere Informationen zu unterstützten Architekturen finden Sie unter Unterstützte Paketplattformen und -architekturen.
12. (Optional) Erweitern Sie Scripts (Skripts) und überprüfen Sie die Installations- und Deinstallationsskripts, die Distributor für Ihre installierbare Software generiert hat.
13. Zum Hinzufügen weiterer installierbarer Softwaredateien zu der neuen Version wählen Sie Add Software (Software hinzufügen). Andernfalls fahren Sie mit dem nächsten Schritt fort.
14. (Optional) Erweitern Sie Manifest und überprüfen Sie das JSON-Manifest des Pakets, das Distributor für Ihre installierbare Software generiert hat. Wenn Sie Informationen über Ihre installierbare Software geändert haben, nachdem Sie mit dieser Prozedur begonnen haben, beispielsweise die Plattformversion oder die Zielplattform, wählen Sie Generate Manifest (Manifest erzeugen), um das Paketmanifest zu aktualisieren.

    Sie können das Manifest manuell bearbeiten, wenn Sie möchten, dass für eine installierbare Software mehr als ein Betriebssystem als Ziel festgelegt wird, wie in Schritt 9 beschrieben. Weitere Informationen zum Bearbeiten des Manifests finden Sie unter Schritt 2: Erstellen des JSON-Paketmanifests.
15. Wählen Sie nach dem Hinzufügen der Software und der Überprüfung der Daten zur Zielplattform, zur Version und zur Architektur Sie Add version (Version hinzufügen).
16. Warten Sie, bis Distributor das Hochladen Ihrer Software und das Erstellen Ihres neuen Pakets abgeschlossen Hat. Distributor zeigt den Uploadstatus für jede installierbare Datei einzeln an. Abhängig von der Anzahl und Größe der Pakete, die Sie hinzufügen, kann dies einige Minuten dauern. Distributor leitet Sie automatisch auf die Seite Package details (Paketdetails) für das neue Paket weiter, Sie können diese Seite aber jederzeit selbst öffnen, nachdem die Software hochgeladen ist. Auf der Seite Package details (Paketdetails) werden erst dann alle Informationen zu Ihrem Paket angezeigt, wenn Distributor den Erstellungsprozess für die neue Paketversion abgeschlossen hat. Um den Uploadvorgang bzw. den prozess zur Erstellung der Paketversion anzuhalten, wählen Sie Stop upload (Upload anhalten).

17. Wenn Distributor keine installierbaren Softwaredateien hochladen kann, wird eine Fehlermeldung Upload failed (Upload fehlgeschlagen) angezeigt. Um den Uploadversuch zu wiederholen, wählen Sie Retry Upload (Uploadversuch wiederholen). Weitere Informationen zur Fehlerbehebung bei der Paketversionserstellung finden Sie unter Fehlerbehebung für AWS Systems ManagerDistributor.
18. Wenn Distributor die neue Paketversion erstellt hat. zeigen Sie auf der Seite Details (Details) auf der Registerkarte Versions (Versionen) die neue Version in der Liste der verfügbaren Paketversionen an. Legen Sie die Standardversion des Pakets fest, indem Sie eine Version auswählen. Wählen Sie anschließend Set default version (Als Standardversion festlegen) aus.

    Wenn Sie keine Standardversion festlegen, ist die neueste Paketversion die Standardversion.

Hinzufügen einer Paketversion mithilfe des erweiterten Workflows

Um eine Paketversion hinzuzufügen, erstellen Sie ein Paket und verwenden Sie anschließend Distributor, um eine Paketversion hochzuladen, indem Sie dem für ältere Versionen vorhandenen SSM-Dokument einen Eintrag hinzufügen. Um Zeit zu sparen, aktualisieren Sie das Manifest für eine ältere Version des Pakets, ändern den Wert des Eintrags `version` im Manifest (z. B. von `Test_1.0` in `Test_2.0`) und speichern das Manifest als Manifest für die neue Version. Sie müssen das Manifest so ändern, dass eine neue Version des Pakets hinzugefügt wird. Dies führen Sie mit dem Advanced-Workflow durch.

Hinzufügen einer Paketversion mithilfe des erweiterten Workflows

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Distributor-Startseite das Paket aus, dem Sie eine weitere Version hinzufügen möchten, an wählen Sie dann Add version (Version hinzufügen).
4. Geben Sie unter Version name (Versionsname) den exakten Wert des Eintrags `version` Ihrer Manifestdatei ein.
5. Wählen Sie für S3 bucket name (S3-Bucketname), einen vorhandenen S3-Bucket aus der Liste aus. Dabei kann es sich um den Bucket handeln, den Sie zum Speichern installierbarer Dateien für ältere Versionen verwendet haben, aber die installierbaren Dateinamen müssen unterschiedlich sein, damit das Überschreiben vorhandener installierbaren Dateien in dem Bucket vermieden wird.

6. Geben Sie unter S3 key prefix (S3-Schlüsselpräfix) den Unterordner des Buckets ein, in dem Ihre installierbaren Komponenten gespeichert sind.
7. Wählen Sie unter Manifest die Option Extract from package (Aus Paket extrahieren) aus, um ein Manifest zu verwenden, das Sie mit Ihren ZIP-Dateien in den S3-Bucket hochgeladen haben.

   (Optional) Wenn Sie kein aktualisiertes JSON-Manifest in den Amazon S3-Bucket mit Ihren ZIP-Dateien hochgeladen haben, wählen Sie New manifest (Neues Manifest) aus. Sie können das gesamte Manifest in dem JSON-Editor erstellen oder in ihn hineinkopieren. Weitere Informationen zum Erstellen des JSON-Manifests finden Sie unter Schritt 2: Erstellen des JSON-Paketmanifests.
8. Wenn das Manifest fertiggestellt ist, wählen Sie Add package version (Paketversion hinzufügen).
9. Zeigen Sie auf der Seite Details (Details) auf der Registerkarte Versions (Versionen) die neue Version in der Liste der verfügbaren Paketversionen an. Legen Sie die Standardversion des Pakets fest, indem Sie eine Version auswählen. Wählen Sie anschließend Set default version (Als Standardversion festlegen) aus.

   Wenn Sie keine Standardversion festlegen, ist die neueste Paketversion die Standardversion.

Hinzufügen einer Paketversion mit dem AWS CLI

Sie können das verwenden AWS CLI , um eine neue Paketversion hinzuzufügenDistributor. Vor der Ausführung der folgenden Befehle müssen Sie eine neue Paketversion erstellen und diese zu S3 hochladen wie am Anfang dieses Themas beschrieben.

Um eine Paketversion hinzuzufügen, verwenden Sie AWS CLI

1. Führen Sie den folgenden Befehl aus, um das AWS Systems Manager Dokument mit einem Eintrag für eine neue Paketversion zu bearbeiten. *`document-name`*Ersetzen Sie es durch den Namen Ihres Dokuments. *`amzn-s3-demo-bucket`*Ersetzen Sie es durch die URL des JSON-Manifests, das Sie kopiert habenSchritt 3: Hochladen von Paket und Manifest zu einem Amazon S3-Bucket. *`S3-bucket-URL-of-package`*ist die URL des Amazon S3 S3-Buckets, in dem das gesamte Paket gespeichert ist. *`version-name-from-updated-manifest`*Ersetzen Sie es durch den Wert von `version` im Manifest. Legen Sie den Parameter `--document-version` auf `$LATEST` fest, um das Dokument für diese Paketversion als aktuelle Version des Dokuments festzulegen.

   ```
   aws ssm update-document \
       --name "document-name" \
   ```

```
    --content "S3-bucket-URL-to-manifest-file" \
    --attachments Key="SourceUrl",Values="amzn-s3-demo-bucket" \
    --version-name version-name-from-updated-manifest \
    --document-version $LATEST
```

Im Folgenden wird ein -Beispiel gezeigt.

```
aws ssm update-document \
    --name ExamplePackage \
    --content "https://s3.amazonaws.com/amzn-s3-demo-bucket/ExamplePackage/
manifest.json" \
    --attachments Key="SourceUrl",Values="https://s3.amazonaws.com/amzn-s3-demo-
bucket/ExamplePackage" \
    --version-name 1.1.1 \
    --document-version $LATEST
```

2. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob Ihr Paket aktualisiert wurde, und das Paketmanifest anzuzeigen. *package-name*Ersetzen Sie es durch den Namen Ihres Pakets und optional durch die Versionsnummer des Dokuments (nicht identisch *document-version* mit der Paketversion), das Sie aktualisiert haben. Wenn diese Paketversion der aktuellen Version des Dokuments zugeordnet ist, können Sie `$LATEST` als Wert des optionalen Parameters `--document-version` angeben.

```
aws ssm get-document \
    --name "package-name" \
    --document-version "document-version"
```

Informationen zu anderen Optionen, die Sie mit dem update-document Befehl verwenden können, finden Sie update-documentim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Installieren oder Aktualisieren von Distributor-Paketen

Sie können Pakete auf Ihren AWS Systems Manager verwalteten Knoten bereitstellenDistributor, indem Sie ein Tool in verwenden AWS Systems Manager. Um die Pakete bereitzustellen, verwenden Sie entweder AWS-Managementkonsole oder AWS Command Line Interface (AWS CLI). Sie können pro Befehl eine Version eines Pakets bereitstellen. Sie können neue Pakete installieren oder vorhandene Installationen direkt aktualisieren. Sie können wählen, ob Sie eine bestimmte Version oder stets die aktuelle Version eines Pakets bereitstellen möchten. Wir empfehlenState Manager, zur Installation von Paketen ein Tool in AWS Systems Manager zu verwenden. Durch die Verwendung

wird State Manager sichergestellt, dass auf Ihren verwalteten Knoten immer die neueste up-to-date Version Ihres Pakets ausgeführt wird.

> **Important**
>
> Mit Distributor installierte Pakete sollten nur mithilfe von Distributor deinstalliert werden. Andernfalls kann Systems Manager die Anwendung weiterhin als `INSTALLED` erkennen und es können andere unbeabsichtigte Ergebnisse auftreten.

Präferenz	AWS Systems Manager Aktion	Weitere Informationen
Installieren oder aktualisieren Sie ein Paket sofort.	Run Command	• Einmaliges Installieren oder Aktualisieren eines Pakets mithilfe der Konsole • Einmaliges Installieren eines Pakets mit dem AWS CLI • Einmaliges Aktualisieren eines Pakets mit dem AWS CLI
Installieren Sie ein Paket nach einem Zeitplan, sodass die Installation immer die Standardversion enthält.	State Manager	• Planen einer Paketinstallation oder -aktualisierung mithilfe der Konsole • Planung einer Paketinstallation mit dem AWS CLI • Planung einer Paket-Aktualisierung mit dem AWS CLI
Installieren Sie ein Paket automatisch auf neuen verwalteten Knoten, die ein bestimmtes Tag oder einen bestimmten Satz von Tags besitzen. Zum Beispiel die Installation des CloudWatch	State Manager	Eine Möglichkeit besteht in der Anwendung von Tags auf neue verwaltete Knoten und die anschließende Auflistung der Tags als Ziele in der State Manager-Zuordnung. State Manager installiert

Präferenz	AWS Systems Manager Aktion	Weitere Informationen
Amazon-Agenten auf neuen Instances.		automatisch das Paket in einer Zuordnung auf verwalteten Knoten, deren Tags übereinst immen. Siehe Verstehen von Zielen und Ratensteuerungen in State Manager Zuordnung en.

Themen

- Einmaliges Installieren oder Aktualisieren eines Pakets mithilfe der Konsole
- Planen einer Paketinstallation oder -aktualisierung mithilfe der Konsole
- Einmaliges Installieren eines Pakets mit dem AWS CLI
- Einmaliges Aktualisieren eines Pakets mit dem AWS CLI
- Planung einer Paketinstallation mit dem AWS CLI
- Planung einer Paket-Aktualisierung mit dem AWS CLI

Einmaliges Installieren oder Aktualisieren eines Pakets mithilfe der Konsole

Sie können die AWS Systems Manager Konsole verwenden, um ein Paket einmal zu installieren oder zu aktualisieren. Wenn Sie eine einmalige Installation konfigurierenAWS Systems Manager Run Command, Distributor verwendet ein Tool in AWS Systems Manager, um die Installation durchzuführen.

So installieren oder aktualisieren Sie ein Paket einmalig mithilfe der Konsole

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Distributor-Startseite das Paket aus, das Sie installieren möchten.
4. Wählen Sie Install one time (Einmal installieren) aus.

   Dieser Befehl öffnet Run Command mit dem Befehlsdokument `AWS-ConfigureAWSPackage`. Ihr Distributor-Paket ist vorausgewählt.

5. Wählen Sie unter Document version (Dokumentversion) die Version des `AWS-ConfigureAWSPackage`-Dokuments aus, das Sie ausführen möchten.
6. Wählen Sie für Action (Aktion) die Option Install (Installieren).
7. Wählen Sie unter Installation type (Installationstyp) eine der folgenden Optionen aus:

   - Uninstall and reinstall (Deinstallieren und neu installieren): Das Paket wird vollständig deinstalliert und dann neu installiert. Die Anwendung ist bis zum Abschluss der Neuinstallation nicht verfügbar.
   - In-place update (Direkte Aktualisierung): Der vorhandenen Installation werden entsprechend den Anweisungen, die Sie in einem `update`-Skript angeben, nur neue oder geänderte Dateien hinzugefügt. Die Anwendung ist während des Aktualisierungsprozesses weiterhin verfügbar. Diese Option wird für AWS veröffentlichte Pakete außer dem `AWSEC2Launch-Agent` Paket nicht unterstützt.
8. Überprüfen Sie, ob unter Name der Name des ausgewählten Pakets angegeben ist.
9. (Optional) Geben Sie unter Version den Versionsnamen des Pakets ein. Wenn Sie dieses Feld leer lassen, installiert Run Command die von Ihnen in Distributor ausgewählte Standardversion.
10. Wählen Sie im Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Geräte manuell auswählen oder eine Ressourcengruppe angeben.

    > **Note**
    >
    > Wenn kein verwalteter Knoten in der Liste angezeigt wird, lesen Sie Problembehandlung bei der Verfügbarkeit verwalteter Knoten.

11. Für Weitere Parameter:

    - Geben Sie im Feld Kommentar Informationen zu diesem Befehl ein.
    - Geben Sie für Timeout (Sekunden) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.
12. Für Rate control (Temposteuerung):

    - Geben Sie unter Concurrency (Nebenläufigkeit) entweder eine Zahl oder einen Prozentsatz von Zielen an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

> **Note**
>
> Wenn Sie Ziele ausgewählt haben, indem Sie Tags oder Ressourcengruppen angeben, und Sie noch nicht sicher sind, wie viele verwaltete Knoten anvisiert sind, sollten Sie die Anzahl von Zielen, die das Dokument gleichzeitig ausführen können, beschränken, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung des Befehls auf anderen Zielen beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von verwalteten Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

13. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, sind diejenigen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle (hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, und nicht die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

14. Aktivieren Sie das Kontrollkästchen SNS-Benachrichtigungen aktivieren im Abschnitt SNS-Benachrichtigungen, wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

15. Wenn Sie bereit sind, das Paket zu installieren, klicken Sie auf Run (Ausführen).
16. Im Bereich Command status (Befehlsstatus) wird der Fortschritt der Installation angezeigt. Wenn der Befehl noch ausgeführt wird, klicken Sie oben links in der Konsole auf das Aktualisierungssymbol, bis in der Spalte Overall status (Gesamtstatus) oder Detailed status (Detailstatus) der Status Success (Erfolgreich) oder Failed (Fehlgeschlagen) angezeigt wird.
17. Klicken Sie im Bereich Targets and outputs (Ziele und Ausgaben) auf die Schaltfläche neben dem Namen eines verwalteten Knotens und wählen Sie dann View output (Ausgabe anzeigen).

    Der Befehlsausgabeseite zeigt die Ergebnisse der Befehlsausführung an.
18. (Optional) Wenn Sie die Befehlsausgabe in einen Amazon S3-Bucket schreiben möchten, wählen Sie Amazon S3, um die Ausgabeprotokolldaten anzuzeigen.

Planen einer Paketinstallation oder -aktualisierung mithilfe der Konsole

Sie können die AWS Systems Manager Konsole verwenden, um die Installation oder Aktualisierung eines Pakets zu planen. Wenn Sie die Paketinstallation oder -aktualisierung planen, verwendet Distributor AWS Systems Manager State Manager zum Installieren oder Aktualisieren.

So planen Sie eine Paketinstallation mithilfe der Konsole

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Distributor-Startseite das Paket aus, das Sie installieren oder aktualisieren möchten.
4. Wählen Sie unter Package (Paket) die Option Install on a schedule (Nach Plan installieren) aus.

    Dieser Befehl öffnet State Manager zu einer neuen Zuordnung, die für Sie erstellt wurde.
5. Geben Sie unter Name einen Namen ein (z. B. **`Deploy-test-agent-package`**). Dies ist zwar optional, wird aber empfohlen. Der Name darf keine Leerzeichen enthalten.
6. In der Liste Document (Dokument) ist der Dokumentname `AWS-ConfigureAWSPackage` bereits ausgewählt.
7. Überprüfen Sie unter Action (Aktion), ob Install (Installieren) ausgewählt ist.
8. Wählen Sie unter Installation type (Installationstyp) eine der folgenden Optionen aus:

- Uninstall and reinstall (Deinstallieren und neu installieren): Das Paket wird vollständig deinstalliert und dann neu installiert. Die Anwendung ist bis zum Abschluss der Neuinstallation nicht verfügbar.
- In-place update (Direkte Aktualisierung): Der vorhandenen Installation werden entsprechend den Anweisungen, die Sie in einem `update`-Skript angeben, nur neue oder geänderte Dateien hinzugefügt. Die Anwendung ist während des Aktualisierungsprozesses weiterhin verfügbar.

9. Überprüfen Sie unter Name, ob der Name Ihres Pakets angegeben ist.
10. Geben Sie unter Version die Versionskennung ein, wenn Sie eine andere Paketversion als die zuletzt veröffentlichte Version installieren möchten.
11. Wählen Sie unter Targets (Ziele) die Optionen Selecting all managed instances in this account (Alle verwalteten Instances in diesem Konto auswählen), Specifying tags (Tags angeben) oder Manually Selecting Instance (Instance manuell auswählen) aus. Wenn Sie die Zielressourcen mithilfe von Tags ausgewählt haben, geben Sie einen Tag-Schlüssel und einen Tag-Wert in die entsprechenden Felder ein.

    > **Note**
    >
    > Sie können verwaltete AWS IoT Greengrass Kerngeräte auswählen, indem Sie entweder Alle verwalteten Instanzen in diesem Konto auswählen oder Instanz manuell auswählen wählen.

12. Wählen Sie unter Specify schedule (Plan angeben) die Option On Schedule (Nach Plan) aus, um die Zuordnung nach einem regelmäßigen Zeitplan auszuführen, oder No Schedule (Kein Plan), um die Zuordnung einmalig auszuführen. Weitere Informationen zu diesen Optionen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager. Verwenden Sie die Steuerelemente, um einen `cron`- oder Rate-Zeitplan für die Zuordnung zu erstellen.
13. Wählen Sie Zuordnung erstellen.
14. Klicken Sie auf der Seite Association (Zuordnung) auf die Schaltfläche neben der von Ihnen erstellten Zuordnung und wählen Sie dann Apply association now (Zuordnung jetzt anwenden) aus.

    State Manager erstellt die Zuordnung und führt sie sofort auf den angegebenen Zielen aus. Weitere Informationen zu den Ergebnissen der Ausführung von Zuordnungen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager in diesem Handbuch.

Weitere Informationen zur Verwendung der Optionen unter Advanced Options (Erweiterte Optionen), Rate control (Ratensteuerung) und Output options (Ausgabeoptionen) finden Sie unter Arbeiten mit Zuordnungen in Systems Manager.

Einmaliges Installieren eines Pakets mit dem AWS CLI

Sie können send-command den ausführen AWS CLI , um ein Distributor Paket einmal zu installieren. Wenn das Paket bereits installiert ist, wird die Anwendung offline geschaltet, während das Paket deinstalliert und stattdessen die neue Version installiert wird.

Um ein Paket einmal zu installieren, verwenden Sie AWS CLI

- Führen Sie in der AWS CLI den folgenden aus.

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "instance-IDs" \
    --parameters '{"action":["Install"],"installationType":["Uninstall and
 reinstall"],"name":["package-name (in same account) or package-ARN (shared from
 different account)"]}'
```

> **Note**
>
> Das Standardverhalten für `installationType` ist `Uninstall and reinstall`. Sie können `"installationType":["Uninstall and reinstall"]` im Befehl weglassen, wenn Sie ein komplettes Paket installieren.

Im Folgenden wird ein -Beispiel gezeigt.

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "i-00000000000000" \
    --parameters '{"action":["Install"],"installationType":["Uninstall and
 reinstall"],"name":["ExamplePackage"]}'
```

Informationen zu anderen Optionen, die Sie mit dem send-command Befehl verwenden können, finden Sie send-commandim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Einmaliges Aktualisieren eines Pakets mit dem AWS CLI

Sie können das Programm ausführen send-command AWS CLI , um ein Distributor Paket zu aktualisieren, ohne die zugehörige Anwendung offline zu schalten. Nur neue oder aktualisierte Dateien im Paket werden ersetzt.

Um ein Paket einmal zu aktualisieren, verwenden Sie AWS CLI

- Führen Sie in der AWS CLI den folgenden aus.

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "instance-IDs" \
    --parameters '{"action":["Install"],"installationType":["In-place
 update"],"name":["package-name (in same account) or package-ARN (shared from
 different account)"]}'
```

> **Note**
>
> Wenn Sie neue oder geänderte Dateien hinzufügen, müssen Sie `"installationType":["In-place update"]` in den Befehl einschließen.

Im Folgenden wird ein -Beispiel gezeigt.

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "i-02573cafcfEXAMPLE" \
    --parameters '{"action":["Install"],"installationType":["In-place
 update"],"name":["ExamplePackage"]}'
```

Informationen zu anderen Optionen, die Sie mit dem send-command Befehl verwenden können, finden Sie send-commandim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Planung einer Paketinstallation mit dem AWS CLI

Sie können create-association den ausführen AWS CLI , um ein Distributor Paket nach einem Zeitplan zu installieren. Der Wert für `--name`, d. h. der Name des Dokuments, ist stets `AWS-ConfigureAWSPackage`. Der folgende Befehl verwendet den Schlüssel `InstanceIds` zur Angabe

von verwalteten Knoten als Ziel. Wenn das Paket bereits installiert ist, wird die Anwendung offline geschaltet, während das Paket deinstalliert und stattdessen die neue Version installiert wird.

```
aws ssm create-association \
    --name "AWS-ConfigureAWSPackage" \
    --parameters '{"action":["Install"],"installationType":["Uninstall and
 reinstall"],"name":["package-name (in same account) or package-ARN (shared from
 different account)"]}' \
    --targets [{\"Key\":\"InstanceIds\",\"Values\":[\"instance-ID1\",\"instance-
ID2\"]}]
```

> **Note**
>
> Das Standardverhalten für `installationType` ist `Uninstall and reinstall`. Sie können `"installationType":["Uninstall and reinstall"]` im Befehl weglassen, wenn Sie ein komplettes Paket installieren.

Im Folgenden wird ein -Beispiel gezeigt.

```
aws ssm create-association \
    --name "AWS-ConfigureAWSPackage" \
    --parameters '{"action":["Install"],"installationType":["Uninstall and
 reinstall"],"name":["Test-ConfigureAWSPackage"]}' \
    --targets [{\"Key\":\"InstanceIds\",\"Values\":[\"i-02573cafcfEXAMPLE\",
\"i-0471e04240EXAMPLE\"]}]
```

Informationen zu anderen Optionen, die Sie mit dem create-association Befehl verwenden können, finden Sie create-associationim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Planung einer Paket-Aktualisierung mit dem AWS CLI

Sie können den ausführen create-association AWS CLI , um ein Distributor Paket nach einem Zeitplan zu aktualisieren, ohne die zugehörige Anwendung offline zu nehmen. Nur neue oder aktualisierte Dateien im Paket werden ersetzt. Der Wert für `--name`, d. h. der Name des Dokuments, ist stets `AWS-ConfigureAWSPackage`. Der folgende Befehl verwendet den Schlüssel `InstanceIds` zur Angabe von Ziel-Instances.

```
aws ssm create-association \
    --name "AWS-ConfigureAWSPackage" \
```

```
    --parameters '{"action":["Install"],"installationType":["In-place update"],"name":
["package-name (in same account) or package-ARN (shared from different account)"]}' \
    --targets [{\"Key\":\"InstanceIds\",\"Values\":[\"instance-ID1\",\"instance-
ID2\"]}]
```

> **Note**
>
> Wenn Sie neue oder geänderte Dateien hinzufügen, müssen Sie `"installationType": ["In-place update"]` in den Befehl einschließen.

Im Folgenden wird ein -Beispiel gezeigt.

```
aws ssm create-association \
    --name "AWS-ConfigureAWSPackage" \
    --parameters '{"action":["Install"],"installationType":["In-place update"],"name":
["Test-ConfigureAWSPackage"]}' \
    --targets [{\"Key\":\"InstanceIds\",\"Values\":[\"i-02573cafcfEXAMPLE\",
\"i-0471e04240EXAMPLE\"]}]
```

Informationen zu anderen Optionen, die Sie mit dem create-association Befehl verwenden können, finden Sie create-associationim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Deinstallieren eines Distributor-Pakets

Sie können die AWS-Managementkonsole oder die AWS Command Line Interface (AWS CLI) verwenden, um Distributor-Pakete aus Ihren von AWS Systems Manager verwalteten Knoten mithilfe von Run Command zu entfernen. Distributor und Run Command sind Tools in AWS Systems Manager. In dieser Version können Sie pro Befehl eine Version eines Pakets deinstallieren. Sie können eine bestimmte Version oder die Standardversion deinstallieren.

> **Important**
>
> Mit Distributor installierte Pakete sollten nur mithilfe von Distributor deinstalliert werden. Andernfalls kann Systems Manager die Anwendung weiterhin als `INSTALLED` erkennen und es können andere unbeabsichtigte Ergebnisse auftreten.

Themen

- Deinstallation eines Pakets über die Konsole
- Deinstallation eines Pakets mit dem AWS CLI

Deinstallation eines Pakets über die Konsole

Sie können Run Command in der Systems Manager-Konsole verwenden, um ein Paket einmalig zu deinstallieren. Distributor verwendet AWS Systems Manager Run Command zur Deinstallation von Paketen.

Um ein Paket mit der Konsole zu deinstallieren

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie auf der Run Command-Startseite Run command (Befehl ausführen) aus.
4. Wählen Sie das Befehlsdokument `AWS-ConfigureAWSPackage` aus.
5. Wählen Sie in Action (Aktion) die Option Uninstall (Deinstallieren) aus.
6. Geben Sie in Name (Name) den Namen des Pakets ein, das Sie deinstallieren möchten.
7. Für Targets (Ziele) wählen Sie aus, wie Sie Ihre verwaltete Knoten anvisieren möchten. Sie können einen Tag-Schlüssel und Werte angeben, die von den Zielen geteilt werden. Sie können Ziele auch angeben, indem Sie Attribute wie ID, Plattform und SSM Agent-Version auswählen.
8. Sie können in den erweiterten Optionen Kommentare zur Operation hinzufügen, die Werte für Concurrency (Gleichzeitigkeit) und Error threshold (Fehlerschwellenwert) in Rate control (Ratenkontrolle) ändern, Ausgabeoptionen angeben oder Amazon Simple Notification Service (Amazon SNS)-Benachrichtigungen konfigurieren. Weitere Informationen finden Sie unter Ausführen von Befehlen über die Konsole in diesem Handbuch.
9. Wenn Sie zur Deinstallation des Pakets bereit sind, wählen Sie Run (Ausführen) und dann View results (Ergebnisse anzeigen) aus.
10. Wählen Sie in der Befehlsliste den `AWS-ConfigureAWSPackage` aus, den Sie ausgeführt haben. Wenn der Befehl noch in Bearbeitung ist, wählen Sie das Aktualisierungssymbol oben rechts in der Konsole aus.
11. Wenn die Spalte Status Success (Erfolg) oder Failed (Fehlgeschlagen) anzeigt, wählen Sie die Registerkarte Output (Ausgabe).
12. Wählen Sie View output (Ausgabe anzeigen) aus. Der Befehlsausgabeseite zeigt die Ergebnisse der Befehlsausführung an.

Deinstallation eines Pakets mit dem AWS CLI

Sie können die AWS CLI verwenden, um ein Distributor-Paket mittels Run Command von Ihren verwalteten Knoten zu deinstallieren.

Um ein Paket zu deinstallieren, verwenden Sie AWS CLI

- Führen Sie in der AWS CLI den folgenden aus.

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "instance-IDs" \
    --parameters '{"action":["Uninstall"],"name":["package-name (in same account)
 or package-ARN (shared from different account)"]}'
```

  Im Folgenden wird ein Beispiel gezeigt.

```
aws ssm send-command \
    --document-name "AWS-ConfigureAWSPackage" \
    --instance-ids "i-02573cafcfEXAMPLE" \
    --parameters '{"action":["Uninstall"],"name":["Test-ConfigureAWSPackage"]}'
```

Informationen zu anderen Optionen, die Sie mit dem Befehl send-command verwenden können, finden Sie unter send-command im Abschnitt AWS Systems Manager der AWS CLI-Command Reference.

Löschen eines Distributor-Pakets

In diesem Abschnitt wird beschrieben, wie Sie ein Paket löschen. Sie können eine Version eines Pakets nicht löschen, sondern nur das gesamte Paket.

Löschen eines Pakets über die Konsole

Sie können die AWS Systems Manager-Konsole verwenden, um ein Paket oder eine Paketversion aus Distributor zu löschen, einem Tool in AWS Systems Manager. Durch das Löschen eines Pakets werden alle Versionen des Pakets aus Distributor gelöscht.

So löschen Sie ein Paket mithilfe der Konsole

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Distributor-Startseite das Paket aus, das Sie löschen möchten.
4. Wählen Sie auf der Detailseite des Pakets Delete package (Paket löschen) aus.
5. Wenn Sie zum Bestätigen des Löschvorgangs aufgefordert werden, wählen Sie Delete package (Paket löschen) aus.

Löschen einer Paketversion über die Konsole

Sie können die Systems Manager-Konsole verwenden, um eine Paketversion aus Distributor zu löschen.

Löschen einer Paketversion über die Konsole

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Distributor aus.
3. Wählen Sie auf der Distributor-Startseite das Paket aus, von dem Sie eine Version löschen möchten.
4. Wählen Sie auf der Versionsseite für das Paket die zu löschende Version und anschließend die Option Delete version (Version löschen) aus.
5. Wenn Sie zum Bestätigen des Löschvorgangs aufgefordert werden, wählen Sie Delete package version (Paketversion löschen) aus.

Ein Paket mithilfe einer Befehlszeile löschen

Sie können Ihr bevorzugtes Befehlszeilen-Tool verwenden, um ein Paket aus Distributor zu löschen.

Linux & macOS

So löschen Sie ein Paket mithilfe der AWS CLI

1. Führen Sie den folgenden Befehl aus, um Dokumente für spezifische Pakete aufzulisten. Suchen Sie in den Ergebnissen dieses Befehls das Paket, das Sie löschen möchten.

```
aws ssm list-documents \
    --filters Key=Name,Values=package-name
```

2. Führen Sie den folgenden Befehl aus, um ein Paket zu löschen. Ersetzen Sie *package-name* durch den Namen des Pakets.

```
aws ssm delete-document \
    --name "package-name"
```

3. Führen Sie den Befehl list-documents erneut aus, um zu überprüfen, ob das Paket gelöscht wurde. Das Paket, das Sie gelöscht haben, sollte nicht in die Liste aufgenommen werden.

```
aws ssm list-documents \
    --filters Key=Name,Values=package-name
```

Windows

So löschen Sie ein Paket mithilfe der AWS CLI

1. Führen Sie den folgenden Befehl aus, um Dokumente für spezifische Pakete aufzulisten. Suchen Sie in den Ergebnissen dieses Befehls das Paket, das Sie löschen möchten.

```
aws ssm list-documents ^
    --filters Key=Name,Values=package-name
```

2. Führen Sie den folgenden Befehl aus, um ein Paket zu löschen. Ersetzen Sie *package-name* durch den Namen des Pakets.

```
aws ssm delete-document ^
    --name "package-name"
```

3. Führen Sie den Befehl list-documents erneut aus, um zu überprüfen, ob das Paket gelöscht wurde. Das Paket, das Sie gelöscht haben, sollte nicht in die Liste aufgenommen werden.

```
aws ssm list-documents ^
    --filters Key=Name,Values=package-name
```

PowerShell

Löschen eines Pakets mit Tools for PowerShell

1. Führen Sie den folgenden Befehl aus, um Dokumente für spezifische Pakete aufzulisten. Suchen Sie in den Ergebnissen dieses Befehls das Paket, das Sie löschen möchten.

```
$filter = New-Object
 Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$filter.Key = "Name"
$filter.Values = "package-name"

Get-SSMDocumentList `
    -Filters @($filter)
```

2. Führen Sie den folgenden Befehl aus, um ein Paket zu löschen. Ersetzen Sie *package-name* durch den Namen des Pakets.

```
Remove-SSMDocument `
    -Name "package-name"
```

3. Führen Sie den Befehl Get-SSMDocumentList erneut aus, um zu überprüfen, ob das Paket gelöscht wurde. Das Paket, das Sie gelöscht haben, sollte nicht in die Liste aufgenommen werden.

```
$filter = New-Object
 Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$filter.Key = "Name"
$filter.Values = "package-name"

Get-SSMDocumentList `
    -Filters @($filter)
```

Eine Paketversion mithilfe einer Befehlszeile löschen

Sie können Ihr bevorzugtes Befehlszeilen-Tool verwenden, um eine Paketversion aus Distributor zu löschen.

Linux & macOS

So löschen Sie eine Paketversion mithilfe der AWS CLI

1. Führen Sie den folgenden Befehl aus, um die Versionen Ihres Pakets aufzulisten. Suchen Sie in den Ergebnissen dieses Befehls die Paketversion, die Sie löschen möchten.

```
aws ssm list-document-versions \
    --name "package-name"
```

2. Führen Sie den folgenden Befehl aus, um eine Paketversion zu löschen. Ersetzen Sie *package-name* durch den Paketnamen und *version* durch die Versionsnummer.

```
aws ssm delete-document \
    --name "package-name" \
    --document-version version
```

3. Führen Sie den Befehl list-document-versions aus, um zu überprüfen, ob die Version des Pakets gelöscht wurde. Die von Ihnen gelöschte Paketversion sollte nicht gefunden werden.

```
aws ssm list-document-versions \
    --name "package-name"
```

Windows

So löschen Sie eine Paketversion mithilfe der AWS CLI

1. Führen Sie den folgenden Befehl aus, um die Versionen Ihres Pakets aufzulisten. Suchen Sie in den Ergebnissen dieses Befehls die Paketversion, die Sie löschen möchten.

```
aws ssm list-document-versions ^
    --name "package-name"
```

2. Führen Sie den folgenden Befehl aus, um eine Paketversion zu löschen. Ersetzen Sie *package-name* durch den Paketnamen und *version* durch die Versionsnummer.

```
aws ssm delete-document ^
    --name "package-name" ^
    --document-version version
```

3. Führen Sie den Befehl list-document-versions aus, um zu überprüfen, ob die Version des Pakets gelöscht wurde. Die von Ihnen gelöschte Paketversion sollte nicht gefunden werden.

```
aws ssm list-document-versions ^
    --name "package-name"
```

PowerShell

So löschen Sie eine Paketversion mit Tools für PowerShell

1. Führen Sie den folgenden Befehl aus, um die Versionen Ihres Pakets aufzulisten. Suchen Sie in den Ergebnissen dieses Befehls die Paketversion, die Sie löschen möchten.

```
Get-SSMDocumentVersionList `
    -Name "package-name"
```

2. Führen Sie den folgenden Befehl aus, um eine Paketversion zu löschen. Ersetzen Sie *package-name* durch den Paketnamen und *version* durch die Versionsnummer.

```
Remove-SSMDocument `
    -Name "package-name" `
    -DocumentVersion version
```

3. Führen Sie den Befehl Get-SSMDocumentVersionList aus, um zu überprüfen, ob die Version des Pakets gelöscht wurde. Die von Ihnen gelöschte Paketversion sollte nicht gefunden werden.

```
Get-SSMDocumentVersionList `
    -Name "package-name"
```

Informationen zu anderen Optionen, die Sie mit dem Befehl list-documents verwenden können, finden Sie unter list-documents im Abschnitt AWS Systems Manager der AWS CLI-Command Reference. Informationen zu anderen Optionen, die Sie mit dem Befehl delete-document verwenden können, finden Sie unter delete-document.

Prüfen und Protokollieren von Distributor-Aktivitäten

Sie können AWS CloudTrail zur Prüfung der Aktivität im Zusammenhang mit Distributor verwenden, einem Tool in AWS Systems Manager. Weitere Informationen zu den Prüfungs- und Protokollierungsoptionen für Systems Manager finden Sie unter Einloggen und Überwachen AWS Systems Manager.

Distributor-Aktivität mithilfe von CloudTrail prüfen

CloudTrail erfasst über die AWS Systems Manager-Konsole, die AWS Command Line Interface (AWS CLI) und das Systems Manager SDK ausgeführte API-Aufrufe. Sie können die Informationen in der CloudTrail-Konsole oder in einem Amazon Simple Storage Service (Amazon S3)-Bucket anzeigen. Für alle CloudTrail-Protokolle in Ihrem Konto wird nur ein Bucket benötigt.

Protokolle von Run Command- und State Manager-Aktionen zeigen Aktivitäten im Zusammenhang mit Dokumenterstellung, Paketinstallation und Paketdeinstallation an. Run Command und State Manager sind Tools in AWS Systems Manager. Weitere Informationen zum Anzeigen und Verwenden von CloudTrail-Protokollen von Systems Manager-Aktivitäten finden Sie unter Protokollierung von AWS Systems Manager-API-Aufrufen mit AWS CloudTrail.

Fehlerbehebung für AWS Systems ManagerDistributor

Die folgenden Informationen können Ihnen helfen, Fehler zu beheben, die bei der Verwendung von Distributor, einem Tool in AWS Systems Manager, möglicherweise auftreten.

Themen

- Falsches Paket mit identischem Namen installiert
- Fehler: Abruf des Manifests fehlgeschlagen: Aktuelle Version des Pakets wurde nicht gefunden
- Fehler: Fehler beim Abrufen des Manifests: Validierungsausnahme
- Paket wird nicht unterstützt (dem Paket fehlt Installationsaktion)
- Fehler: Manifest konnte nicht heruntergeladen werden: Dokument mit dem Namen ist nicht vorhanden
- Hochladen fehlgeschlagen.
- Fehler: Failed to find platform: no manifest found for platform: oracle, version 8.9, architecture x86_64

Falsches Paket mit identischem Namen installiert

Problem: Sie haben ein Paket installiert, Distributor hat stattdessen ein anderes Paket installiert.

Ursache: Während der Installation findet Systems Manager von AWS veröffentlichte Pakete als Ergebnisse, bevor benutzerdefinierte externe Pakete gefunden werden. Wenn Ihr benutzerdefinierter Paketname mit dem Namen eines AWS veröffentlichten Pakets identisch ist, wird das AWS Paket anstelle Ihres Pakets installiert.

Lösung: Um dieses Problem zu vermeiden, geben Sie Ihrem Paket einen anderen Namen als den Namen eines AWS veröffentlichten Pakets.

Fehler: Abruf des Manifests fehlgeschlagen: Aktuelle Version des Pakets wurde nicht gefunden

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
Failed to retrieve manifest: ResourceNotFoundException: Could not find the latest
 version of package
arn:aws:ssm:::package/package-name status code: 400, request id: guid
```

Ursache: Sie verwenden eine SSM Agent-Version mit einer früheren Distributor-Version als Version 2.3.274.0.

Lösung: Aktualisieren Sie die SSM Agent-Version auf Version 2.3.274.0 oder höher. Weitere Informationen finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command oder Exemplarische Vorgehensweise: Automatisches Update SSM Agent mit AWS CLI.

Fehler: Fehler beim Abrufen des Manifests: Validierungsausnahme

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
Failed to retrieve manifest: ValidationException: 1 validation error detected: Value
 'documentArn'
at 'packageName' failed to satisfy constraint: Member must satisfy regular expression
 pattern:
arn:aws:ssm:region-id:account-id:package/package-name
```

Ursache: Sie verwenden eine SSM Agent-Version mit einer früheren Distributor-Version als Version 2.3.274.0.

Lösung: Aktualisieren Sie die SSM Agent-Version auf Version 2.3.274.0 oder höher. Weitere Informationen finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command oder Exemplarische Vorgehensweise: Automatisches Update SSM Agent mit AWS CLI.

Paket wird nicht unterstützt (dem Paket fehlt Installationsaktion)

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
Package is not supported (package is missing install action)
```

Ursache: Die Paketverzeichnisstruktur ist falsch.

Lösung: Zipen Sie kein übergeordnetes Verzeichnis, das die Software und die erforderlichen Skripte enthält. Erstellen Sie stattdessen eine `.zip`-Datei aller erforderlichen Inhalte direkt im absoluten Pfad. Um zu überprüfen, ob die`.zip`-Datei korrekt erstellt wurde, entpacken Sie das Zielplattformverzeichnis und überprüfen Sie die Verzeichnisstruktur. Der absolute Pfad für das Installationsskript sollte beispielsweise */ExamplePackage_targetPlatform*`/install.sh` sein.

Fehler: Manifest konnte nicht heruntergeladen werden: Dokument mit dem Namen ist nicht vorhanden

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
Failed to download manifest - failed to retrieve package document description:
 InvalidDocument: Document with name filename does not exist.
```

Ursache 1: Distributor kann das Paket nicht anhand des Paketnamens finden, wenn ein Distributor-Paket von einem anderen Konto geteilt wird.

Lösung 1: Wenn Sie ein Paket von einem anderen Konto freigeben, verwenden Sie den vollständigen Amazon-Ressourcennamen (ARN) für das Paket und nicht nur den Namen.

Ursache 2: Wenn Sie eine VPC verwenden, haben Sie Ihrem IAM-Instanzprofil keinen Zugriff auf den AWS verwalteten S3-Bucket gewährt, der das Dokument `AWS-ConfigureAWSPackage` für das enthält, auf das AWS-Region Sie abzielen.

Lösung 2: Stellen Sie sicher, dass Ihr IAM-Instanzprofil Zugriff auf den AWS verwalteten S3-Bucket bietetSSM Agent, der das Dokument `AWS-ConfigureAWSPackage` für das, auf das AWS-Region Sie abzielen, enthält, wie unter erklärt. SSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets

Hochladen fehlgeschlagen.

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
Upload failed. At least one of your files was not successfully uploaded to your S3
 bucket.
```

Ursache: Der Name Ihres Softwarepakets enthält ein Leerzeichen. Zum Beispiel würde bei `Hello World.msi` der Upload fehlschlagen.

Fehler: Failed to find platform: no manifest found for platform: oracle, version 8.9, architecture x86_64

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
Failed to find platform: no manifest found for platform: oracle, version 8.9,
 architecture x86_64
```

Ursache: Der Fehler bedeutet, dass das JSON-Paketmanifest keine Definition für diese Plattform hat, in diesem Fall Oracle Linux.

Lösung: Laden Sie das Paket, das Sie verteilen möchten, von der Trend Micro Deep Security Software-Website herunter. Erstellen Sie ein `.rpm`-Softwarepaket mit dem the section called “Erstellen Sie ein Paket mithilfe des einfachen Workflows”. Legen Sie die folgenden Werte für das Paket fest und schließen Sie das Upload-Softwarepaket mithilfe von Distributor ab:

```
Platform version: _any
Target platform: oracle
Architecture: x86_64
```

AWS Systems Manager Fleet Manager

Fleet Manager, ein Tool in AWS Systems Manager, ist eine einheitliche Benutzeroberfläche (UI), mit der Sie Ihre auf AWS oder On-Premises ausgeführten Knoten remote verwalten können. Mit Fleet Manager können Sie sich den Zustand und den Leistungsstatus Ihrer gesamten Serverflotte von einer Konsole aus ansehen. Sie können auch Daten aus einzelnen Knoten sammeln, um allgemeine Problembehandlungs- und Verwaltungsaufgaben über die Konsole auszuführen. Dies umfasst die Verbindung mit Windows-Instances über das Remote Desktop Protocol (RDP), das Anzeigen von Ordner- und Dateiinhalten, die Verwaltung der Windows-Registry, die Benutzerverwaltung des Betriebssystems und vieles mehr.

Um mit Fleet Manager zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Fleet Manager aus.

An wen richtet sich Fleet Manager?

Alle AWS-Kunden, die eine zentralisierte Methode zur Verwaltung ihrer Knotenflotte möchten, sollten Fleet Manager verwenden.

Welche Vorteile bietet Fleet Manager meiner Organisation?

Fleet Manager bietet die folgenden Vorteile:

- Ausführen einer Vielzahl gängiger Systemverwaltungs-Aufgaben, ohne eine manuelle Verbindung zu Ihren verwalteten Knoten herstellen zu müssen.
- Verwalten von Knoten, die auf mehreren Plattformen ausgeführt werden, über eine einzige einheitliche Konsole.
- Verwalten von Knoten, auf denen verschiedene Betriebssysteme ausgeführt werden, über eine einzige einheitliche Konsole.
- Verbessern Sie die Effizienz Ihrer Systemadministration.

Über welche Features verfügt Fleet Manager?

Nachstehend sind einige der wichtigsten Features von Fleet Manager aufgelistet:

- Zugreifen auf das Red-Hat-Knowledgebase-Portal

  Greifen Sie über Ihre Red Hat Enterprise Linux (RHEL)-Instances auf Binärdateien, Wissensfreigaben und Diskussionsforen im Knowledgebase-Portal von Red Hat zu.

- Status des verwalteten Knotens

  Anzeigen, welche verwalteten Knoten `running` sind und welche `stopped` sind. Weitere Informationen zu gestoppten Instances finden Sie unter Anhalten und Starten Ihrer Instance im Amazon-EC2-Benutzerhandbuch. Für AWS IoT Greengrass-Core-Geräte können Sie sehen, welche `online` oder `offline` sind, bzw. den Status `Connection lost` anzeigen.

> **Note**
>
> Wenn Sie Ihre verwaltete Instance vor dem 12. Juli 2021 angehalten haben, wird die `stopped`-Markierung nicht angezeigt. Um die Markierung anzuzeigen, starten und beenden Sie die Instance.

- Anzeigen von Instance-Informationen

  Zeigen Sie Informationen zu den Ordner- und Dateidaten an, die auf den an Ihre verwalteten Instances angeschlossenen Volumes gespeichert sind, sowie Leistungsdaten zu Ihren Instances in Echtzeit und auf Ihren Instances gespeicherte Protokolldaten.
- Informationen über Edge-Gerät anzeigen

  Aufrufen des AWS IoT Greengrass-Objektnamens für das Gerät, SSM Agent-Ping-Status und der Version und mehr.
- Verwalten von Konten und Registrierung

  Verwalten von Betriebssystem-Benutzerkonten auf Ihren Instances und Registrieren auf Ihren Windows-Instances.
- Steuern des Zugriffs auf Features

  Steuern des Zugriffs auf Fleet Manager-Features mit AWS Identity and Access Management (IAM)-Richtlinien. Mit diesen Richtlinien können Sie steuern, welche Benutzer oder Gruppen in Ihrer Organisation verschiedene Fleet Manager-Features verwenden können und welche verwalteten Knoten sie verwalten können.

Themen

- Einrichten von Fleet Manager
- Arbeiten mit verwalteten Knoten
- Automatisches Verwalten von EC2 Instanzen mit der Standard-Host-Management-Konfiguration
- Eine Verbindung zu einer von Windows Server verwalteten Instance mit Remote Desktop herstellen
- Verwaltung von Amazon-EBS-Volumes auf verwalteten Instances
- Zugriff auf das Wissensdatenbank-Portal von Red Hat
- Problembehandlung bei der Verfügbarkeit verwalteter Knoten

Einrichten von Fleet Manager

Bevor Benutzer in Ihrem AWS-Konto Fleet Manager verwenden können, ein Tool in AWS Systems Manager, um Ihre verwalteten Knoten zu überwachen und zu verwalten, müssen ihnen die erforderlichen Berechtigungen erteilt werden. Darüber hinaus müssen alle Amazon Elastic Compute Cloud (Amazon EC2)-Instances, AWS IoT Greengrass-Core-Geräte und On-Premises-Server, Edge-Geräte und virtuelle Maschinen (VMs), die mithilfe von Fleet Manager, überwacht und verwaltet werden können, verwaltete Knoten von Systems Manager sein. Ein verwalteter Knoten ist jede Maschine, die für die Verwendung mit Systems Manager in Hybrid- und Multi-Cloud-Umgebungen konfiguriert ist.

Dies bedeutet, dass Ihre Knoten bestimmte Voraussetzungen erfüllen und mit dem AWS Systems Manager-Agent (SSM Agent) konfiguriert sein müssen.

Je nach Maschinentyp sollten Sie sich mit einem der folgenden Themen befassen, um sicherzustellen, dass Ihre Computer die Anforderungen für verwaltete Knoten erfüllen.

- Amazon-EC2-Instances: Verwalten von EC2-Instances mit Systems Manager

> **Tip**
>
> Sie können auch Quick Setup, ein Tool in AWS Systems Manager, verwenden, um Ihre Amazon-EC2-Instances schnell als verwaltete Instances in einem einzelnen Konto konfigurieren zu können. Wenn Ihr Unternehmen oder Ihre Organisation AWS Organizations verwendet, können Sie auch Instances über mehrere Organisationseinheiten (OUs) und AWS-Regionen hinweg konfigurieren. Weitere Informationen zur Verwendung von Quick Setup zum Konfigurieren verwalteter Instance finden Sie unter Amazon-EC2-Host-Verwaltung mit Quick Setup einrichten.

- On-Premises und andere Servertypen in der Cloud: Verwalten von Knoten in Hybrid- und Multi-Cloud-Umgebungen mit Systems Manager
- AWS IoT Greengrass-(Edge)-Geräte: Verwalten von Edge-Geräten mit Systems Manager

Beispielrichtlinien

- Steuern des Zugriffs auf Fleet Manager

Steuern des Zugriffs auf Fleet Manager

Um ein Tool in verwenden zu können Fleet Manager AWS Systems Manager, muss Ihr AWS Identity and Access Management (IAM-) Benutzer oder Ihre Rolle über die erforderlichen Berechtigungen verfügen. Sie können eine IAM-Richtlinie erstellen, die Zugriff auf alle Fleet Manager-Features bieten, oder ändern Sie Ihre Richtlinie, um Zugriff auf die ausgewählten Features zu gewähren. Anschließend gewähren Sie diese Berechtigungen Benutzern oder Identitäten in Ihrem Konto.

Aufgabe 1: Erstellen von IAM-Richtlinien zur Definition von Zugriffsberechtigungen

Folgen Sie einer der Methoden, die im folgenden Thema des IAM-Benutzerhandbuchs beschrieben werden, um ein IAM zu erstellen, das Identitäten (Benutzern, Rollen oder Benutzergruppen) Zugriff gewährt auf: Fleet Manager

- Benutzerdefinierte IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien definieren

Sie können eine der unten aufgeführten Beispielrichtlinien verwenden oder sie entsprechend den Berechtigungen ändern, die Sie gewähren möchten. Wir bieten Beispielrichtlinien für vollständigen Fleet Manager-Zugriff und schreibgeschützten Zugriff.

Aufgabe 2: Ordnen Sie den Benutzern die IAM-Richtlinien zu, um ihnen Berechtigungen zu gewähren

Nachdem Sie die IAM-Richtlinie(n) erstellt haben, die Zugriffsberechtigungen für Fleet Manager definieren, verwenden Sie eines der folgenden Verfahren im IAM-Benutzerhandbuch, um diese Berechtigungen für Identitäten in Ihrem Konto zu gewähren:

- Hinzufügen von IAM-Identitätsberechtigungen (Konsole)
- Hinzufügen von IAM-Identitätsberechtigungen (AWS CLI)
- Hinzufügen von IAM-Identitätsberechtigungen (AWS -API)

Themen

- Beispielrichtlinie für Fleet Manager-Administratorzugriff
- Beispielrichtlinie für schreibgeschützten Fleet Manager-Zugriff

Beispielrichtlinie für Fleet Manager-Administratorzugriff

Die folgende Richtlinie gewährt Berechtigungen für alle Fleet Manager-Features. Dies bedeutet, dass ein Benutzer lokale Benutzer und Gruppen erstellen und löschen, die Gruppenmitgliedschaft für jede lokale Gruppe ändern und Windows Server-Registrierungsschlüssel oder -werte ändern kann. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "EC2",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags",
                "ec2:DeleteTags",
                "ec2:DescribeInstances",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        },
        {
            "Sid": "General",
            "Effect": "Allow",
            "Action": [
                "ssm:AddTagsToResource",
                "ssm:DescribeInstanceAssociationsStatus",
                "ssm:DescribeInstancePatches",
                "ssm:DescribeInstancePatchStates",
                "ssm:DescribeInstanceProperties",
                "ssm:GetCommandInvocation",
                "ssm:GetServiceSetting",
                "ssm:GetInventorySchema",
                "ssm:ListComplianceItems",
                "ssm:ListInventoryEntries",
                "ssm:ListTagsForResource",
                "ssm:ListCommandInvocations",
                "ssm:ListAssociations",
                "ssm:RemoveTagsFromResource"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DefaultHostManagement",
            "Effect": "Allow",
            "Action": [
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
```

```
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:servicesetting/ssm/
managed-instance/default-ec2-instance-management-role"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/service-role/
AWSSystemsManagerDefaultEC2InstanceManagementRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ssm.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "SendCommand",
            "Effect": "Allow",
            "Action": [
                "ssm:GetDocument",
                "ssm:SendCommand",
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:*:111122223333:instance/*",
                "arn:aws:ssm:*:111122223333:managed-instance/*",
                "arn:aws:ssm:*:111122223333:document/SSM-SessionManagerRunShell",
                "arn:aws:ssm:*:*:document/AWS-PasswordReset",
                "arn:aws:ssm:*:*:document/AWSFleetManager-AddUsersToGroups",
                "arn:aws:ssm:*:*:document/AWSFleetManager-CopyFileSystemItem",
                "arn:aws:ssm:*:*:document/AWSFleetManager-CreateDirectory",
                "arn:aws:ssm:*:*:document/AWSFleetManager-CreateGroup",
                "arn:aws:ssm:*:*:document/AWSFleetManager-CreateUser",
                "arn:aws:ssm:*:*:document/AWSFleetManager-CreateUserInteractive",
                "arn:aws:ssm:*:*:document/AWSFleetManager-
CreateWindowsRegistryKey",
                "arn:aws:ssm:*:*:document/AWSFleetManager-DeleteFileSystemItem",
                "arn:aws:ssm:*:*:document/AWSFleetManager-DeleteGroup",
                "arn:aws:ssm:*:*:document/AWSFleetManager-DeleteUser",
```

```
                "arn:aws:ssm:*:*:document/AWSFleetManager-
DeleteWindowsRegistryKey",
                "arn:aws:ssm:*:*:document/AWSFleetManager-
DeleteWindowsRegistryValue",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetDiskInformation",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetFileContent",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetFileSystemContent",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetGroups",
                "arn:aws:ssm:*:*:document/AWSFleetManager-
GetPerformanceCounters",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetProcessDetails",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetUsers",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetWindowsEvents",
                "arn:aws:ssm:*:*:document/AWSFleetManager-
GetWindowsRegistryContent",
                "arn:aws:ssm:*:*:document/AWSFleetManager-MountVolume",
                "arn:aws:ssm:*:*:document/AWSFleetManager-MoveFileSystemItem",
                "arn:aws:ssm:*:*:document/AWSFleetManager-RemoveUsersFromGroups",
                "arn:aws:ssm:*:*:document/AWSFleetManager-RenameFileSystemItem",
                "arn:aws:ssm:*:*:document/AWSFleetManager-
SetWindowsRegistryValue",
                "arn:aws:ssm:*:*:document/AWSFleetManager-StartProcess",
                "arn:aws:ssm:*:*:document/AWSFleetManager-TerminateProcess"
            ]
        },
        {
            "Sid": "TerminateSession",
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/aws:ssmmessages:session-id": [
                        "${aws:userid}"
                    ]
                }
            }
        }
    ]
}
```

Beispielrichtlinie für schreibgeschützten Fleet Manager-Zugriff

Die folgende Richtlinie gewährt Berechtigungen für alle schreibgeschützten Fleet Manager–Features. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "EC2",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        },
        {
            "Sid": "General",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeInstanceAssociationsStatus",
                "ssm:DescribeInstancePatches",
                "ssm:DescribeInstancePatchStates",
                "ssm:DescribeInstanceProperties",
                "ssm:GetCommandInvocation",
                "ssm:GetServiceSetting",
                "ssm:GetInventorySchema",
                "ssm:ListComplianceItems",
                "ssm:ListInventoryEntries",
                "ssm:ListTagsForResource",
                "ssm:ListCommandInvocations",
                "ssm:ListAssociations"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SendCommand",
            "Effect": "Allow",
            "Action": [
                "ssm:GetDocument",
```

```
                "ssm:SendCommand",
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:*:111122223333:instance/*",
                "arn:aws:ssm:*:111122223333:managed-instance/*",
                "arn:aws:ssm:*:111122223333:document/SSM-SessionManagerRunShell",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetDiskInformation",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetFileContent",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetFileSystemContent",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetGroups",
                "arn:aws:ssm:*:*:document/AWSFleetManager-
GetPerformanceCounters",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetProcessDetails",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetUsers",
                "arn:aws:ssm:*:*:document/AWSFleetManager-GetWindowsEvents",
                "arn:aws:ssm:*:*:document/AWSFleetManager-
GetWindowsRegistryContent"
            ]
        },
        {
            "Sid": "TerminateSession",
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/aws:ssmmessages:session-id": [
                        "${aws:userid}"
                    ]
                }
            }
        }
    ]
}
```

Arbeiten mit verwalteten Knoten

Ein verwalteter Knoten ist ein beliebiger Computer, für den er konfiguriert ist. AWS Systems Manager Sie können die folgenden Maschinentypen als verwaltete Knoten konfigurieren:

- Amazon Elastic Compute Cloud (Amazon EC2) -Instanzen
- Server in Ihren eigenen Räumlichkeiten (On-Premises-Server)
- AWS IoT Greengrass Kerngeräte
- AWS IoT und Geräte, die nicht zu den AWS Edge-Geräten gehören
- Virtuelle Maschinen (VMs), auch VMs in anderen Cloud-Umgebungen

Jede Maschine mit dem Präfix „mi-“ in der Systems-Manager-Konsole wurde mit einer Hybrid-Aktivierung als verwalteter Knoten konfiguriert. Edge-Geräte zeigen ihren AWS IoT Ding-Namen an.

> **Note**
>
> Die einzige unterstützte Funktion für macOS-Instances ist die Anzeige des Dateisystems.

Üner Systems Manager Instances-Kontingente

AWS Systems Manager bietet eine Stufe „Standard-Instances“ und eine Stufe „Advanced-Instances“. Beide unterstützen verwaltete Knoten in Ihrer Hybrid- und Multi-Cloud-Umgebung. Die Stufe „Standard-Instances“ ermöglicht es Ihnen, maximal 1.000 Maschinen pro Person zu registrieren. AWS-Konto AWS-Region Wenn Sie mehr als 1 000 Maschinen in einem einzigen Konto und einer Region anmelden müssen, verwenden Sie das Advanced-Instances-Kontingent. Sie können im Advanced-Instances-Kontingent so viele verwaltete Knoten erstellen, wie Sie möchten. Alle verwalteten Knoten, die für Systems Manager konfiguriert sind, werden auf pay-per-use Basis von Preisen berechnet. Weitere Informationen über das Aktivieren des Advanced-Instances-Kontingent finden Sie unter Aktivieren des Kontingents für erweiterte Instances. Weitere Informationen über die Preise finden Sie unter AWS Systems Manager – Preise.

Beachten Sie die folgenden zusätzlichen Informationen zur Ebene für Standard-Instances und zur Ebene für erweiterte Instances:

- Mithilfe von Advanced Instances können Sie in einer Hybrid- und Multi-Cloud-Umgebung auch eine Verbindung zu Ihren EC2 Nicht-Nodes AWS Systems Manager Session Manager herstellen. Session Managerbietet interaktiven Shell-Zugriff auf Ihre Instanzen. Weitere Informationen finden Sie unter AWS Systems Manager Session Manager.
- Das Kontingent für Standardinstanzen gilt auch für EC2 Instanzen, die eine lokale Systems Manager Manager-Aktivierung verwenden (was kein übliches Szenario ist).

- Um von Microsoft veröffentlichte Anwendungen auf lokalen Instanzen virtueller Maschinen (VMs) zu patchen, aktivieren Sie die Stufe „Advanced-Instances“. Die Nutzung des Advanced-Instances-Kontingents ist kostenpflichtig. Für Patch-Anwendungen, die von Microsoft auf Amazon Elastic Compute Cloud (Amazon EC2) -Instances veröffentlicht wurden, fallen keine zusätzlichen Gebühren an. Weitere Informationen finden Sie unter Informationen zum Patchen von Anwendungen, die von Microsoft unter Windows Server veröffentlicht wurden.

Anzeigen von verwalteten Knoten

Wenn Ihre verwalteten Knoten nicht in der Konsole aufgeführt werden, gehen Sie wie folgt vor:

1. Stellen Sie sicher, dass die Konsole in dem Bereich geöffnet ist, in AWS-Region dem Sie Ihre verwalteten Knoten erstellt haben. Über die Liste in der oberen, rechten Ecke der Konsole können Sie zwischen dein einzelnen Regionen wechseln.
2. Überprüfen Sie, ob die Einrichtungsschritte für Ihre verwalteten Knoten den Voraussetzungen von Systems Manager entsprechen. Weitere Informationen finden Sie unter Einrichten von verwalteten Knoten für AWS Systems Manager.
3. Vergewissern Sie sich, dass Sie den Hybrid-Aktivierungsprozess abgeschlossen haben, falls es sich nicht um EC2 Maschinen handelt. Weitere Informationen finden Sie unter Verwalten von Knoten in Hybrid- und Multi-Cloud-Umgebungen mit Systems Manager.

Beachten Sie folgende Zusatzinformationen:

- Die Fleet Manager Konsole zeigt keine EC2 Amazon-Knoten an, die beendet wurden.
- Systems Manager erfordert genaue Zeitreferenzen, um Operationen auf Ihren Maschinen auszuführen. Wenn auf Ihren verwalteten Knoten das Datum und die Uhrzeit nicht korrekt eingestellt wurden, stimmen sie möglicherweise nicht mit dem Signaturdatum Ihrer API-Anforderungen überein. Weitere Informationen finden Sie unter Anwendungsfälle und bewährte Methoden.
- Wenn Sie Tags erstellen oder bearbeiten, kann das System bis zu einer Stunde benötigen, bis Änderungen im Tabellenfilter angezeigt werden.
- Nachdem der Status eines verwalteten Knotens mindestens 30 Tage lang `Connection Lost` gewesen ist, wird der Knoten möglicherweise nicht mehr in der Fleet Manager-Konsole aufgeführt. Beheben Sie das Problem, das den Verbindungsverlust verursacht hat, um ihn wieder in die Liste aufzunehmen. Tipps zur Fehlerbehebung finden Sie unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten.

Überprüfen des Supports für Systems Manager auf einem verwalteten Knoten

AWS Config stellt AWS verwaltete Regeln bereit. Dabei handelt es sich um vordefinierte, anpassbare Regeln, AWS Config anhand derer bewertet wird, ob Ihre AWS Ressourcenkonfigurationen den gängigen bewährten Methoden entsprechen. AWS Config Zu den verwalteten Regeln gehört die instance-managed-by-systemsec2-manager-Regel. Diese Regel prüft, ob die EC2 Amazon-Instances in Ihrem Konto von Systems Manager verwaltet werden. Weitere Informationen finden Sie unter AWS Config Managed Rules.

Erhöhen des Sicherheitsstatus auf verwalteten Knoten

Informationen zur Steigerung des Sicherheitsstatus in Bezug auf nicht autorisierte Befehle auf Root-Ebene auf Ihren verwalteten Knoten finden Sie unter Einschränken des Zugriffs auf Befehle auf Stammebene durch SSM Agent.

Abmelden verwalteter Knoten

Sie können verwaltete Knoten jederzeit abmelden. Wenn Sie beispielsweise mehrere Knoten mit derselben AWS Identity and Access Management (IAM-) Rolle verwalten und irgendein bösartiges Verhalten feststellen, können Sie jederzeit eine beliebige Anzahl von Computern abmelden. (Um dasselbe Gerät erneut zu registrieren, müssen Sie einen anderen Hybrid-Aktivierungscode und eine andere Aktivierungs-ID als zuvor verwenden.) Informationen über das Abmelden verwalteter Knoten finden Sie unter Aufheben der Registrierung von verwalteten Knoten in einer Hybrid- und Multi-Cloud-Umgebung.

Themen

- Konfigurieren von Instance-Kontingenten
- Zurücksetzen von Passwörtern für verwaltete Knoten
- Aufheben der Registrierung von verwalteten Knoten in einer Hybrid- und Multi-Cloud-Umgebung
- Arbeiten mit Betriebssystemdateisystemen mithilfe von Fleet Manager
- Überwachung der Leistung verwalteter Knoten
- Arbeiten mit Prozessen
- Protokolle auf verwalteten Knoten anzeigen
- Verwalten von Betriebssystembenutzerkonten und Gruppen auf verwalteten Knoten mithilfe von Fleet Manager
- Verwalten der Windows-Registrierung auf verwalteten Knoten

Konfigurieren von Instance-Kontingenten

In diesem Thema werden die Szenarien beschrieben, in denen Sie das Advanced-Instances-Kontingent aktivieren müssen.

AWS Systems Manager bietet ein Standard-Instances-Kontingent und ein Kontingent für erweiterte Instances für Nicht-EC2-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung.

Sie können maximal 1 000 standardmäßige hybrid-aktivierte Knoten pro Konto pro AWS-Region ohne zusätzliche Kosten registrieren. Um mehr als 1 000 Hybrid-Knoten zu registrieren, müssen Sie jedoch das Advanced-Instances-Kontingent aktivieren. Die Nutzung des Advanced-Instances-Kontingents ist kostenpflichtig. Weitere Informationen finden Sie unter AWS Systems Manager – Preise.

Selbst mit weniger als 1 000 registrierten hybrid-aktivierten Knoten erfordern zwei weitere Szenarien des Advanced-Instances-Kontingents:

- Sie möchten Session Manager verwenden, um eine Verbindung zu Nicht-EC2-Knoten herzustellen.
- Sie möchten Anwendungen (nicht Betriebssysteme), die von Microsoft auf Nicht-EC2-Knoten veröffentlicht wurden, patchen.

> **Note**
>
> Für Patch-Anwendungen, die von Microsoft auf Amazon-EC2-Instances veröffentlicht wurden, fallen keine Gebühren an.

Detaillierte Szenarien für Advanced-Instances-Kontingent

Die folgenden Informationen enthalten Details zu den drei Szenarien, für die Sie das Advanced-Instances-Kontingent aktivieren müssen.

Szenario 1: Sie möchten mehr als 1 000 hybrid-aktivierte Knoten registrieren

Mithilfe des Standard-Instances-Kontingents können Sie maximal 1 000 Nicht-EC2-Knoten in einer Hybrid- und Multi-Cloud-Umgebung pro AWS-Region in einem bestimmten Konto ohne zusätzliche Kosten registrieren. Wenn Sie mehr als 1 000 Nicht-EC2-Knoten in einer Region anmelden müssen, müssen Sie das Advanced-Instances-Kontingent verwenden. Sie können dann so viele Maschinen für Ihre Hybrid- und Multi-Cloud-Umgebung aktivieren, wie Sie möchten. Die Gebühren für das Advanced-Instance-Kontingent basieren auf der Anzahl der Advanced-

Knoten, die als von Systems Manager verwaltete Knoten aktiviert wurden, und den Stunden, in denen diese Knoten ausgeführt werden.

Für alle von Systems Manager verwalteten Knoten, die den unter Eine Hybridaktivierung zum Registrieren von Knoten bei Systems Manager erstellen beschriebenen Aktivierungsprozess verwenden, wird eine Gebühr erhoben, wenn Sie in einer Region in einem bestimmten Konto die Zahl von 1 000 On-Premises-Knoten überschreiten.

> **Note**
>
> Sie können auch vorhandene Amazon Elastic Compute Cloud (Amazon EC2)-Instances mithilfe von Hybrid-Aktivierungen von Systems Manager installieren und mit ihnen als Nicht-EC2-Instances arbeiten, z. B. zu Testzwecken. Diese zählen auch als Hybrid-Knoten. Dies ist kein übliches Szenario.

Szenario 2: Patchen von von Microsoft veröffentlichten Anwendungen auf hybrid-aktivierten Knoten

Das Advanced-Instances-Kontingent ist auch erforderlich, wenn Sie von Microsoft veröffentlichte Anwendungen auf Nicht-EC2-Knoten in einer Hybrid- und Multi-Cloud-Umgebung patchen möchten. Wenn Sie das Advanced-Instances-Kontingent aktivieren, um Microsoft-Anwendungen auf Nicht-EC2-Knoten zu patchen, fallen Gebühren für alle On-Premises-Knoten an, selbst wenn Sie weniger als 1 000 haben.

Für Patch-Anwendungen, die von Microsoft auf Amazon Elastic Compute Cloud (Amazon EC2)-Instances veröffentlicht wurden, fallen keine zusätzlichen Gebühren an. Weitere Informationen finden Sie unter Informationen zum Patchen von Anwendungen, die von Microsoft unter Windows Server veröffentlicht wurden.

Szenario 3: Herstellen einer Verbindung zu hybrid-aktivierten Knoten mit Session Manager

Session Manager bietet interaktiven Shell-Zugriff auf Ihre Instances. Um über Session Manager eine Verbindung zu hybrid-aktivierten Knoten herzustellen, müssen Sie die Advanced-Instances-Kontingente aktivieren. Dann fallen Gebühren für alle hybrid-aktivierten Knoten an, auch wenn Sie weniger als 1 000 haben.

Zusammenfassung: Wann brauche ich das Advanced-Instances-Kontingent?

Anhand der folgenden Tabelle können Sie überprüfen, wann Sie das Advanced-Instances-Kontingent verwenden müssen und für welche Szenarien zusätzliche Gebühren anfallen.

Szenario	Advanced-Instances-Kontingent erforderlich?	Es fallen zusätzliche Gebühren an.
Die Anzahl der hybrid-aktivierten Knoten in meiner Region in einem bestimmten Konto beträgt mehr als 1 000.	Ja	Ja
Ich möchte Patch Manager benutzen, um von Microsoft veröffentlichte Anwendungen auf einer beliebigen Anzahl hybrid-aktivierter Knoten zu patchen, sogar weniger als 1 000.	Ja	Ja
Ich möchte Session Manager benutzen, um sich mit einer beliebigen Anzahl hybrid-aktivierter Knoten zu verbinden, sogar weniger als 1 000.	Ja	Ja
1. Die Anzahl der hybrid-aktivierten Knoten in meiner Region in einem bestimmten Konto beträgt 1 000 oder weniger; und 2. Ich patche keine Microsoft-Anwendungen auf hybrid-aktivierten Knoten; und 3. Ich verbinde mich nicht mit hybrid-aktivierten Knoten mit Session Manager.	Nein	Nein

Themen

- Aktivieren des Kontingents für erweiterte Instances

- Wechsel vom Kontingent für erweiterte Instances zurück zum Kontingent für Standard-Instances

Aktivieren des Kontingents für erweiterte Instances

AWS Systems Manager bietet in einer Hybrid- und Multi-Cloud-Umgebung eine Stufe „Standard-Instances“ und eine Stufe „Advanced-Instances“ für EC2 Nicht-Computer. Mit der Stufe „Standard-Instances“ können Sie pro Person maximal 1.000 hybridaktivierte Maschinen registrieren. AWS-Konto AWS-Region Die Advanced-Instance-Stufe ist auch erforderlich, Patch Manager um von Microsoft veröffentlichte Anwendungen auf EC2 Nicht-Knoten zu patchen und mithilfe von Verbindungen zu Nicht-Knoten herzustellen. EC2 Session Manager Weitere Informationen finden Sie unter Aktivieren des Kontingents für erweiterte Instances.

In diesem Abschnitt wird beschrieben, wie Sie Ihre Hybrid- und Multi-Cloud-Umgebung für die Nutzung des Kontingents für erweiterte Instances konfigurieren.

Bevor Sie beginnen

Prüfen Sie die Preisdetails für erweiterte Instances. Advanced-Instances sind auf einem verfügbar. per-use-basis Weitere Informationen dazu finden Sie unter AWS Systems Manager -Preise.

Konfigurieren von Berechtigungen zum Aktivieren des Kontingents für erweiterte Instances

Vergewissern Sie sich, dass Sie in AWS Identity and Access Management (IAM) berechtigt sind, Ihre Umgebung von der Stufe Standard-Instances auf die Stufe Advanced-Instances umzustellen. Sie müssen entweder die `AdministratorAccess`-IAM-Richtlinie an Ihren Benutzer, Ihre Gruppe oder Ihre Rolle anfügen oder über die Berechtigung zum Ändern der Service-Einstellung für die Aktivierungsebene in Systems Manager verfügen. Diese nutzt die folgenden API-Operationen:

- GetServiceSetting
- UpdateServiceSetting
- ResetServiceSetting

Gehen Sie wie folgt vor, um einem Benutzerkonto eine IAM-Richtlinie hinzufügen. Mit dieser Richtlinie können Benutzer die aktuelle Einstellung für das Kontingent für verwaltete Instances anzeigen. Diese Richtlinie ermöglicht es dem Benutzer auch, die aktuelle Einstellung im angegebenen und zu ändern oder zurückzusetzen. AWS-Konto AWS-Region

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.

2. Klicken Sie im Navigationsbereich auf Users (Benutzer).
3. Wählen Sie in der Liste den Namen des Benutzers aus, in den Sie die Richtlinie einbinden möchten.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Klicken Sie rechts auf der Seite unter Permission policies (Berechtigungsrichtlinien) auf Add inline policy (Inline-Richtlinie hinzufügen).
6. Wählen Sie den Tab JSON.
7. Ersetzen Sie den Standardinhalt durch folgenden Inhalt:

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:servicesetting/
ssm/managed-instance/activation-tier"
        }
    ]
}
```

8. Wählen Sie Richtlinie prüfen.
9. Geben Sie auf der Seite Richtlinie prüfen im Feld Name einen Namen für die Inline-Richtlinie ein. Beispiel: **`Managed-Instances-Tier`**.
10. Wählen Sie Richtlinie erstellen aus.

Administratoren können schreibgeschützte Berechtigungen angeben, indem sie dem Benutzer die folgende Inline-Richtlinie zuweisen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "*"
        }
    ]
}
```

Informationen zum Erstellen einer IAM-Richtlinie finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Aktivieren des Kontingents für erweiterte Instances (Konsole)

Das folgende Verfahren zeigt Ihnen, wie Sie mit der Systems Manager Manager-Konsole alle EC2 Nicht-Knoten, die mithilfe der Aktivierung von verwalteten Instanzen hinzugefügt wurden, in der angegebenen AWS-Konto und AWS-Region auf die Advanced-Instance-Stufe umstellen.

Bevor Sie beginnen

Stellen Sie sicher, dass die Konsole in dem Bereich geöffnet ist, in AWS-Region dem Sie Ihre verwalteten Instanzen erstellt haben. Über die Liste in der oberen, rechten Ecke der Konsole können Sie zwischen dein einzelnen Regionen wechseln.

Stellen Sie sicher, dass Sie die Einrichtungsvoraussetzungen für Ihre Amazon Elastic Compute Cloud (Amazon EC2) -Instances und EC2 Nicht-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung erfüllt haben. Weitere Informationen finden Sie unter Einrichten von verwalteten Knoten für AWS Systems Manager.

> **Important**
>
> Nachfolgend wird beschrieben, wie Sie eine Einstellung auf Kontoebene ändern. Durch diese Änderung fallen für Ihr Konto Gebühren an.

So aktivieren Sie das Kontingent für erweiterte Instances (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie Einstellungen und anschließend Einstellungen der Instance-Ebene ändern.
4. Überprüfen Sie die Informationen im Dialogfeld zum Ändern der Kontoeinstellungen.
5. Wenn Sie zustimmen, wählen Sie die Option, die Sie akzeptieren möchten, und klicken Sie dann auf Einstellung ändern.

Es kann einige Minuten dauern, bis alle Instances vom Kontingent für Standard-Instances in das Kontingent für erweiterte Instances verschoben wurden.

> **Note**
>
> Informationen zum Wechsel zurück zum Kontingent für Standard-Instances finden Sie unter Wechsel vom Kontingent für erweiterte Instances zurück zum Kontingent für Standard-Instances.

Aktivieren des Kontingents für erweiterte Instances (AWS CLI)

Das folgende Verfahren zeigt Ihnen, wie Sie alle lokalen Server AWS Command Line Interface , die mithilfe der Aktivierung von verwalteten Instanzen hinzugefügt wurden VMs , in der angegebenen Version ändern AWS-Konto und AWS-Region, um die Stufe Advanced-Instances zu verwenden.

> **Important**
>
> Nachfolgend wird beschrieben, wie Sie eine Einstellung auf Kontoebene ändern. Durch diese Änderung fallen für Ihr Konto Gebühren an.

Um die Stufe „Advanced-Instances“ zu aktivieren, verwenden Sie den AWS CLI

1. Öffnen Sie den AWS CLI und führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   Linux & macOS

   ```
   aws ssm update-service-setting \
       --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-instance/activation-tier \
       --setting-value advanced
   ```

   Windows

   ```
   aws ssm update-service-setting ^
       --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-instance/activation-tier ^
       --setting-value advanced
   ```

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl aus, um die aktuellen Diensteinstellungen für verwaltete Knoten im aktuellen AWS-Konto und anzuzeigen AWS-Region.

   Linux & macOS

   ```
   aws ssm get-service-setting \
       --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-instance/activation-tier
   ```

   Windows

   ```
   aws ssm get-service-setting ^
   ```

```
    --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
instance/activation-tier
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "ServiceSetting": {
        "SettingId": "/ssm/managed-instance/activation-tier",
        "SettingValue": "advanced",
        "LastModifiedDate": 1555603376.138,
        "LastModifiedUser": "arn:aws:sts::123456789012:assumed-role/
Administrator/User_1",
        "ARN": "arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/managed-
instance/activation-tier",
        "Status": "PendingUpdate"
    }
}
```

Aktivieren des Kontingents für erweiterte Instances (PowerShell)

Das folgende Verfahren zeigt Ihnen, wie Sie alle lokalen Server AWS Tools for Windows PowerShell , die mithilfe der Aktivierung von verwalteten Instanzen hinzugefügt wurden VMs , in der angegebenen AWS-Konto Ebene ändern und, um die Stufe „ AWS-Region Advanced-Instances“ zu verwenden.

> **Important**
>
> Nachfolgend wird beschrieben, wie Sie eine Einstellung auf Kontoebene ändern. Durch diese Änderung fallen für Ihr Konto Gebühren an.

Um die Advanced-Instance-Stufe zu aktivieren, verwenden Sie PowerShell

1. Öffnen Sie den folgenden AWS Tools for Windows PowerShell Befehl und führen Sie ihn aus. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

```
Update-SSMServiceSetting `
    -SettingId "arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
instance/activation-tier" `
    -SettingValue "advanced"
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl aus, um die aktuellen Diensteinstellungen für verwaltete Knoten im aktuellen AWS-Konto und anzuzeigen AWS-Region.

```
Get-SSMServiceSetting `
    -SettingId "arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
instance/activation-tier"
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
ARN:arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/managed-instance/
activation-tier
LastModifiedDate : 4/18/2019 4:02:56 PM
LastModifiedUser : arn:aws:sts::123456789012:assumed-role/Administrator/User_1
SettingId        : /ssm/managed-instance/activation-tier
SettingValue     : advanced
Status           : PendingUpdate
```

Es kann einige Minuten dauern, bis alle Knoten vom Standard-Instances-Kontingent in das Advanced-Instances-Kontingent verschoben wurden.

> **Note**
>
> Informationen zum Wechsel zurück zum Kontingent für Standard-Instances finden Sie unter Wechsel vom Kontingent für erweiterte Instances zurück zum Kontingent für Standard-Instances.

Wechsel vom Kontingent für erweiterte Instances zurück zum Kontingent für Standard-Instances

In diesem Abschnitt wird beschrieben, wie hybrid-aktivierte Knoten, die im Advanced-Instances-Kontingent ausgeführt werden, wieder zum Standard-Instances-Kontingent umgestellt werden. Diese Konfiguration gilt für alle hybrid-aktivierten Knoten in einem AWS-Konto und einer einzelnen AWS-Region.

Bevor Sie beginnen

Lesen Sie die folgenden wichtigen Details.

> **Note**
>
> - Sie können nicht wieder zum Kontingent für Standard-Instances zurückkehren, wenn Sie im Konto und in der Region mehr als 1 000 hybrid-aktivierte Knoten ausführen. Sie müssen also zunächst die Registrierung für Knoten aufheben, bis 1 000 oder weniger vorhanden sind. Dies gilt auch für Amazon Elastic Compute Cloud (Amazon EC2)-Instances, die eine Hybrid-Aktivierung von Systems Manager verwenden (kein gängiges Szenario). Weitere Informationen finden Sie unter Aufheben der Registrierung von verwalteten Knoten in einer Hybrid- und Multi-Cloud-Umgebung.
> - Nach dem Zurücksetzen können Sie Session Manager, ein Tool in AWS Systems Manager, nicht mehr verwenden, um interaktiv auf Ihre hybrid-aktivierten Knoten zuzugreifen.
> - Nach dem Zurücksetzen können Sie Patch Manager, ein Tool in AWS Systems Manager, nicht mehr verwenden, um von Microsoft veröffentlichte Anwendungen auf hybrid-aktivierten Knoten zu patchen.
> - Das Zurücksetzen aller hybrid-aktivierten Knoten auf das Kontingent für Standard-Instances kann 30 Minuten oder länger dauern.

In diesem Abschnitt wird beschrieben, wie alle hybrid-aktvierten Knoten in einem AWS-Konto und einer AWS-Region vom Advanced-Instances-Kontingent auf das Standard-Instances-Kontingent zurückgesetzt werden können.

Zurücksetzen auf das Kontingent für Standard-Instances (Konsole)

Das folgende Verfahren veranschaulicht, wie Sie die Systems-Manager-Konsole verwenden, um alle hybrid-aktivierten Knoten in Ihrer Hybrid- und Multi-Cloud-Umgebung so zu ändern, dass das Kontingent für Standard-Instances im angegebenen AWS-Konto und in der angegebenen AWS-Region verwendet wird.

So stellen Sie das Kontingent für Standard-Instances wieder her (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie das Dropdown-Menü Account settings (Kontoeinstellungen) und wählen Sie Instance tier settings (Instance-Kontingenteneinstellungen).
4. Wählen Sie die Option Change account settings (Kontoeinstellungen ändern) aus.

5. Lesen Sie die Informationen zum Ändern der Kontoeinstellungen im angezeigten Popup-Fenster und wählen Sie ggf. die Option zum Akzeptieren und Fortfahren aus.

Zurücksetzen auf das Kontingent für Standard-Instances (AWS CLI)

Das folgende Verfahren veranschaulicht, wie Sie die AWS Command Line Interface verwenden, um alle hybrid-aktivierten Knoten in Ihrer Hybrid- und Multi-Cloud-Umgebung so zu ändern, dass das Kontingent für Standard-Instances im angegebenen AWS-Konto und in der angegebenen AWS-Region verwendet wird.

So stellen Sie das Kontingent für Standard-Instances mithilfe der AWS CLI wieder her

1. Öffnen Sie die AWS CLI und führen Sie den folgenden Befehl aus: Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux & macOS

   ```
   aws ssm update-service-setting \
       --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
   instance/activation-tier \
       --setting-value standard
   ```

   Windows

   ```
   aws ssm update-service-setting ^
       --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
   instance/activation-tier ^
       --setting-value standard
   ```

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl 30 Minuten danach aus, um die Einstellungen für verwaltete Instances im aktuellen AWS-Konto und in der aktuellen AWS-Region anzuzeigen.

   Linux & macOS

   ```
   aws ssm get-service-setting \
       --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
   instance/activation-tier
   ```

Windows

```
aws ssm get-service-setting ^
    --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
instance/activation-tier
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "ServiceSetting": {
        "SettingId": "/ssm/managed-instance/activation-tier",
        "SettingValue": "standard",
        "LastModifiedDate": 1555603376.138,
        "LastModifiedUser": "System",
        "ARN": "arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/managed-
instance/activation-tier",
        "Status": "Default"
    }
}
```

Der Status ändert sich auf Standard, nachdem die Anfrage genehmigt wurde.

Zurücksetzen auf das Kontingent für Standard-Instances (PowerShell)

Das folgende Verfahren veranschaulicht, wie Sie AWS Tools for Windows PowerShell verwenden, um alle hybrid-aktivierten Knoten in Ihrer Hybrid- und Multi-Cloud-Umgebung so zu ändern, dass das Kontingent für Standard-Instances im angegebenen AWS-Konto und in der angegebenen AWS-Region verwendet wird.

So stellen Sie das Kontingent für Standard-Instances mithilfe von PowerShell wieder her

1. Öffnen sie AWS Tools for Windows PowerShell und führen Sie den folgenden Befehl aus:

```
Update-SSMServiceSetting `
    -SettingId "arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
instance/activation-tier" `
    -SettingValue "standard"
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl 30 Minuten danach aus, um die Einstellungen für verwaltete Instances im aktuellen AWS-Konto und in der aktuellen AWS-Region anzuzeigen.

```
Get-SSMServiceSetting `
    -SettingId "arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
instance/activation-tier"
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
ARN: arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/managed-instance/
activation-tier
LastModifiedDate : 4/18/2019 4:02:56 PM
LastModifiedUser : System
SettingId        : /ssm/managed-instance/activation-tier
SettingValue     : standard
Status           : Default
```

Der Status ändert sich auf Standard, nachdem die Anfrage genehmigt wurde.

Zurücksetzen von Passwörtern für verwaltete Knoten

Sie können das Passwort für einen beliebigen Benutzer auf einem verwalteten Knoten zurücksetzen. Dazu gehören Amazon Elastic Compute Cloud (Amazon EC2) -Instances, AWS IoT Greengrass Kerngeräte und lokale Server, Edge-Geräte und virtuelle Maschinen (VMs), die von AWS Systems Manager verwaltet werden. Die Funktionalität zum Zurücksetzen des Passworts basiert auf Session Manager, einem Tool in AWS Systems Manager. Sie können diese Funktionalität verwenden, um eine Verbindung zu verwalteten Knoten herzustellen, ohne eingehende Ports öffnen, Bastion-Hosts zu pflegen oder SSH-Schlüssel verwalten zu müssen.

Die Option zum Zurücksetzen des Passworts ist nützlich, wenn ein Benutzer ein Passwort vergessen hat, oder wenn Sie schnell ein Passwort aktualisieren möchten, ohne eine RDP- oder SSH-Verbindung mit einem verwalteten Knoten herzustellen.

Voraussetzungen

Um das Passwort auf einem verwalteten Knoten zurücksetzen zu können, müssen die folgenden Anforderungen erfüllt sein:

- Der verwaltete Knoten, auf dem Sie ein Passwort ändern möchten, muss ein von Systems Manager verwalteter Knoten sein. Auch muss SSM Agent Version 2.3.668.0 oder höher auf dem

verwalteten Knoten installiert sein. Informationen über das Installieren oder Aktualisieren von SSM Agent finden Sie unter Arbeiten mit SSM Agent.

- Die Funktionalität zum Zurücksetzen des Passworts verwendet die Session Manager-Konfiguration, die für Ihr Konto eingerichtet ist, um eine Verbindung mit dem verwalteten Knoten herzustellen. Aus diesem Grund müssen die Voraussetzungen für die Verwendung von Session Manager für Ihr Konto in der aktuellen AWS-Region erfüllt sein. Weitere Informationen finden Sie unter Einrichten von Session Manager.

  > **Note**
  >
  > Session Manager-Support für On-Premises-Knoten wird nur für das Advanced-Instances-Kontingent bereitgestellt. Weitere Informationen finden Sie unter Aktivieren des Kontingents für erweiterte Instances.

- Der AWS Benutzer, der das Passwort ändert, muss über die `ssm:SendCommand` Berechtigung für den verwalteten Knoten verfügen. Weitere Informationen finden Sie unter Den Zugriff von Run Command anhand von Tags beschränken.

Beschränken des Zugriffs

Sie können die Fähigkeit eines Benutzers, Passwörter zurückzusetzen, auf bestimmte verwaltete Knoten beschränken. Dies geschieht durch Verwendung von identitätsbasierten Richtlinien für die Session Manager-Operation `ssm:StartSession` über dem SSM-Dokument `AWS-PasswordReset`. Weitere Informationen finden Sie unter Kontrollieren des Sitzungszugriffs von Benutzern auf Instances.

Verschlüsseln von Daten

Aktivieren Sie AWS Key Management Service (AWS KMS) die vollständige Verschlüsselung für Session Manager Daten, um die Option zum Zurücksetzen des Kennworts für verwaltete Knoten zu verwenden. Weitere Informationen finden Sie unter So aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole).

Zurücksetzen eines Passworts auf einem verwalteten Knoten

Sie können ein Passwort auf einem von Systems Manager verwalteten Knoten mithilfe der Systems Manager Fleet ManagerManager-Konsole oder der Taste AWS Command Line Interface (AWS CLI) zurücksetzen.

So ändern Sie das Passwort auf einem verwalteten Knoten (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem Knoten, der ein neues Passwort benötigt.
4. Wählen Sie Instance-Aktionen, Passwort zurücksetzen.
5. Geben Sie im Feld User name (Benutzername) den Namen des Benutzers ein, für den Sie das Passwort ändern. Dabei kann es sich um jeden Benutzernamen handeln, der ein Konto auf dem Knoten hat.
6. Wählen Sie Absenden aus.
7. Befolgen Sie die Eingabe-Prompts im Befehlsfenster Neues Passwort eingeben, um das neue Passwort anzugeben.

> **Note**
>
> Wenn die Version von SSM Agent auf dem verwalteten Knoten kein Zurücksetzen von Passwörtern unterstützt, werden Sie aufgefordert, mit Run Command, einem Tool in AWS Systems Manager, eine unterstützte Version zu installieren.

So setzen Sie das Passwort auf einem verwalteten Knoten zurück (AWS CLI)

1. Um das Passwort für einen Benutzer auf einem verwalteten Knoten zurückzusetzen, führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

> **Note**
>
> Um das AWS CLI zum Zurücksetzen eines Passworts zu verwenden, muss das Session Manager Plugin auf Ihrem lokalen Computer installiert sein. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugin für die AWS CLI.

Linux & macOS

```
aws ssm start-session \
```

```
    --target instance-id \
    --document-name "AWS-PasswordReset" \
    --parameters '{"username": ["user-name"]}'
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name "AWS-PasswordReset" ^
    --parameters username="user-name"
```

2. Befolgen Sie die Eingabe-Prompts im Befehlsfenster Neues Passwort eingeben, um das neue Passwort anzugeben.

Fehlerbehebung beim Zurücksetzen von Passwörtern auf verwalteten Knoten

Viele Probleme beim Zurücksetzen von Passwörtern können behoben werden, wenn Sie sicherstellen, dass Sie die Voraussetzungen für das Zurücksetzen von Passwörtern gelesen haben. Bei anderen Problemen nehmen Sie die folgenden Informationen zur Behebung von Problemen beim Zurücksetzen von Passwörtern zu Hilfe.

Themen

- Verwalteter Knoten nicht verfügbar
- SSM Agentnicht up-to-date (Konsole)
- Optionen zum Zurücksetzen des Passworts werden nicht bereitgestellt (AWS CLI)
- Keine Berechtigung zum Ausführen von ssm:SendCommand
- Session Manager-Fehlermeldung

Verwalteter Knoten nicht verfügbar

Problem: Sie möchten auf der Seite Managed Instances (Verwaltete Instances) der Konsole das Passwort für einen verwalteten Knoten zurücksetzen, der jedoch nicht in der Liste enthalten ist.

- Solution (Lösung): Der verwaltete Knoten, mit dem Sie eine Verbindung herstellen möchten, wurde möglicherweise nicht für Systems Manager konfiguriert. Um eine EC2 Instance mit Systems Manager zu verwenden, muss der Instance ein AWS Identity and Access Management (IAM-) Instanzprofil angehängt werden, das Systems Manager die Erlaubnis erteilt, Aktionen auf Ihren

Instances durchzuführen. Informationen finden Sie unter Konfiguration von erforderliche Instance-Berechtigungen für Systems Manager.

Um einen EC2 Computer mit Systems Manager zu verwenden, erstellen Sie eine IAM-Servicerolle, die Systems Manager die Berechtigung erteilt, Aktionen auf Ihren verwalteten Knoten auszuführen. Weitere Informationen finden Sie unter Erstellen der für Systems Manager erforderlichen IAM-Servicerolle in Hybrid- und Multicloud-Umgebungen. (Session ManagerUnterstützung für lokale Server und VMs wird nur für die Stufe „Advanced-Instances“ bereitgestellt. Weitere Informationen finden Sie unter.) Aktivieren des Kontingents für erweiterte Instances

SSM Agentnicht up-to-date (Konsole)

Problem: Es wird eine Meldung angezeigt, dass die Version von SSM Agent die Funktionalität zum Zurücksetzen von Passwörtern nicht unterstützt.

- Lösung: Es ist Version 2.3.668.0 oder höher von SSM Agent erforderlich, um Passwörter zurückzusetzen. In der Konsole können Sie den Agent auf dem verwalteten Knoten aktualisieren, indem Sie Update SSM Agent (aktualisieren) auswählen.

  Wenn Systems Manager neue Tools hinzugefügt oder Aktualisierungen an den vorhandenen Tools vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Tools und Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent in Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die SSM Agent-Versionshinweise-Seite in GitHub, um Benachrichtigungen über SSM Agent-Updates zu erhalten.

Optionen zum Zurücksetzen des Passworts werden nicht bereitgestellt (AWS CLI)

Problem: Sie haben mit dem AWS CLI `start-session` Befehl erfolgreich eine Verbindung zu einem verwalteten Knoten hergestellt. Sie haben das SSM-Dokument `AWS-PasswordReset` und einen gültigen Benutzernamen angegeben, aber die Eingabeaufforderungen zur Änderung des Passworts werden nicht angezeigt.

- Lösung: Die Version von SSM Agent auf dem verwalteten Knoten ist es nicht up-to-date. Es ist Version 2.3.668.0 oder höher erforderlich, um das Passwort zurückzusetzen.

Wenn Systems Manager neue Tools hinzugefügt oder Aktualisierungen an den vorhandenen Tools vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Tools und Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent in Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die SSM Agent-Versionshinweise-Seite in GitHub, um Benachrichtigungen über SSM Agent-Updates zu erhalten.

Keine Berechtigung zum Ausführen von **`ssm:SendCommand`**

Problem: Sie versuchen, eine Verbindung zu einem verwalteten Knoten herzustellen, um das Passwort zu ändern, aber es wird eine Fehlermeldung angezeigt, dass Sie nicht autorisiert sind, `ssm:SendCommand` auf dem verwalteten Knoten auszuführen.

- Lösung: Ihre IAM-Richtlinie muss die Berechtigung zum Ausführen des `ssm:SendCommand`-Befehls enthalten. Weitere Informationen finden Sie unter Den Zugriff von Run Command anhand von Tags beschränken.

Session Manager-Fehlermeldung

Problem: Sie erhalten eine Fehlermeldung zu Session Manager.

- Lösung: Die Unterstützung für das Zurücksetzen von Passwörtern erfordert, dass Session Manager korrekt konfiguriert ist. Weitere Informationen finden Sie unter Einrichten von Session Manager und Fehlerbehebung für Session Manager.

Aufheben der Registrierung von verwalteten Knoten in einer Hybrid- und Multi-Cloud-Umgebung

Wenn Sie einen lokalen Server, ein Edge-Gerät oder eine virtuelle Maschine (VM) nicht mehr mithilfe von verwenden möchten AWS Systems Manager, können Sie die Registrierung aufheben. Wenn Sie einen hybridaktivierten Knoten deregistrieren, wird er aus der Liste der verwalteten Knoten in Systems Manager entfernt. AWS Systems Manager Der Agent (SSM Agent) der auf dem hybrid-aktivierten Knoten läuft, kann sein Autorisierungs-Token nicht mehr aktualisieren, da es nicht mehr registriert ist. SSM Agent wird in den Ruhezustand versetzt und reduziert seine Ping-Frequenz auf Systems Manager in der Cloud auf einmal pro Stunde. Systems Manager speichert den Befehlsverlauf für einen verwaltete Knoten, deren Registrierung aufgehoben wurde, 30 Tage lang.

> **Note**
>
> Sie können einen lokalen Server, ein Edge-Gerät oder eine VM mit demselben Aktivierungscode und derselben ID erneut registrieren, solange Sie das Instance-Limit für den angegebenen Aktivierungscode und die angegebene ID nicht erreicht haben. Sie können das Instance-Limit in der Konsole überprüfen, indem Sie Node-Tools und dann Hybrid-Aktivierungen auswählen. Wenn der Wert für Registrierte Instances unter dem Registrierungslimit liegt, können Sie eine Maschine mit demselben Aktivierungscode und derselben ID erneut registrieren. Wenn der Wert über dem Limit liegt, müssen Sie einen anderen Aktivierungscode und eine andere ID verwenden.

Im folgenden Verfahren wird beschrieben, wie Sie die Registrierung für einen hybrid-aktivierten Knoten mithilfe der Systems-Manager-Konsole aufheben. Informationen dazu, wie Sie dies mithilfe von tun können AWS Command Line Interface, finden Sie unter deregister-managed-instance.

Verwandte Informationen finden Sie in den folgenden Themen:

- Abmelden und Neuregistrieren eines verwalteten Knotens (Linux) (Linux)
- Abmelden und Neuregistrieren eines verwalteten Knotens (Windows Server) (Windows Server)

Um die Registrierung eines hybrid-aktivierten Knotens aufzuheben (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Aktivieren Sie das Kontrollkästchen neben dem verwalteten Knoten, den Sie abmelden möchten.
4. Wählen Sie „Knotenaktionen“, „Tools“, „Diesen verwalteten Knoten abmelden“.
5. Überprüfen Sie die Informationen im Dialogfeld Diesen verwalteten Knoten deregistrieren. Wenn Sie zustimmen, wählen Sie Deregister aus.

Arbeiten mit Betriebssystemdateisystemen mithilfe von Fleet Manager

Sie können Fleet Manager, ein Tool in AWS Systems Manager, verwenden, um mit dem Dateisystem auf Ihren verwalteten Knoten zu arbeiten. Mit Fleet Manager können Sie Informationen über das Verzeichnis und die Dateidaten anzeigen, die auf den Volumes gespeichert sind, die an

Ihre verwalteten Knoten angefügt sind. Sie können beispielsweise den Namen, die Größe, die Erweiterung, den Besitzer und die Berechtigungen für Ihre Verzeichnisse und Dateien anzeigen. Bis zu 10.000 Zeilen von Dateidaten können als Text in der Fleet Manager-Konsole angezeigt werden. Sie können diese Funktion auch für `tail`-Dateien verwenden. Bei Verwendung von `tail`, um Dateidaten anzuzeigen, werden zunächst die letzten 10 Zeilen der Datei angezeigt. Wenn neue Datenzeilen in die Datei geschrieben werden, wird die Ansicht in Echtzeit aktualisiert. Daher können Sie Protokolldaten von der Konsole aus überprüfen, was die Effizienz Ihrer Fehlerbehebung und Systemverwaltung verbessern kann. Darüber hinaus können Sie Verzeichnisse erstellen und Dateien und Verzeichnisse kopieren, ausschneiden, einfügen, umbenennen oder löschen.

Wir empfehlen Ihnen, regelmäßige Backups zu erstellen oder Snapshots der Amazon Elastic Block Store (Amazon EBS)-Volumes zu erstellen, die an Ihre verwalteten Knoten angefügt sind. Beim Kopieren oder Ausschneiden und Einfügen von Dateien werden vorhandene Dateien und Verzeichnisse im Zielpfad mit dem gleichen Namen wie die neuen Dateien oder Verzeichnisse ersetzt. Ernsthafte Probleme können auftreten, wenn Sie Systemdateien und -Verzeichnisse ersetzen oder ändern. AWS garantiert nicht, dass diese Probleme gelöst werden können. Ändern Sie Systemdateien auf eigenes Risiko. Sie sind für alle Änderungen an Dateien und Verzeichnissen verantwortlich und stellen sicher, dass Sie Backups haben. Das Löschen oder Ersetzen von Dateien und Verzeichnissen kann nicht rückgängig gemacht werden.

> **Note**
>
> Fleet Manager nutzt Session Manager, ein Tool in AWS Systems Manager, um Textvorschau und `tail`-Dateien anzuzeigen. Für Amazon Elastic Compute Cloud (Amazon EC2)-Instances muss das Instance-Profil, das Ihren verwalteten Instances angefügt ist, Berechtigungen für Session Manager bereitstellen, um diese Funktion zu verwenden. Weitere Informationen zum Hinzufügen von Session Manager-Berechtigungen an ein Instance-Profil finden Sie unter Session Manager-Berechtigungen für eine vorhandene IAM-Rolle hinzufügen.

Themen

- Anzeigen des OS-Dateisystems mithilfe von Fleet Manager
- Vorschau von Betriebssystemdateien mit Fleet Manager
- Speichern von Betriebssystemdateien mit Fleet Manager
- Kopieren, Ausschneiden und Einfügen von Betriebssystemdateien oder -verzeichnissen mit Fleet Manager

- Umbenennen von Betriebssystemdateien und -verzeichnissen mit Fleet Manager
- Löschen von Betriebssystemdateien und -verzeichnissen mit Fleet Manager
- Betriebssystemverzeichnisse erstellen mit Fleet Manager
- Ausschneiden, Kopieren und Einfügen von Betriebssystemverzeichnissen mit Fleet Manager

Anzeigen des OS-Dateisystems mithilfe von Fleet Manager

Sie können Fleet Manager verwenden, um das Betriebssystemdateisystem auf einem von Systems Manager verwalteten Knoten anzuzeigen.

Anzeigen des OS-Dateisystems mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung des verwalteten Knotens mit dem Dateisystem, das Sie anzeigen möchten.
4. Wählen Sie Tools, Dateisysteme.

Vorschau von Betriebssystemdateien mit Fleet Manager

Sie können Fleet Manager verwenden, um eine Vorschau von Textdateien auf einem Betriebssystem anzuzeigen.

So zeigen Sie Textvorschauen mit Fleet Manager an

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung des verwalteten Knotens mit den Dateien, die Sie anzeigen möchten.
4. Wählen Sie Tools, Dateisysteme.
5. Wählen Sie den File name (Dateiname) des Verzeichnisses, das die Datei enthält, die Sie in der Vorschau anzeigen möchten.
6. Wählen Sie die Schaltfläche neben der Datei, deren Inhalt Sie in der Vorschau anzeigen möchten.

7. Wählen Sie Aktionen, Vorschau als Text.

Speichern von Betriebssystemdateien mit Fleet Manager

Sie können sie Fleet Manager verwenden, um eine Datei auf einem verwalteten Knoten als Tail zu speichern.

So verfolgen Sie OS-Dateien mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung des verwalteten Knotens mit den Dateien, die Sie verfolgen möchten.
4. Wählen Sie Tools, Dateisysteme.
5. Wählen Sie den File name (Dateiname) des Verzeichnisses, das die Datei enthält, die verfolgen möchten.
6. Wählen Sie die Schaltfläche neben der Datei, deren Inhalt Sie verfolgen möchten.
7. Wählen Sie Aktionen, Enddatei.

Kopieren, Ausschneiden und Einfügen von Betriebssystemdateien oder -verzeichnissen mit Fleet Manager

Sie können Fleet Manager mit Betriebssystemdateien auf einem verwalteten Knoten kopieren, ausschneiden und einfügen.

So kopieren oder fügen Sie Dateien oder Verzeichnisse mit Fleet Manager ein

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung des verwalteten Knotens mit den Dateien, die Sie kopieren oder ausschneiden und einfügen möchten.
4. Wählen Sie Tools, Dateisysteme.
5. Um eine Datei zu kopieren oder auszuschneiden, wählen Sie den File name (Dateiname) des Verzeichnisses, das die Datei enthält, die Sie kopieren oder ausschneiden möchten. Um

ein Verzeichnis zu kopieren oder auszuschneiden, wählen Sie die Schaltfläche neben dem Verzeichnis, das Sie kopieren oder ausschneiden möchten, und fahren Sie dann mit Schritt 8 fort.

6. Wählen Sie die Schaltfläche neben der Datei, die Sie kopieren oder ausschneiden möchten.
7. Wählen Sie im Menü Actions (Aktionen) Copy (Kopieren) oder Cut (Ausschneiden).
8. Wählen Sie in der Ansicht Dateisystem die Schaltfläche neben dem Verzeichnis, in das Sie die Datei einfügen möchten.
9. Wählen Sie im Menü Aktionen Einfügen.

Umbenennen von Betriebssystemdateien und -verzeichnissen mit Fleet Manager

Sie können Fleet Manager verwenden, um Dateien und Verzeichnisse auf einem verwalteten Knoten in Ihrem Konto umzubenennen.

So können Sie Dateien oder Verzeichnisse mit Fleet Manager umbenennen

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung des verwalteten Knotens mit den Dateien oder Verzeichnissen, die Sie umbenennen möchten.
4. Wählen Sie Tools, Dateisysteme.
5. Um eine Datei umzubenennen, wählen Sie den File name (Dateiname) des Verzeichnisses, das die Datei enthält, die Sie umbenennen möchten. Um ein Verzeichnis umzubenennen, wählen Sie die Schaltfläche neben dem Verzeichnis, das Sie umbenennen möchten, und fahren Sie dann mit Schritt 8 fort.
6. Wählen Sie die Schaltfläche neben der Datei, deren Inhalt Sie umbenennen möchten.
7. Wählen Sie Aktionen, Umbenennen.
8. Geben Sie im Feld Dateiname den neuen Namen für die Datei ein und wählen Sie Umbenennen.

Löschen von Betriebssystemdateien und -verzeichnissen mit Fleet Manager

Sie können Fleet Manager verwenden, um Dateien und Verzeichnisse auf einem verwalteten Knoten in Ihrem Konto zu löschen.

So löschen Sie Dateien oder Verzeichnisse mithilfe von Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung des verwalteten Knotens mit den Dateien oder Verzeichnissen, die Sie löschen möchten.
4. Wählen Sie Tools, Dateisysteme.
5. Um eine Datei zu löschen, wählen Sie File name (Dateiname) des Verzeichnisses, das die Datei enthält, die Sie löschen möchten. Um ein Verzeichnis zu löschen, wählen Sie die Schaltfläche neben dem Verzeichnis, das Sie löschen möchten, und fahren Sie dann mit Schritt 7 fort.
6. Wählen Sie die Schaltfläche neben der Datei, deren Inhalt Sie löschen möchten.
7. Wählen Sie Aktionen, Löschen.

Betriebssystemverzeichnisse erstellen mit Fleet Manager

Sie können Fleet Manager verwenden, um Dateien und Verzeichnisse auf einem verwalteten Knoten in Ihrem Konto zu erstellen.

Erstellen eines Verzeichnisses mithilfe von Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung des verwalteten Knotens, in dem Sie ein Verzeichnis erstellen möchten.
4. Wählen Sie Tools, Dateisysteme.
5. Wählen Sie den File name (Dateiname) des Verzeichnisses, in dem Sie ein neues Verzeichnis erstellen möchten.
6. Wählen Sie Create directory (Verzeichnis erstellen).
7. Geben Sie im Feld Verzeichnisname den Namen für das neue Verzeichnis ein und wählen Sie Verzeichnis erstellen.

Ausschneiden, Kopieren und Einfügen von Betriebssystemverzeichnissen mit Fleet Manager

Sie können Fleet Manager verwenden, um Verzeichnisse auf einem verwalteten Knoten in Ihrem Konto auszuschneiden, zu kopieren und einzufügen.

So kopieren oder fügen Sie Verzeichnisse mit Fleet Manager ein

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung des verwalteten Knotens mit den Dateien, die Sie kopieren oder ausschneiden und einfügen möchten.
4. Wählen Sie Tools, Dateisysteme.
5. Wählen Sie die Schaltfläche neben dem Verzeichnis, das Sie kopieren oder ausschneiden möchten, und fahren Sie dann mit Schritt 8 fort.
6. Wählen Sie im Menü Aktionen Kopieren oder Ausschneiden.
7. Wählen Sie in der Ansicht Dateisystem die Schaltfläche neben dem Verzeichnis, in das Sie die Datei einfügen möchten.
8. Wählen Sie im Menü Aktionen Einfügen.

Überwachung der Leistung verwalteter Knoten

Sie können Fleet Manager verwenden, ein Tool in AWS Systems Manager, um Leistungsdaten für Ihre verwalteten Knoten in Echtzeit anzuzeigen. Die Leistungsdaten werden von Leistungsindikatoren abgerufen.

Die folgenden Leistungsindikatoren sind in Fleet Manager verfügbar:

- CPU-Nutzung
- Festplatten-Input/Output-(I/O)-Auslastung
- Netzwerkdatenverkehr
- Speicherauslastung

> **Note**
>
> Fleet Manager verwendet Session Manager, ein Tool in AWS Systems Manager, um Leistungsdaten abzurufen. Für Amazon Elastic Compute Cloud (Amazon EC2)-Instances muss das Instance-Profil, das Ihren verwalteten Instances angefügt ist, Berechtigungen für Session Manager bereitstellen, um diese Funktion zu verwenden. Weitere Informationen zum Hinzufügen von Session Manager-Berechtigungen an ein Instance-Profil finden Sie unter Session Manager-Berechtigungen für eine vorhandene IAM-Rolle hinzufügen.

Anzeigen von Leistungsdaten mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, dessen Leistung Sie überwachen möchten.
4. Wählen Sie die Option Details anzeigen aus.
5. Wählen Sie Tools, Leistungsindikatoren.

Arbeiten mit Prozessen

Sie können Fleet Manager verwenden, ein Tool in AWS Systems Manager, um mit Prozessen auf Ihren verwalteten Knoten zu arbeiten. Mit Fleet Manager können Sie Informationen über Prozesse anzeigen. Beispielsweise können Sie zusätzlich zu ihren Handles und Threads die CPU-Auslastung und Speicherauslastung von Prozessen sehen. Mit Fleet Manager können Sie Prozesse von der Konsole aus starten und beenden.

> **Note**
>
> Fleet Manager verwendet Session Manager, ein Tool in AWS Systems Manager, um Prozessdaten abzurufen. Für Amazon Elastic Compute Cloud (Amazon EC2)-Instances muss das Instance-Profil, das Ihren verwalteten Instances angefügt ist, Berechtigungen für Session Manager bereitstellen, um diese Funktion zu verwenden. Weitere Informationen zum Hinzufügen von Session Manager-Berechtigungen an ein Instance-Profil finden Sie unter Session Manager-Berechtigungen für eine vorhandene IAM-Rolle hinzufügen.

Themen

- Anzeigen von Details zu Betriebssystemprozessen mithilfe von Fleet Manager
- Starten eines Betriebssystemprozesses auf einem verwalteten Knoten mit Fleet Manager
- Beenden eines Betriebssystemprozesses mit Fleet Manager

Anzeigen von Details zu Betriebssystemprozessen mithilfe von Fleet Manager

Sie können Fleet Manager nutzen, um Details zu Prozessen auf Ihren verwalteten Knoten anzuzeigen.

So zeigen Sie Details zu Prozessen mit Fleet Manager an

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung der Knoten aus, deren Prozesse Sie anzeigen möchten.
4. Wählen Sie Tools, Prozesse.

Starten eines Betriebssystemprozesses auf einem verwalteten Knoten mit Fleet Manager

Sie können Fleet Manager verwenden, um einen Prozess auf einem verwalteten Knoten zu starten.

So starten Sie einen Prozess mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung des verwalteten Knotens aus, auf dem Sie einen Prozess starten möchten.
4. Wählen Sie Tools, Prozesse.
5. Wählen Sie Start new process (Neuen Prozess starten).
6. Geben Sie im Feld Prozessname oder vollständiger Pfad den Namen des Prozesses oder den vollständigen Pfad zur ausführbaren Datei ein.
7. (Optional) Geben Sie im Feld Arbeitsverzeichnis den Verzeichnispfad ein, in dem der Prozess ausgeführt werden soll.

Beenden eines Betriebssystemprozesses mit Fleet Manager

So beenden Sie einen Betriebssystemprozess mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Verknüpfung des verwalteten Knotens aus, auf dem Sie einen Prozess starten möchten.
4. Wählen Sie Tools, Prozesse.
5. Wählen Sie die Schaltfläche neben dem Prozess, den Sie beenden möchten.
6. Wählen Sie Aktionen, Prozess beenden oder Aktionen, Prozessstruktur beenden.

> **Note**
>
> Durch das Beenden eines Prozessbaums werden auch alle Prozesse und Anwendungen beendet, die diesen Prozess verwenden.

Protokolle auf verwalteten Knoten anzeigen

Sie können Fleet Manager verwenden, ein Tool in AWS Systems Manager, um Protokolldaten anzuzeigen, die auf Ihren verwalteten Knoten gespeichert sind. Bei Windows-verwalteten Knoten können Sie aus der Konsole Windows-Ereignisprotokolle anzeigen und ihre Details kopieren. Um Ihnen bei der Suche nach Ereignissen zu helfen, filtern Sie Windows-Ereignisprotokolle nach Event level (Event-Ebene), Event ID (Ereignis-ID), Event source (Ereignisquelle) und Time created (Erstellungszeitpunkt). Sie können auch andere Protokolldaten mit dem Verfahren zum Anzeigen des Dateisystemen anzeigen. Weitere Informationen zum Anzeigen des Dateisystems mit Fleet Manager finden Sie unter Arbeiten mit Betriebssystemdateisystemen mithilfe von Fleet Manager.

Anzeigen von Windows-Ereignisprotokolle mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, dessen Ereignisprotokolle Sie anzeigen möchten.

4. Wählen Sie die Option Details anzeigen aus.
5. Wählen Sie Tools, Windows-Ereignisprotokolle.
6. Wählen Sie den Log name (Protokollnamen), welcher die Ereignisse enthält, die Sie anzeigen möchten.
7. Wählen Sie die Schaltfläche neben dem Log name (Protokollnamen), den Sie anzeigen möchten und wählen Sie dann View events (Anzeigen von Ereignissen).
8. Wählen Sie die Schaltfläche neben dem Ereignis, das Sie anzeigen möchten und wählen Sie dann View event details (Eventdetails anzeigen).
9. (Optional) Wählen Sie Copy as JSON (Copy as JSON), um die Ereignisdetails in die Zwischenablage zu kopieren.

Verwalten von Betriebssystembenutzerkonten und Gruppen auf verwalteten Knoten mithilfe von Fleet Manager

Sie könnenFleet Manager, ein Tool in AWS Systems Manager, verwenden, um Benutzerkonten und Gruppen des Betriebssystems (OS) auf Ihren verwalteten Knoten zu verwalten. Sie können beispielsweise Benutzer und Gruppen erstellen und löschen. Darüber hinaus können Sie Details wie Gruppenmitgliedschaft, Benutzerrollen und Status anzeigen.

> **Important**
>
> Fleet Managerverwendet Run Command undSession Manager, Tools in AWS Systems Manager, für verschiedene Benutzerverwaltungsvorgänge. Daher könnte ein Benutzer einem Betriebssystembenutzerkonto Berechtigungen erteilen, die andernfalls nicht möglich wären. Das liegt daran, dass AWS Systems Manager Agent (SSM Agent) auf Amazon Elastic Compute Cloud (Amazon EC2) -Instances mit Root-Rechten (Linux) oder SYSTEM-Berechtigungen (Windows Server) ausgeführt wird. Weitere Informationen zum Einschränken des Zugriffs auf Befehle auf Root-Ebene über SSM Agent finden Sie unter Einschränken des Zugriffs auf Befehle auf Stammebene durch SSM Agent. Um den Zugriff auf diese Funktion einzuschränken, empfehlen wir, AWS Identity and Access Management (IAM-) Richtlinien für Ihre Benutzer zu erstellen, die nur Zugriff auf die von Ihnen definierten Aktionen gewähren. Weitere Informationen zum Erstellen von IAM-Richtlinien für Fleet Manager finden Sie unter Steuern des Zugriffs auf Fleet Manager.

Themen

- Einen Betriebssystembenutzer oder eine Betriebssystemgruppe erstellen mit Fleet Manager
- Benutzer- oder Gruppenmitgliedschaft aktualisieren mit Fleet Manager
- Löschen eines Betriebssystembenutzers oder einer Betriebssystemgruppe mit Fleet Manager

Einen Betriebssystembenutzer oder eine Betriebssystemgruppe erstellen mit Fleet Manager

> **Note**
>
> Fleet Manager nutzt Session Manager, um Kennwörter für neue Benutzer festzulegen. Für Amazon-EC2-Instances muss das Instance-Profil, das Ihren verwalteten Instances angefügt ist, Berechtigungen für Session Manager bereitstellen, um dieses Feature zu verwenden. Weitere Informationen zum Hinzufügen von Session Manager-Berechtigungen an ein Instance-Profil finden Sie unter Session Manager-Berechtigungen für eine vorhandene IAM-Rolle hinzufügen.

Anstatt sich direkt bei einem Server anzumelden, um ein Benutzerkonto oder eine Gruppe zu erstellen, können Sie die Fleet Manager-Konsole verwenden, um dieselben Aufgaben auszuführen.

So erstellen Sie ein BS-Benutzerkonto mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem Sie einen neuen Benutzer erstellen möchten.
4. Wählen Sie die Option Details anzeigen aus.
5. Wählen Sie Tools, Benutzer und Gruppen.
6. Wählen Sie die Registerkarte Users und anschließend die Option Create user (Benutzer erstellen).
7. Geben Sie einen Wert für den Namen des neuen Benutzers an.
8. (Empfohlen) Aktivieren Sie das Kontrollkästchen neben Set password (Passwort festlegen). Am Ende des Verfahrens werden Sie aufgefordert, ein Passwort für den neuen Benutzer einzugeben.

9. Wählen Sie Create user (Benutzer erstellen). Wenn Sie das Kontrollkästchen zum Erstellen eines Kennworts für den neuen Benutzer aktiviert haben, werden Sie aufgefordert, einen Wert für das Kennwort einzugeben und Done (Fertig) zu wählen. Wenn das angegebene Passwort die Anforderungen der lokalen oder Domain-Richtlinien Ihres verwalteten Knotens nicht erfüllt, wird ein Fehler zurückgegeben.

So erstellen Sie eine OS-Gruppe mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem Sie eine Gruppe erstellen möchten.
4. Wählen Sie die Option Details anzeigen aus.
5. Wählen Sie Tools, Benutzer und Gruppen.
6. Wählen Sie die Registerkarte Groups (Gruppen) und dann Create group (Gruppe erstellen) aus.
7. Geben Sie einen Wert für den Namen der neuen Gruppe an.
8. (Optional) Geben Sie einen Wert für die Description (Beschreibung) der neuen Gruppe ein.
9. (Optional) Wählen Sie die Benutzer aus, die zu den Group members (Gruppenmitgliedern) hinzugefügt werden sollen.
10. Wählen Sie Create group (Gruppe erstellen).

Benutzer- oder Gruppenmitgliedschaft aktualisieren mit Fleet Manager

Anstatt sich direkt bei einem Server anzumelden, um ein Benutzerkonto oder eine Gruppe zu aktualisieren, können Sie die Fleet Manager-Konsole verwenden, um dieselben Aufgaben auszuführen.

So fügen Sie ein Betriebssystem-Benutzerkonto zu einer neuen Gruppe mit Fleet Manager hinzu

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem sich das Benutzerkonto befindet, das Sie aktualisieren möchten.

4. Wählen Sie die Option Details anzeigen aus.
5. Wählen Sie Tools, Benutzer und Gruppen.
6. Wählen Sie die Registerkarte Users.
7. Wählen Sie die Schaltfläche neben dem Benutzer, den Sie aktualisieren möchten.
8. Wählen Sie Aktionen, Benutzer zu Gruppe hinzufügen.
9. Wählen Sie unter Add to group (Zur Gruppe hinzufügen) die Gruppe aus, zu der Sie den Benutzer hinzufügen möchten.
10. Wählen Sie Add user to group (Benutzer zur Gruppe hinzufügen).

So bearbeiten Sie die Mitgliedschaft einer Betriebssystemgruppe mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem sich die Gruppe befindet, die Sie aktualisieren möchten.
4. Wählen Sie die Option Details anzeigen aus.
5. Wählen Sie Tools, Benutzer und Gruppen.
6. Wählen Sie die Registerkarte Groups (Gruppen).
7. Wählen Sie die Schaltfläche neben der Gruppe, die Sie aktualisieren möchten.
8. Wählen Sie Aktionen, Gruppe ändern.
9. Wählen Sie unter Gruppenmitglieder die Benutzer aus, die Sie hinzufügen oder entfernen möchten.
10. Wählen Sie Modify group (Gruppe ändern).

Löschen eines Betriebssystembenutzers oder einer Betriebssystemgruppe mit Fleet Manager

Anstatt sich direkt bei einem Server anzumelden, um ein Benutzerkonto oder eine Gruppe zu löschen, können Sie die Fleet Manager-Konsole verwenden, um dieselben Aufgaben auszuführen.

So löschen Sie Betriebssystem-Benutzerkonten mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem sich das Benutzerkonto befindet, das Sie löschen möchten.
4. Wählen Sie die Option Details anzeigen aus.
5. Wählen Sie Tools, Benutzer und Gruppen.
6. Wählen Sie die Registerkarte Users.
7. Wählen Sie die Schaltfläche neben dem Benutzer, dem Sie löschen möchten.
8. Wählen Sie Aktionen, Lokalen Benutzer löschen.

So löschen Sie eine Betriebssystemgruppe mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem sich die Gruppe befindet, die Sie löschen möchten.
4. Wählen Sie die Option Details anzeigen aus.
5. Wählen Sie Tools, Benutzer und Gruppen.
6. Wählen Sie die Registerkarte Group (Gruppe).
7. Wählen Sie die Schaltfläche neben der Gruppe, die Sie aktualisieren möchten.
8. Wählen Sie Aktionen, Lokale Gruppe löschen.

Verwalten der Windows-Registrierung auf verwalteten Knoten

Sie können Fleet Manager verwenden, ein Tool in AWS Systems Manager, um die Registrierung auf Ihren verwalteten Windows Server-Knoten zu verwalten. Von der Fleet Manager-Konsole können Sie Registrierungseinträge und -werte erstellen, kopieren, aktualisieren und löschen.

> **Important**
>
> Wir empfehlen, ein Backup der Registry oder einen Snapshot des Amazon Elastic Block Store (Amazon EBS)-Root-Volume zu erstellen, das Ihrem verwalteten Knoten angefügt ist, bevor Sie die Registry ändern. Schwerwiegende Probleme können auftreten, wenn Sie eine falsche Änderung in der Registrierung vornehmen. Für diese Probleme müssen

Sie möglicherweise das Betriebssystem neu installieren oder das Root-Volume Ihres Knotens von einem Snapshot wiederherstellen. AWS garantiert nicht, dass diese Probleme gelöst werden können. Ändern Sie die Registrierung auf eigenes Risiko. Sie sind für alle Registrierungsänderungen verantwortlich und stellen sicher, dass Sie Backups haben.

Erstellen eines Windows-Registrierungsschlüssels oder -Eintrags

Erstellen eines Windows-Registrierungsschlüssel mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem Sie einen Registry-Schlüssel erstellen möchten.
4. Wählen Sie die Option Details anzeigen aus.
5. Wählen Sie Tools, Windows Registry.
6. Wählen Sie den Hive aus, in dem Sie einen neuen Registrierungsschlüssel erstellen möchten, indem Sie den Registry name (Registry-Name) wählen.
7. Wählen Sie Registrierungsschlüssel erstellen.
8. Wählen Sie die Schaltfläche neben dem Registrierungseintrag, in dem Sie einen Schlüssel erstellen möchten.
9. Wählen Sie Create registry key (Registry-Schlüssel erstellen).
10. Geben Sie einen Wert für den Namen des neuen Registrierungsschlüssels ein und wählen Sie Submit (Senden)

Erstellen eines Windows-Registrierungseintrags mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben der Instance, in der Sie einen Registrierungseintrag erstellen möchten.
4. Wählen Sie die Option Details anzeigen aus.
5. Wählen Sie Tools, Windows Registry.

6. Wählen Sie den Hive und den darauffolgenden Registrierungsschlüssel aus, in dem Sie einen neuen Registrierungseintrag erstellen möchten, indem Sie den Registry name (Registry-Name) wählen.
7. Wählen Sie Erstellen, Registrierungseintrag erstellen.
8. Geben Sie einen Wert für den Namen des neuen Registrierungseintrags ein.
9. Wählen Sie den Typ des Wertes, den Sie für den Registrierungseintrag erstellen möchten. Weitere Informationen zu Registry-Werttypen finden Sie unter Registry value types (Registry-Wertetypen).
10. Geben Sie einen Wert für den Value (Wert) des neuen Registrierungseintrags ein.

Aktualisieren eines Windows-Registrierungseintrags

Aktualisieren eines Windows-Registrierungseintrags mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem Sie einen Registry-Eintrag aktualisieren möchten.
4. Wählen Sie die Option Details anzeigen aus.
5. Wählen Sie Tools, Windows Registry.
6. Wählen Sie den Hive und den nachfolgenden Registrierungsschlüssel aus, den Sie aktualisieren möchten, indem Sie das Kontrollkästchen Registry name (Registry-Name) anklicken.
7. Wählen Sie die Schaltfläche neben dem Registrierungseintrag aus, den Sie aktualisieren möchten.
8. Wählen Sie Aktionen, Registrierungseintrag aktualisieren.
9. Geben Sie den neuen Wert für den Value (Wert) des Registrierungseintrags ein.
10. Wählen Sie Aktualisieren.

Löschen eines Windows-Registrierungseintrags oder -schlüssels

Erstellen eines Windows-Registrierungsschlüssel mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem Sie einen Registry-Schlüssel löschen möchten.
4. Wählen Sie Tools, Windows Registry.
5. Wählen Sie den Hive und den nachfolgenden Registrierungsschlüssel aus, den Sie löschen möchten, indem Sie das Kontrollkästchen Registry name (Registry-Name) anklicken.
6. Wählen Sie die Schaltfläche neben dem Registrierungsschlüssel, den Sie löschen möchten
7. Wählen Sie Aktionen, Registrierungsschlüssel löschen.

Löschen eines Windows-Registrierungseintrags mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben dem verwalteten Knoten, auf dem Sie einen Registry-Eintrag löschen möchten.
4. Wählen Sie die Option Details anzeigen aus.
5. Wählen Sie Tools, Windows Registry.
6. Wählen Sie den Hive und den nachfolgenden Registrierungsschlüssel, der den Eintrag enthält, aus, den Sie löschen möchten, indem Sie das Kontrollkästchen Registry name (Registry-Name) anklicken.
7. Wählen Sie die Schaltfläche neben dem Registrierungseintrag, den Sie löschen möchten
8. Wählen Sie Aktionen, Registrierungseintrag löschen.

Automatisches Verwalten von EC2 Instanzen mit der Standard-Host-Management-Konfiguration

Mit der Einstellung Standard-Host-Management-Konfiguration können AWS Systems Manager Sie Ihre EC2 Amazon-Instances automatisch als verwaltete Instances verwalten. Eine verwaltete Instanz ist eine EC2 Instanz, die für die Verwendung mit Systems Manager konfiguriert ist.

Die Verwaltung Ihrer Instances mit Systems Manager bietet unter anderem folgende Vorteile:

- Stellen Sie eine sichere Connect zu Ihren EC2 Instances her mitSession Manager.

- Führen Sie automatisierte Patch-Scans mit Patch Manager durch.
- Zeigen Sie mit Systems Manager Inventory detaillierte Informationen zu Ihren Instances an.
- Verfolgen und verwalten Sie Instances mithilfe von Fleet Manager.
- Halten Sie SSM Agent automatisch auf dem neuesten Stand.

Fleet Manager, Inventory, Patch Manager und Session Manager sind Tools in Systems Manager.

Mit der Standard-Host-Management-Konfiguration können Sie EC2 Instanzen verwalten, ohne manuell ein AWS Identity and Access Management (IAM-) Instanzprofil erstellen zu müssen. Stattdessen erstellt und wendet die Standard-Host-Management-Konfiguration eine Standard-IAM-Rolle an, um sicherzustellen, dass Systems Manager über Berechtigungen zur Verwaltung aller Instances in der AWS-Konto und an der AWS-Region Stelle verfügt, an der sie aktiviert ist.

Wenn die bereitgestellten Berechtigungen für Ihren Anwendungsfall nicht ausreichen, können Sie auch Richtlinien zur Standard-IAM-Rolle hinzufügen, die von der Standardkonfiguration für die Host-Verwaltung erstellt wird. Wenn Sie keine Berechtigungen für alle Funktionen benötigen, die von der Standard-IAM-Rolle bereitgestellt werden, können Sie alternativ Ihre eigene benutzerdefinierte Rolle und Richtlinien erstellen. Alle Änderungen an der IAM-Rolle, die Sie für die Standard-Host-Management-Konfiguration auswählen, gelten für alle verwalteten EC2 Amazon-Instances in der Region und im Konto.

Weitere Informationen zu der Richtlinie, die von der Standardkonfiguration für die Host-Verwaltung verwendet wird, finden Sie unter AWS verwaltete Richtlinie: Amazon SSMManaged EC2 InstanceDefaultPolicy.

Implementieren des Zugriffs mit geringsten Berechtigungen

Die in diesem Thema beschriebenen Verfahren sollten nur von Administratoren durchgeführt werden. Daher empfehlen wir, Zugriff mit den geringsten Berechtigungen zu implementieren, um zu verhindern, dass nichtadministrative Benutzer die Standardkonfiguration für die Host-Verwaltung konfigurieren oder ändern. Beispielrichtlinien, die den Zugriff auf die Standardkonfiguration für die Host-Verwaltung einschränken, finden Sie unter Beispiele für Richtlinien mit den geringsten Berechtigungen für die Standardkonfiguration für die Host-Verwaltung weiter unten in diesem Thema.

> **Important**
>
> Registrierungsinformationen für Instances, die mit der Standardkonfiguration für die Host-Verwaltung registriert wurden, speichern Registrierungsinformationen lokal in den

Verzeichnissen `var/lib/amazon/ssm` oder `C:\ProgramData\Amazon`. Das Entfernen dieser Verzeichnisse oder der enthaltenen Dateien verhindert, dass die Instance die erforderlichen Anmeldeinformationen für die Verbindung mit Systems Manager über die Standardkonfiguration für die Host-Verwaltung erhält. In diesen Fällen müssen Sie ein Instance-Profil verwenden, um Ihrer IAM-Instance die erforderlichen Berechtigungen zu erteilen, oder die Instance neu erstellen.

Themen

- Voraussetzungen
- Die Umgebung der Standardkonfiguration für die Host-Verwaltung aktivieren
- Die Umgebung der Standardkonfiguration für die Host-Verwaltung deaktivieren
- Beispiele für Richtlinien mit den geringsten Berechtigungen für die Standardkonfiguration für die Host-Verwaltung

Voraussetzungen

Um die Standard-Host-Management-Konfiguration in der AWS-Region und an der AWS-Konto Stelle zu verwenden, an der Sie die Einstellung aktivieren, müssen die folgenden Anforderungen erfüllt sein.

- Eine zu verwaltende Instanz muss den Instanz-Metadatendienst Version 2 (IMDSv2) verwenden.

  Die Standardkonfiguration für die Host-Verwaltung unterstützt die Instance-Metadaten-Service-Version 1 nicht. Informationen zur Umstellung auf IMDSv2 Version 2 finden Sie unter Umstellung auf die Nutzung von Instance Metadata Service Version 2 im EC2 Amazon-Benutzerhandbuch
- SSM Agent-Version 3.2.582.0 oder höher muss auf der zu verwaltenden Instance installiert sein.

  Informationen zur Überprüfung der auf Ihrer Instance installierten Version von SSM Agent finden Sie unter Überprüfen der SSM Agent-Versionsnummer.

  Weitere Informationen zur Aktualisierung von SSM Agent finden Sie unter Automatische Aktualisierung von SSM Agent.
- Sie als Administrator, der die Aufgaben in diesem Thema ausführt, benötigen Berechtigungen für die UpdateServiceSettingAPI-Operationen GetServiceSettingResetServiceSetting, und. Darüber hinaus müssen Sie über Berechtigungen für die `iam:PassRole`-Berechtigung für die `AWSSystemsManagerDefaultEC2InstanceManagementRole`-IAM-Rolle verfügen. Im

Folgenden finden Sie ein Beispiel für eine Richtlinie, die diese Berechtigungen vorsieht. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting",
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:servicesetting/ssm/
managed-instance/default-ec2-instance-management-role"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/service-role/
AWSSystemsManagerDefaultEC2InstanceManagementRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ssm.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

- Wenn ein IAM-Instanzprofil bereits an eine EC2 Instanz angehängt ist, die mit Systems Manager verwaltet werden soll, müssen Sie der Instanz alle Berechtigungen entziehen, die den `ssm:UpdateInstanceInformation` Vorgang zulassen. SSM Agentversucht, Instanzprofilberechtigungen zu verwenden, bevor die Standardberechtigungen für die Host-Management-Konfiguration verwendet werden. Wenn Sie die `ssm:UpdateInstanceInformation`-Operation in Ihrem eigenen IAM-Instance-Profil zulassen,

wird die Instance die Berechtigungen der Standardkonfiguration für die Host-Verwaltung nicht verwenden.

Die Umgebung der Standardkonfiguration für die Host-Verwaltung aktivieren

Sie können die Standard-Host-Management-Konfiguration von der Fleet Manager Konsole aus oder mithilfe von AWS Command Line Interface oder aktivieren AWS Tools for Windows PowerShell.

Sie müssen die Standard-Host-Management-Konfiguration nacheinander in jeder Region aktivieren, in der Sie Ihre EC2 Amazon-Instances mit dieser Einstellung verwalten möchten.

Nachdem Sie die Standardkonfiguration für die Hostverwaltung aktiviert haben, kann es bis zu 30 Minuten dauern, bis Ihre Instances die Anmeldeinformationen der Rolle verwenden, die Sie im folgenden Verfahren in Schritt 5 ausgewählt haben.

So aktivieren Sie die Standardkonfiguration für die Host-Verwaltung (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie im Kontoverwaltung, Standardkonfiguration für die Host-Verwaltung konfigurieren.
4. Aktivieren Sie Standardkonfiguration für die Host-Verwaltung aktivieren.
5. Wählen Sie die AWS Identity and Access Management (IAM) -Rolle aus, die verwendet wird, um die Systems Manager Manager-Tools für Ihre Instances zu aktivieren. Wir empfehlen die Verwendung der Standardrolle, die in der Standardkonfiguration für die Host-Verwaltung bereitgestellt wird. Es enthält die Mindestberechtigungen, die für die Verwaltung Ihrer EC2 Amazon-Instances mit Systems Manager erforderlich sind. Wenn Sie es vorziehen, eine benutzerdefinierte Rolle zu verwenden, muss die Vertrauensrichtlinie der Rolle Systems Manager als vertrauenswürdige Entität zulassen.
6. Wählen Sie Konfigurieren, um die Einrichtung abzuschließen.

So aktivieren Sie die Standardkonfiguration für die Host-Verwaltung (Befehlszeile)

1. Erstellen Sie auf Ihrem lokalen Computer eine JSON-Datei, die die folgende Vertrauensbeziehungsrichtlinie enthält.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"",
            "Effect":"Allow",
            "Principal":{
                "Service":"ssm.amazonaws.com"
            },
            "Action":"sts:AssumeRole"
        }
    ]
}
```

2. Öffnen Sie AWS CLI oder Tools für Windows PowerShell und führen Sie je nach Betriebssystemtyp Ihres lokalen Computers einen der folgenden Befehle aus, um eine Servicerolle in Ihrem Konto zu erstellen. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   Linux & macOS

   ```
   aws iam create-role \
   --role-name AWSSystemsManagerDefaultEC2InstanceManagementRole \
   --path /service-role/ \
   --assume-role-policy-document file://trust-policy.json
   ```

   Windows

   ```
   aws iam create-role ^
   --role-name AWSSystemsManagerDefaultEC2InstanceManagementRole ^
   --path /service-role/ ^
   --assume-role-policy-document file://trust-policy.json
   ```

   PowerShell

   ```
   New-IAMRole `
   -RoleName "AWSSystemsManagerDefaultEC2InstanceManagementRole" `
   -Path "/service-role/" `
   ```

```
-AssumeRolePolicyDocument "file://trust-policy.json"
```

3. Führen Sie den folgenden Befehl aus, um Ihrer neu erstellten Rolle die von `AmazonSSMManagedEC2InstanceDefaultPolicy` verwaltete Richtlinie anzufügen. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   Linux & macOS

   ```
   aws iam attach-role-policy \
   --policy-arn arn:aws:iam::aws:policy/AmazonSSMManagedEC2InstanceDefaultPolicy \
   --role-name AWSSystemsManagerDefaultEC2InstanceManagementRole
   ```

   Windows

   ```
   aws iam attach-role-policy ^
   --policy-arn arn:aws:iam::aws:policy/AmazonSSMManagedEC2InstanceDefaultPolicy ^
   --role-name AWSSystemsManagerDefaultEC2InstanceManagementRole
   ```

   PowerShell

   ```
   Register-IAMRolePolicy `
   -PolicyArn "arn:aws:iam::aws:policy/AmazonSSMManagedEC2InstanceDefaultPolicy" `
   -RoleName "AWSSystemsManagerDefaultEC2InstanceManagementRole"
   ```

4. Öffnen Sie AWS CLI oder Tools für Windows PowerShell und führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   Linux & macOS

   ```
   aws ssm update-service-setting \
   --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
   default-ec2-instance-management-role \
   --setting-value service-role/AWSSystemsManagerDefaultEC2InstanceManagementRole
   ```

   Windows

   ```
   aws ssm update-service-setting ^
   --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
   default-ec2-instance-management-role ^
   --setting-value service-role/AWSSystemsManagerDefaultEC2InstanceManagementRole
   ```

PowerShell

```
Update-SSMServiceSetting `
-SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
default-ec2-instance-management-role" `
-SettingValue "service-role/AWSSystemsManagerDefaultEC2InstanceManagementRole"
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

5. Führen Sie den folgenden Befehl aus, um die aktuellen Diensteinstellungen für die Standard-Hostverwaltungskonfiguration im aktuellen AWS-Konto und anzuzeigen AWS-Region.

Linux & macOS

```
aws ssm get-service-setting \
--setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
default-ec2-instance-management-role
```

Windows

```
aws ssm get-service-setting ^
--setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
default-ec2-instance-management-role
```

PowerShell

```
Get-SSMServiceSetting `
-SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
default-ec2-instance-management-role"
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "ServiceSetting": {
        "SettingId": "/ssm/managed-instance/default-ec2-instance-management-role",
        "SettingValue": "service-role/
AWSSystemsManagerDefaultEC2InstanceManagementRole",
        "LastModifiedDate": "2022-11-28T08:21:03.576000-08:00",
        "LastModifiedUser": "System",
```

```
        "ARN": "arn:aws:ssm:us-east-2:-123456789012:servicesetting/ssm/managed-
instance/default-ec2-instance-management-role",
        "Status": "Custom"
    }
}
```

Die Umgebung der Standardkonfiguration für die Host-Verwaltung deaktivieren

Sie können die Standard-Host-Management-Konfiguration von der Fleet Manager Konsole aus oder mithilfe von AWS Command Line Interface oder deaktivieren AWS Tools for Windows PowerShell.

Sie müssen die Einstellung für die Standard-Host-Management-Konfiguration nacheinander in jeder Region deaktivieren, in der Ihre EC2 Amazon-Instances nicht mehr mit dieser Konfiguration verwaltet werden sollen. Wenn Sie sie in einer Region deaktivieren, wird sie nicht in allen Regionen deaktiviert.

Wenn Sie die Standard-Host-Management-Konfiguration deaktivieren und Ihren EC2 Amazon-Instances kein Instance-Profil angehängt haben, das den Zugriff auf Systems Manager ermöglicht, werden sie nicht mehr von Systems Manager verwaltet.

So deaktivieren Sie die Standardkonfiguration für die Host-Verwaltung (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie im Kontoverwaltung, Standardkonfiguration für die Host-Verwaltung.
4. Deaktivieren Sie Einstellung der Standardkonfiguration für die Host-Verwaltung aktivieren.
5. Wählen Sie Konfigurieren, um die Standardkonfiguration für die Host-Verwaltung zu deaktivieren.

So deaktivieren Sie die Standardkonfiguration für die Host-Verwaltung (Befehlszeile)

- Öffnen Sie AWS CLI oder Tools für Windows PowerShell und führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

  Linux & macOS

  ```
  aws ssm reset-service-setting \
  --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
  default-ec2-instance-management-role
  ```

Windows

```
aws ssm reset-service-setting ^
--setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
default-ec2-instance-management-role
```

PowerShell

```
Reset-SSMServiceSetting `
-SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/managed-instance/
default-ec2-instance-management-role"
```

Beispiele für Richtlinien mit den geringsten Berechtigungen für die Standardkonfiguration für die Host-Verwaltung

Die folgenden Beispielrichtlinien zeigen, wie Sie verhindern können, dass Mitglieder Ihrer Organisation Änderungen an der Standardkonfiguration für die Host-Verwaltung in Ihrem Konto vornehmen.

Richtlinie zur Dienststeuerung für AWS Organizations

Die folgende Richtlinie zeigt, wie Sie verhindern können, dass Mitglieder, die keine Administratorrechte haben, Ihre AWS Organizations Einstellung für die Standard-Host-Management-Konfiguration aktualisieren. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Effect": "Deny",
            "Action": [
                "ssm:UpdateServiceSetting",
                "ssm:ResetServiceSetting"
            ],
            "Resource": "arn:aws:ssm:*:*:servicesetting/ssm/managed-instance/
default-ec2-instance-management-role",
            "Condition": {
```

```
                "StringNotEqualsIgnoreCase": {
                    "aws:PrincipalTag/job-function": [
                        "administrator"
                    ]
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/service-role/
AWSSystemsManagerDefaultEC2InstanceManagementRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ssm.amazonaws.com"
                },
                "StringNotEqualsIgnoreCase": {
                    "aws:PrincipalTag/job-function": [
                        "administrator"
                    ]
                }
            }
        },
        {
            "Effect": "Deny",
            "Resource": "arn:aws:iam::*:role/service-role/
AWSSystemsManagerDefaultEC2InstanceManagementRole",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:DeleteRole"
            ],
            "Condition": {
                "StringNotEqualsIgnoreCase": {
                    "aws:PrincipalTag/job-function": [
                        "administrator"
                    ]
                }
            }
        }
    ]
}
```

Richtlinie für IAM-Prinzipale

Die folgende Richtlinie zeigt, wie Sie verhindern können, dass IAM-Gruppen, -Rollen oder Benutzer in AWS Organizations Ihrem Unternehmen Ihre Einstellung für die Standard-Host-Management-Konfiguration aktualisieren. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ssm:UpdateServiceSetting",
                "ssm:ResetServiceSetting"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:servicesetting/ssm/
managed-instance/default-ec2-instance-management-role"
        },
        {
            "Effect": "Deny",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:DeleteRole",
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/service-role/
AWSSystemsManagerDefaultEC2InstanceManagementRole"
        }
    ]
}
```

Eine Verbindung zu einer von Windows Server verwalteten Instance mit Remote Desktop herstellen

Sie könnenFleet Manager, ein Tool in, verwenden AWS Systems Manager, um mithilfe von (RDP) eine Verbindung zu Ihren Windows Server Amazon Elastic Compute Cloud Remote Desktop Protocol (Amazon EC2) -Instances herzustellen. Fleet Manager Remote Desktop, das von Amazon DCV

unterstützt wird, bietet Ihnen eine sichere Verbindung zu Ihren Windows Server-Instances direkt von der Systems-Manager-Konsole aus. Sie können bis zu vier gleichzeitige Verbindungen in einem einzigen Browserfenster haben.

Die Fleet Manager Remote Desktop API heißt AWS Systems Manager GUI Connect. Informationen zur Verwendung der GUI Connect API in Systems Manager finden Sie in der API-Referenz für AWS Systems Manager GUI Connect.

Sie können Remote Desktop nur mit Instances verwenden, auf denen Windows Server 2012 RTM oder höher ausgeführt wird. Remote Desktop unterstützt nur englischsprachige Eingaben.

Fleet Manager Remote Desktop ist ein Service, der nur für Konsolen verfügbar ist und keine Befehlszeilenverbindungen zu Ihren verwalteten Instances unterstützt. Um über eine Shell eine Verbindung zu einer von Windows Server verwalteten Instance herzustellen, können Sie Session Manager verwenden, ein weiteres Tool in AWS Systems Manager. Weitere Informationen finden Sie unter AWS Systems Manager Session Manager.

> **Note**
>
> Die Dauer einer RDP-Verbindung wird nicht durch die Dauer Ihrer AWS Identity and Access Management (IAM)-Anmeldeinformationen bestimmt. Die Verbindung bleibt bestehen, bis die maximale Verbindungsdauer oder das Leerlaufzeitlimit erreicht ist, je nachdem, was zuerst eintritt. Weitere Informationen finden Sie unter Dauer und Gleichzeitigkeit der Remoteverbindung.

Informationen zur Konfiguration von AWS Identity and Access Management (IAM) - Berechtigungen, damit Ihre Instances mit Systems Manager interagieren können, finden Sie unter Instanzberechtigungen für Systems Manager konfigurieren.

Themen

- Einrichten Ihrer Umgebung
- Konfiguration von IAM-Berechtigungen für Remote Desktop
- Authentifizierung von Remote-Desktop-Verbindungen
- Dauer und Gleichzeitigkeit der Remoteverbindung
- Verarbeitung von AWS IAM Identity Center -Attributen in Systems Manager GUI Connect
- Verbindung zu einem verwalteten Knoten über Remote Desktop

- Anzeigen von Informationen über aktuelle und abgeschlossene Verbindungen

Einrichten Ihrer Umgebung

Vergewissern Sie sich vor der Verwendung von Remote Desktop, dass Ihre Umgebung die folgenden Anforderungen erfüllt:

- Konfiguration von verwalteten Knoten

  Stellen Sie sicher, dass Ihre EC2 Amazon-Instances in Systems Manager als verwaltete Knoten konfiguriert sind.

- SSM Agent-Mindestversion

  Stellen Sie sicher, dass auf den Knoten SSM Agent-Version 3.0.222.0 oder höher ausgeführt wird. Hinweise dazu, wie Sie überprüfen können, welche Agentenversion auf einem Knoten läuft, finden Sie unter Überprüfen der SSM Agent-Versionsnummer. Informationen über das Installieren oder Aktualisieren von SSM Agent finden Sie unter Arbeiten mit SSM Agent.

- Konfiguration des RDP-Ports

  Um Remote-Verbindungen zu akzeptieren, muss der Remote Desktop Services-Service auf Ihren Windows Server-Knoten den Standard-RDP-Port 3389 verwenden. Dies ist die Standardkonfiguration von on Amazon Machine Images (AMIs), bereitgestellt von AWS. Sie müssen nicht explizit irgendwelche eingehenden Ports öffnen, um Remote Desktop zu verwenden.

- PSReadLine-Modulversion für Tastaturfunktionen

  Um sicherzustellen, dass Ihre Tastatur in PowerShell ordnungsgemäß funktioniert, stellen Sie sicher, dass auf den Knoten, auf denen Windows Server-2022 läuft, die PSReadLine-Modulversion 2.2.2 oder höher installiert ist. Wenn sie eine ältere Version verwenden, können Sie die erforderliche Version mit den folgenden Befehlen installieren.

  ```
  Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force
  ```

  Führen Sie nach der Installation des NuGet Paketanbieters den folgenden Befehl aus.

  ```
  Install-Module `
   -Name PSReadLine `
   -Repository PSGallery `
   -MinimumVersion 2.2.2 -Force
  ```

- Session-Manager-Konfiguration

  Bevor Sie Remote Desktop verwenden können, müssen Sie die Voraussetzungen für die Einrichtung von Session Manager erfüllen. Wenn Sie über Remote Desktop eine Verbindung zu einer Instanz herstellen, AWS-Region werden alle für Sie AWS-Konto definierten Sitzungseinstellungen angewendet. Weitere Informationen finden Sie unter Einrichten von Session Manager.

  > **Note**
  >
  > Wenn Sie die Aktivitäten von Session Manager mit Amazon Simple Storage Service (Amazon S3) protokollieren, dann erzeugen Ihre Remotedesktop-Verbindungen den folgenden Fehler in `bucket_name/Port/stderr`. Dieser Fehler ist ein erwartetes Verhalten und kann ignoriert werden.
  >
  > ```
  > Setting up data channel with id SESSION_ID failed: failed to create websocket
  >  for datachannel with error: CreateDataChannel failed with no output or
  >  error: createDataChannel request failed: unexpected response from the service
  >  <BadRequest>
  > <ClientErrorMessage>Session is already terminated</ClientErrorMessage>
  > </BadRequest>
  > ```

Konfiguration von IAM-Berechtigungen für Remote Desktop

Zusätzlich zu den erforderlichen IAM-Berechtigungen für Systems Manager und Session Manager, muss der Benutzer oder die Rolle, die Sie verwenden, über Berechtigungen zum Initiieren von Verbindungen verfügen.

Berechtigungen für das Initiieren von Verbindungen

Um RDP-Verbindungen zu EC2 Instanzen in der Konsole herzustellen, sind die folgenden Berechtigungen erforderlich:

- `ssm-guiconnect:CancelConnection`
- `ssm-guiconnect:GetConnection`
- `ssm-guiconnect:StartConnection`

Berechtigungen zum Auflisten von Verbindungen

Um Verbindungslisten in der Konsole anzuzeigen, ist die folgende Berechtigung erforderlich:

`ssm-guiconnect:ListConnections`

Im Folgenden finden Sie Beispiele für IAM-Richtlinien, die Sie einem Benutzer oder einer Rolle zuordnen können, um verschiedene Arten der Interaktion mit Remote Desktop zu erlauben. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen.

Standardrichtlinie für die Verbindung zu Instanzen EC2

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "EC2",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:GetPasswordData"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SSM",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeInstanceProperties",
                "ssm:GetCommandInvocation",
                "ssm:GetInventorySchema"
            ],
            "Resource": "*"
        },
        {
            "Sid": "TerminateSession",
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
```

```
                    "ssm:resourceTag/aws:ssmmessages:session-id": [
                        "${aws:userid}"
                    ]
                }
            }
        },
        {
            "Sid": "SSMStartSession",
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:*:111122223333:instance/*",
                "arn:aws:ssm:*:111122223333:managed-instance/*",
                "arn:aws:ssm:*::document/AWS-StartPortForwardingSession"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": "ssm-guiconnect.amazonaws.com"
                }
            }
        },
        {
            "Sid": "GuiConnect",
            "Effect": "Allow",
            "Action": [
                "ssm-guiconnect:CancelConnection",
                "ssm-guiconnect:GetConnection",
                "ssm-guiconnect:StartConnection",
                "ssm-guiconnect:ListConnections"
            ],
            "Resource": "*"
        }
    ]
}
```

Richtlinie für die Verbindung zu EC2 Instanzen mit bestimmten Tags

> **Note**
>
> In der folgenden IAM-Richtlinie benötigt der `SSMStartSession`-Abschnitt einen Amazon-Ressourcennamen (ARN) für die Aktion `ssm:StartSession`. Wie gezeigt, benötigt der von Ihnen angegebene ARN keine AWS-Konto ID. Wenn Sie eine Konto-ID angeben, gibt Fleet Manager `AccessDeniedException` zurückgegeben.
> Der `AccessTaggedInstances` Abschnitt, der sich weiter unten in der Beispielrichtlinie befindet, erfordert ebenfalls ARNs für`ssm:StartSession`. Für diese ARNs geben Sie an AWS-Konto IDs.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "EC2",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:GetPasswordData"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SSM",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeInstanceProperties",
                "ssm:GetCommandInvocation",
                "ssm:GetInventorySchema"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SSMStartSession",
            "Effect": "Allow",
            "Action": [
```

```
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ssm:*::document/AWS-StartPortForwardingSession"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": "ssm-guiconnect.amazonaws.com"
                }
            }
        },
        {
            "Sid": "AccessTaggedInstances",
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:*:111122223333:instance/*",
                "arn:aws:ssm:*:111122223333:managed-instance/*"
            ],
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/tag key": [
                        "tag value"
                    ]
                }
            }
        },
        {
            "Sid": "GuiConnect",
            "Effect": "Allow",
            "Action": [
                "ssm-guiconnect:CancelConnection",
                "ssm-guiconnect:GetConnection",
                "ssm-guiconnect:StartConnection",
                "ssm-guiconnect:ListConnections"
            ],
            "Resource": "*"
        }
    ]
}
```

Richtlinie für AWS IAM Identity Center Benutzer, um sich mit EC2 Instanzen zu verbinden

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "SSO",
            "Effect": "Allow",
            "Action": [
                "sso:ListDirectoryAssociations*",
                "identitystore:DescribeUser"
            ],
            "Resource": "*"
        },
        {
            "Sid": "EC2",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:GetPasswordData"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SSM",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeInstanceProperties",
                "ssm:GetCommandInvocation",
                "ssm:GetInventorySchema"
            ],
            "Resource": "*"
        },
        {
            "Sid": "TerminateSession",
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession"
            ],
            "Resource": "*",
            "Condition": {
```

```
                "StringLike": {
                    "ssm:resourceTag/aws:ssmmessages:session-id": [
                        "${aws:userName}"
                    ]
                }
            }
        },
        {
            "Sid": "SSMStartSession",
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ssm:*:*:managed-instance/*",
                "arn:aws:ssm:*:*:document/AWS-StartPortForwardingSession"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": "ssm-guiconnect.amazonaws.com"
                }
            }
        },
        {
            "Sid": "SSMSendCommand",
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ssm:*:*:managed-instance/*",
                "arn:aws:ssm:*:*:document/AWSSSO-CreateSSOUser"
            ]
        },
        {
            "Sid": "GuiConnect",
            "Effect": "Allow",
            "Action": [
                "ssm-guiconnect:CancelConnection",
                "ssm-guiconnect:GetConnection",
                "ssm-guiconnect:StartConnection",
                "ssm-guiconnect:ListConnections"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

Authentifizierung von Remote-Desktop-Verbindungen

Wenn Sie eine Remoteverbindung herstellen, können Sie sich mit Windows Anmeldeinformationen oder dem EC2 Amazon-Schlüsselpaar (`.pem`Datei) authentifizieren, das der Instance zugeordnet ist. Informationen zur Verwendung von Schlüsselpaaren finden Sie unter EC2 Amazon-Schlüsselpaare und Windows -Instances im EC2 Amazon-Benutzerhandbuch.

Wenn Sie für die AWS-Managementkonsole Nutzung authentifiziert sind AWS IAM Identity Center, können Sie alternativ eine Verbindung zu Ihren Instances herstellen, ohne zusätzliche Anmeldeinformationen angeben zu müssen. Ein Beispiel für eine Richtlinie, die die Authentifizierung von Fernverbindungen mit IAM Identity Center erlaubt, finden Sie unter Konfiguration von IAM-Berechtigungen für Remote Desktop.

Bevor Sie beginnen

Beachten Sie die folgenden Bedingungen für die Verwendung der IAM Identity Center-Authentifizierung, bevor Sie eine Verbindung über Remote Desktop herstellen.

- Remote Desktop unterstützt die IAM Identity Center-Authentifizierung für Knoten in derselben AWS-Region , in der Sie IAM Identity Center aktiviert haben.
- Remote Desktop unterstützt IAM Identity Center-Benutzernamen mit bis zu 16 Zeichen.
- Remote Desktop unterstützt IAM Identity Center-Benutzernamen, die aus alphanumerischen Zeichen und den folgenden Sonderzeichen bestehen: `. - _`

> **Important**
>
> Für IAM-Identity-Center-Benutzernamen, die die folgenden Zeichen enthalten, können keine Verbindungen hergestellt werden: `+ = ,`
> IAM Identity Center unterstützt diese Zeichen in Benutzernamen, Fleet Manager-RDP-Verbindungen jedoch nicht.
> Wenn darüber hinaus ein IAM-Identity-Center-Benutzername ein oder mehrere `@`-Symbole enthält, ignoriert Fleet Manager das erste `@`-Symbol und alle darauf folgenden Zeichen, unabhängig davon, ob das `@` den Domainteil einer E-Mail-Adresse einleitet oder nicht.

Für den IAM Identity Center-Benutzernamen `diego_ramirez@example.com` wird der `@example.com`-Teil beispielsweise ignoriert und der Benutzername für Fleet Manager wird `diego_ramirez`. Für `diego_r@mirez@example.com` ignoriert Fleet Manager `@mirez@example.com` und der Benutzername für Fleet Manager wird `diego_r`.

- Wenn eine Verbindung mit IAM Identity Center authentifiziert wird, erstellt Remote Desktop einen lokalen Windows-Benutzer in der Gruppe Lokale Administratoren der Instance. Dieser Benutzer bleibt bestehen, nachdem die Remoteverbindung beendet wurde.
- Remote Desktop erlaubt keine IAM Identity Center-Authentifizierung für Knoten, die Microsoft Active Directory-Domain-Controller sind.
- Obwohl Remote Desktop die Verwendung der IAM Identity Center-Authentifizierung für Knoten, die einer Active Directory-Domain angeschlossen sind, ermöglicht, raten wir davon ab, dies zu tun. Diese Authentifizierungsmethode gewährt Benutzern administrative Berechtigungen, die restriktivere, von der Domain gewährte Berechtigungen außer Kraft setzen können.

Unterstützte Regionen für die IAM Identity Center-Authentifizierung

Remote Desktop-Verbindungen, die die IAM Identity Center-Authentifizierung verwenden, werden in den folgenden AWS-Regionen unterstützt:

- USA Ost (Ohio): (us-east-2)
- USA Ost (Nord-Virginia): (us-east-1)
- USA West (Nordkalifornien) (us-west-1)
- USA West (Oregon): (us-west-2)
- Afrika (Kapstadt) (af-south-1)
- Asien-Pazifik (Hongkong) (ap-east-1)
- Asien-Pazifik (Mumbai): (ap-south-1)
- Asien-Pazifik (Tokyo) (ap-northeast-1)
- Asien-Pazifik (Seoul): (ap-northeast-2)
- Asien-Pazifik (Osaka) (ap-northeast-3)
- Asien-Pazifik (Singapur): (ap-southeast-1)
- Asien-Pazifik (Sydney): (ap-southeast-2)
- Asien-Pazifik (Jakarta) (ap-southeast-3)
- Kanada (Zentral): (ca-central-1)

- Europa (Frankfurt) (eu-central-1)
- Europa (Stockholm) (eu-north-1)
- Europa (Irland) (eu-west-1)
- Europa (London) (eu-west-2)
- Europa (Paris) (eu-west-3)
- Israel (Tel Aviv) (il-central-1)
- Südamerika (São Paulo) (sa-east-1)
- Europa (Mailand) (eu-south-1)
- Naher Osten (Bahrain) (me-south-1)
- AWS GovCloud (US-Ost) (us-gov-east-1)
- AWS GovCloud (US-West) (us-gov-west-1)

Dauer und Gleichzeitigkeit der Remoteverbindung

Die folgenden Bedingungen gelten für aktive Remote-Desktop-Verbindungen:

- Verbindungsdauer

  Standardmäßig wird eine Remote-Desktop-Verbindung nach 60 Minuten getrennt. Um zu verhindern, dass eine Verbindung getrennt wird, können Sie die Option Sitzung erneuern wählen, bevor sie getrennt wird, um den Timer für die Verbindungsdauer zurückzusetzen.
- Verbindungstimeout

  Eine Remote-Desktop-Verbindung wird getrennt, nachdem sie länger als 10 Minuten inaktiv war.
- Verbindungspersistenz

  Nachdem Sie mit Remote Desktop eine Verbindung zu Windows Server hergestellt haben, bleibt die Verbindung bestehen, bis die maximale Verbindungsdauer (60 Minuten) oder das Leerlaufzeitlimit (10 Minuten) erreicht ist. Die Verbindungsdauer wird nicht durch die Dauer Ihrer AWS Identity and Access Management (IAM-) Anmeldeinformationen bestimmt. Die Verbindung bleibt auch nach Ablauf der IAM-Anmeldeinformationen bestehen, bis die maximale Verbindungsdauer erreicht wird. Wenn Sie Remote Desktop verwenden, sollten Sie Ihre Verbindung nach Ablauf Ihrer IAM-Anmeldeinformationen beenden, indem Sie die Browserseite verlassen.
- Gleichzeitige Verbindungen

Standardmäßig können Sie für dasselbe und maximal 5 aktive Remotedesktopverbindungen gleichzeitig AWS-Konto haben. AWS-Region Informationen zum Beantragen einer Erhöhung der Service Quotas um bis zu 50 gleichzeitige Verbindungen finden Sie unter Eine Erhöhung der Kontingente beantragen im Benutzerhandbuch für Servicekontingenten.

> **Note**
>
> Die Standardlizenz für Windows Server ermöglicht zwei gleichzeitige RDP-Verbindungen. Um mehr Verbindungen zu unterstützen, müssen Sie zusätzliche Clientzugriffslizenzen (CALs) von Microsoft oder Microsoft Remote Desktop Services-Lizenzen von erwerben AWS. Weitere Informationen zu zusätzlichen Lizenzen finden Sie in den folgenden Themen:
>
> - Clientzugriffslizenzen und Verwaltungslizenzen auf der Microsoft-Website
> - Verwenden benutzerbasierter License-Manager-Abonnements für unterstützte Softwareprodukte im Benutzerhandbuch für License Manager

Verarbeitung von AWS IAM Identity Center -Attributen in Systems Manager GUI Connect

Systems Manager GUI Connect ist die API, die Fleet Manager Verbindungen zu EC2 Instanzen mithilfe von RDP unterstützt. Die folgenden IAM-Identity-Center-Benutzerdaten werden nach dem Trennen einer Verbindung beibehalten:

- `username`

Systems Manager GUI Connect verschlüsselt dieses Identitätsattribut im Ruhezustand Von AWS verwalteter Schlüssel standardmäßig mit einem. Vom Kunden verwaltete Schlüssel werden für die Verschlüsselung dieses Attributs in Systems Manager GUI Connect nicht unterstützt. Wenn Sie einen Benutzer in Ihrer IAM-Identity-Center-Instance löschen, wird das verknüpfte `username`-Attribut 7 Jahre lang in Systems Manager GUI Connect aufbewahrt und anschließend gelöscht. Diese Daten werden zur Unterstützung von Überwachungsereignissen aufbewahrt, z. B. zur Auflistung des Verbindungsverlaufs von Systems Manager GUI Connect. Die Daten können nicht manuell gelöscht werden.

Verbindung zu einem verwalteten Knoten über Remote Desktop

copy/paste Browser-Unterstützung für Text

Mit den Browsern Google Chrome und Microsoft Edge können Sie Text von einem verwalteten Knoten auf Ihren lokalen Computer und von Ihrem lokalen Computer in einen verwalteten Knoten, mit dem Sie verbunden sind, kopieren und einfügen.

Mit dem Mozilla-Firefox-Browser können Sie Text nur von einem verwalteten Knoten auf Ihren lokalen Computer kopieren und einfügen. Das Kopieren von Ihrem lokalen Computer auf den verwalteten Knoten wird nicht unterstützt.

So stellen Sie über Fleet Manager Remote Desktop eine Verbindung zu einem verwalteten Knoten her

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie den Knoten, zu dem Sie eine Verbindung herstellen möchten. Sie können entweder das Kontrollkästchen oder den Knotennamen auswählen.
4. Wählen Sie im Menü Knotenaktionen die Option Mit Remote Desktop verbinden.
5. Wählen Sie den gewünschten Authentication type (Authentifizierungs-Typ). Wenn Sie Benutzeranmeldeinformationen wählen, geben Sie den Benutzernamen und das Passwort für ein Windows-Benutzerkonto auf dem Knoten ein, zu dem Sie eine Verbindung herstellen möchten. Wenn Sie Schlüsselpaar wählen, können Sie die Authentifizierung mit einer der folgenden Methoden durchführen:

   a. Wählen Sie Lokale Maschine durchsuchen, wenn Sie den mit Ihrer Instance verbundenen PEM-Schlüssel aus Ihrem lokalen Dateisystem auswählen möchten.

      – oder –

   b. Wählen Sie Schlüsselpaarinhalt einfügen, wenn Sie den Inhalt der PEM-Datei kopieren und in das vorgesehene Feld einfügen möchten.
6. Wählen Sie Verbinden aus.
7. Um Ihre bevorzugte Bildschirmauflösung zu wählen, wählen Sie im Menü Aktionen die Option Auflösungen, und wählen Sie dann eine der folgenden Optionen:

   - Automatisch anpassen
   - 1920 x 1080
   - 1 400 x 900
   - 1 366 x 768

- 800 x 600

Die Option Automatisch anpassen legt die Auflösung auf der Grundlage der erkannten Bildschirmgröße fest.

Anzeigen von Informationen über aktuelle und abgeschlossene Verbindungen

Sie können den Fleet Manager-Bereich der Systems-Manager-Konsole verwenden, um Informationen zu RDP-Verbindungen anzuzeigen, die in Ihrem Konto hergestellt wurden. Mithilfe einer Reihe von Filtern können Sie die angezeigte Liste der Verbindungen auf einen Zeitraum, eine bestimmte Instance, den Benutzer, der die Verbindungen hergestellt hat, und Verbindungen mit einem bestimmten Status einschränken. Die Konsole bietet auch Registerkarten, auf denen Informationen zu allen derzeit aktiven Verbindungen und allen vergangenen Verbindungen angezeigt werden.

So zeigen Sie Informationen über aktuelle und abgeschlossene Verbindungen an

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie Kontoverwaltung, Mit Remote Desktop Connect.
4. Wählen Sie eine der folgenden Registerkarten:
   - Aktive Verbindungen
   - Verlauf der Verbindung
5. Um die Liste der angezeigten Verbindungsergebnisse weiter einzuschränken, geben Sie einen oder mehrere Filter in das Suchfeld ( ) ein. Sie können auch einen Freitext-Suchbegriff eingeben.

Verwaltung von Amazon-EBS-Volumes auf verwalteten Instances

Amazon Elastic Block Store (Amazon EBS) bietet Volumes für die Speicherung auf Blockebene, die in Verbindung mit Instances von Amazon Elastic Compute Cloud (EC2) verwendet werden. EBS-Volumes verhalten sich wie unformatierte Blockgeräte. Sie können diese Volumes als Geräte auf Ihren Instances mounten.

Sie können Fleet Manager verwenden, ein Tool in AWS Systems Manager, um Amazon-EBS-Volumes auf Ihren verwalteten Instances zu verwalten. Sie können beispielsweise ein EBS-Volume initialisieren, eine Partition formatieren und das Volume mounten, um es für die Nutzung verfügbar zu machen.

> **Note**
>
> Fleet Manager unterstützt derzeit die Amazon-EBS-Volume-Verwaltung nur für Windows Server-Instances.

Anzeigen von Details zu EBS-Volumes

So zeigen Sie Details für ein EBS-Volume mit Fleet Manager an

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben der verwalteten Instance aus, deren EBS-Volumedetails Sie anzeigen möchten.
4. Wählen Sie die Option Details anzeigen aus.
5. Wählen Sie Tools, EBS-Volumes.
6. Um Details zu einem EBS-Volume anzuzeigen, wählen Sie seine ID in der Spalte Volume-ID.

Initialisieren und Formatieren eines EBS-Volumes

So initialisieren und formatieren Sie ein EBS-Volume mit Fleet Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Schaltfläche neben der verwalteten Instance aus, die Sie initialisieren, formatieren und mounten möchten. Sie können ein EBS-Volume nur initialisieren, wenn sein Datenträger leer ist.
4. Wählen Sie die Option Details anzeigen aus.
5. Wählen Sie im Menü Tools die Option EBS-Volumes.

6. Wählen Sie die Schaltfläche neben dem EBS-Volume, das Sie initialisieren und formatieren möchten.
7. Wählen Sie Initialisieren und formatieren.
8. Wählen Sie unter Partitionsstil den Partitionsstil aus, den Sie für das EBS-Volume verwenden möchten.
9. (Optional) Wählen Sie einen Laufwerksbuchstaben für die Partition.
10. (Optional) Geben Sie einen Partitionsnamen ein, um die Partition zu identifizieren.
11. Wählen Sie das Dateisystem aus, das zum Organisieren der in der Partition gespeicherten Dateien und Daten verwendet werden soll.
12. Wählen Sie Bestätigen, um das EBS-Volume zur Verwendung verfügbar zu machen. Sie können die Partitionskonfiguration von der AWS-Managementkonsole nach der Bestätigung nicht mehr ändern. Sie können sich jedoch mit SSH oder RDP bei der Instance anmelden, um die Partitionskonfiguration zu ändern.

Zugriff auf das Wissensdatenbank-Portal von Red Hat

Sie können Fleet Manager verwenden, ein Tool in AWS Systems Manager, um das Wissensdatenbank-Portal aufzurufen, wenn Sie Red-Hat-Kunde sind. Sie gelten als Red-Hat-Kunde, wenn Sie auf AWS Red Hat Enterprise Linux (RHEL)-Instances ausführen oder RHEL-Services verwenden. Das Wissendatenbank-Portal umfasst Binärdateien sowie Wissensaustausch- und Diskussionsforen für Community-Support, die nur von Red Hat lizenzierten Kunden zur Verfügung stehen.

Zusätzlich zu den erforderlichen AWS Identity and Access Management (IAM)-Berechtigungen für Systems Manager und Fleet Manager, muss der Benutzer oder die Rolle, die Sie für den Zugriff auf die Konsole verwenden, die Aktion `rhelkb:GetRhelURL` erlauben, um auf das Wissensdatenbank-Portal zuzugreifen.

Zugreifen auf das Red-Hat-Knowledgebase-Portal

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die RHEL-Instance, die Sie verwenden möchten, um sich mit dem Knowledgebase-Portal von Red Hat zu verbinden.

4. Wählen Sie Kontomanagement, Auf Red-Hat-Wissensdatenbank zugreifen, um die Red-Hat-Wissensdatenbank zu öffnen.

Bei Verwendung von RHEL auf AWS, um vollständig unterstützte RHEL-Workloads auszuführen, können Sie auch mit Ihren AWS-Anmeldeinformationen über die Website von Red Hat auf das Wissensdatenbankportal von Red Hat zugreifen.

Problembehandlung bei der Verfügbarkeit verwalteter Knoten

Bei verschiedenen AWS Systems Manager Tools wie Run Command DistributorSession Manager, und können Sie die verwalteten Knoten, auf denen Sie einen Vorgang ausführen möchten, manuell auswählen. In solchen Fällen zeigt das System, nachdem Sie angegeben haben, dass Sie Knoten manuell auswählen möchten, eine Liste der verwalteten Knoten an, auf denen Sie die Operation ausführen können.

Dieses Thema liefert Informationen zur Diagnose, warum ein verwalteter Knoten, für den Sie bestätigt haben, dass er ausgeführt wird, nicht in Ihren Listen verwalteter Knoten in Systems Manager aufgeführt wird.

Damit ein Knoten von Systems Manager verwaltet und in Listen verwalteter Knoten verfügbar ist, muss er drei primäre Anforderungen erfüllen:

- SSM Agent muss auf dem Knoten installiert sein und mit einem unterstützten Betriebssystem ausgeführt werden.

  > **Note**
  >
  > Einige AWS managed Amazon Machine Images (AMIs) sind so konfiguriert, dass sie Instances mit SSM Agentvorinstallierter Installation starten. (Sie können auch ein benutzerdefiniertes AMI zur Vorinstallation von SSM Agent konfigurieren). Weitere Informationen finden Sie unter Finden Sie AMIs mit dem vorinstallierten SSM Agent.

- Für Amazon Elastic Compute Cloud (Amazon EC2) -Instances müssen Sie ein AWS Identity and Access Management (IAM-) Instance-Profil an die Instance anhängen. Das Instance-Profil ermöglicht es der Instance, mit dem Systems-Manager-Service zu kommunizieren. Wenn Sie der Instance kein Instance-Profil zuweisen, registrieren Sie sie mit einer Hybrid-Aktivierung, was kein übliches Szenario ist.

- SSM Agent muss in der Lage sein, eine Verbindung zu einem Systems Manager-Endpunkt herzustellen, um sich beim Service zu registrieren. Danach muss der verwaltete Knoten für den Service verfügbar sein, was vom Service bestätigt wird, der alle fünf Minuten ein Signal sendet, um den Zustand der Instance zu überprüfen.
- Nachdem der Status eines verwalteten Knotens mindestens 30 Tage lang `Connection Lost` gewesen ist, wird der Knoten möglicherweise nicht mehr in der Fleet Manager-Konsole aufgeführt. Beheben Sie das Problem, das den Verbindungsverlust verursacht hat, um ihn wieder in die Liste aufzunehmen.

Nachdem Sie überprüft haben, dass ein verwalteter Knoten ausgeführt wird, können Sie den folgenden Befehl verwenden, um zu überprüfen, ob SSM Agent erfolgreich beim Systems-Manager-Service registriert wurde. Dieser Befehl gibt keine Ergebnisse zurück, bis eine erfolgreiche Registrierung stattgefunden hat.

Linux & macOS

```
aws ssm describe-instance-associations-status \
    --instance-id instance-id
```

Windows

```
aws ssm describe-instance-associations-status ^
    --instance-id instance-id
```

PowerShell

```
Get-SSMInstanceAssociationsStatus `
    -InstanceId instance-id
```

Wenn die Registrierung erfolgreich war und der verwaltete Knoten jetzt für Systems-Manager-Operationen verfügbar ist, gibt der Befehl ähnliche Ergebnisse wie die folgenden zurück.

```
{
    "InstanceAssociationStatusInfos": [
        {
            "AssociationId": "fa262de1-6150-4a90-8f53-d7eb5EXAMPLE",
            "Name": "AWS-GatherSoftwareInventory",
            "DocumentVersion": "1",
```

```
            "AssociationVersion": "1",
            "InstanceId": "i-02573cafcfEXAMPLE",
            "Status": "Pending",
            "DetailedStatus": "Associated"
        },
        {
            "AssociationId": "f9ec7a0f-6104-4273-8975-82e34EXAMPLE",
            "Name": "AWS-RunPatchBaseline",
            "DocumentVersion": "1",
            "AssociationVersion": "1",
            "InstanceId": "i-02573cafcfEXAMPLE",
            "Status": "Queued",
            "AssociationName": "SystemAssociationForScanningPatches"
        }
    ]
}
```

Wenn die Registrierung noch nicht abgeschlossen wurde oder nicht erfolgreich war, gibt der Befehl ähnliche Ergebnisse wie die folgenden zurück:

```
{
    "InstanceAssociationStatusInfos": []
}
```

Wenn der Befehl nach etwa 5 Minuten keine Ergebnisse zurückgibt, verwenden Sie die folgenden Informationen, um Probleme mit Ihren verwalteten Knoten zu beheben.

Themen

- Lösung 1: Überprüfen Sie, ob SSM Agent installiert ist und auf dem verwalteten Knoten ausgeführt wird
- Lösung 2: Stellen Sie sicher, dass ein IAM-Instance-Profil für die Instance angegeben wurde (nur EC2 Instances)
- Lösung 3: Überprüfen der Konnektivität des Service-Endpunkts
- Lösung 4: Überprüfen der Unterstützung des Zielbetriebssystems
- Lösung 5: Stellen Sie sicher, dass Sie in derselben Instanz arbeiten AWS-Region wie die EC2 Amazon-Instance
- Lösung 6: Überprüfen Sie die Proxy-Konfiguration, die Sie auf SSM Agent in Ihrem verwalteten Knoten angewendet haben

- Lösung 7: Installieren eines TLS-Zertifikats auf verwalteten Instances
- Problembehandlung bei der Verfügbarkeit von verwalteten Knoten mit ssm-cli

Lösung 1: Überprüfen Sie, ob SSM Agent installiert ist und auf dem verwalteten Knoten ausgeführt wird

Stellen Sie sicher, dass die neueste Version von SSM Agent auf dem verwalteten Knoten installiert ist und ausgeführt wird.

Informationen zum Feststellen, ob SSM Agent installiert ist und auf einem verwalteten Knoten ausgeführt wird, finden Sie unter Prüfen des SSM Agent-Status und Starten des Agenten.

Um SSM Agent auf einem verwalteten Knoten zu installieren bzw. deinstallieren, siehe folgende Themen:

- Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux
- So installieren Sie den SSM Agent in Hybrid-Linux-Knoten
- Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Windows Server
- So installieren Sie den SSM Agent in Hybrid-Windows-Knoten

Lösung 2: Stellen Sie sicher, dass ein IAM-Instance-Profil für die Instance angegeben wurde (nur EC2 Instances)

Stellen Sie bei Amazon Elastic Compute Cloud (Amazon EC2) -Instances sicher, dass die Instance mit einem AWS Identity and Access Management (IAM-) Instance-Profil konfiguriert ist, das es der Instance ermöglicht, mit der Systems Manager Manager-API zu kommunizieren. Stellen Sie außerdem sicher, dass Ihr Benutzer über eine IAM-Vertrauensrichtlinie verfügt, die es Ihrem Benutzer ermöglicht, mit der Systems-Manager-API zu kommunizieren.

> **Note**
>
> Lokale Server, Edge-Geräte und virtuelle Maschinen (VMs) verwenden eine IAM-Servicerolle anstelle eines Instanzprofils. Weitere Informationen finden Sie unter Erstellen der für Systems Manager in Hybrid- und Multi-Cloud-Umgebungen erforderlichen IAM-Servicerolle.

Um festzustellen, ob ein Instanzprofil mit den erforderlichen Berechtigungen an eine Instanz angehängt ist EC2

1. Öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie die Instance, die nach einem Instance-Profil überprüft werden soll.
4. Suchen Sie auf der Registerkarte Description (Beschreibung) im unteren Bereich die IAM-Rolle und wählen Sie den Namen der Rolle.
5. Vergewissern Sie sich auf der Seite Summary des Instance-Profils auf der Registerkarte Permissions (Berechtigungen), dass unter den Berechtigungsrichtlinien `AmazonSSMManagedInstanceCore` aufgeführt ist.

   Wenn stattdessen eine benutzerdefinierte Richtlinie verwendet wird, stellen Sie sicher, dass sie dieselben Berechtigungen wie `AmazonSSMManagedInstanceCore` bereitstellt.

   Öffnen Sie `AmazonSSMManagedInstanceCore` in der Konsole

   Informationen über andere Richtlinien, die an ein Instance-Profil für Systems Manager angefügt werden können, finden Sie unter Konfigurieren von erforderlichen Instances-Berechtigungen für Systems Manager.

Lösung 3: Überprüfen der Konnektivität des Service-Endpunkts

Stellen Sie sicher, dass die Instance eine Verbindung zu den Systems Manager Service-Endpunkten hat. Diese Konnektivität wird entweder durch das Erstellen und Konfigurieren von VPC-Endpunkten für Systems Manager oder durch die Genehmigung von ausgehenden HTTPS-Datenverkehr (Port 443) zu den Service-Endpunkten bereitgestellt.

Bei EC2 Amazon-Instances AWS-Region wird der Systems Manager Manager-Serviceendpunkt für die zur Registrierung der Instance verwendet, wenn Ihre Virtual Private Cloud (VPC) -Konfiguration ausgehenden Datenverkehr zulässt. Wenn die VPC-Konfiguration, in der die Instance gestartet wurde, jedoch keinen ausgehenden Datenverkehr zulässt und Sie diese Konfiguration nicht ändern können, um Konnektivität zu den öffentlichen Service-Endpunkten zu erlauben, müssen Sie stattdessen Schnittstellenendpunkte für Ihre VPC konfigurieren.

Weitere Informationen finden Sie unter Verbessern der Sicherheit von EC2 Instances mithilfe von VPC-Endpunkten für Systems Manager.

Lösung 4: Überprüfen der Unterstützung des Zielbetriebssystems

Stellen Sie sicher, dass die ausgewählte Operation für den Typ von verwalteten Knoten ausgeführt werden kann, den Sie in der Liste erwarten. Einige Systems Manager-Vorgänge können nur auf Windows-Instanceen oder nur auf Linux-Instances abzielen. Zum Beispiel können die Systems Manager (SSM) Dokumente `AWS-InstallPowerShellModule` und `AWS-ConfigureCloudWatch` nur auf Windows-Instances ausgeführt werden. Wenn Sie auf der Seite Run a command (Ausführen eines Befehls) eines dieser Dokumente auswählen und die Option Choose instances manually (Instancen manuell auswählen) wählen, werden nur Ihre Windows-Instances aufgelistet und stehen zur Auswahl.

Lösung 5: Stellen Sie sicher, dass Sie in derselben Instanz arbeiten AWS-Region wie die EC2 Amazon-Instance

EC2 Amazon-Instances werden in bestimmten Regionen erstellt und sind verfügbar AWS-Regionen, z. B. in der Region USA Ost (Ohio) (us-east-2) oder Europa (Irland) (eu-west-1). Stellen Sie sicher, dass Sie in derselben AWS-Region EC2 Amazon-Instance arbeiten, mit der Sie arbeiten möchten. Weitere Informationen dazu erhalten Sie unter Choosing a Region (Region wählen) in Getting Started with the AWS-Managementkonsole.

Lösung 6: Überprüfen Sie die Proxy-Konfiguration, die Sie auf SSM Agent in Ihrem verwalteten Knoten angewendet haben

Überprüfen Sie, ob die Proxy-Konfiguration, die Sie auf SSM Agent in Ihrem verwalteten Knoten angewendet haben, korrekt ist. Wenn die Proxy-Konfiguration falsch ist, kann der Knoten keine Verbindung zu den erforderlichen Service-Endpunkten herstellen, oder Systems Manager identifiziert möglicherweise das Betriebssystem des verwalteten Knotens falsch. Weitere Informationen erhalten Sie unter Konfigurieren Sie SSM Agent, um einen Proxy in Linux-Knoten zu verwenden und Konfigurieren des SSM Agent zur Nutzung eines Proxys für Windows Server-Instances.

Lösung 7: Installieren eines TLS-Zertifikats auf verwalteten Instances

Auf jeder verwalteten Instance, die Sie verwenden, muss ein Transport Layer Security (TLS) - Zertifikat installiert sein AWS Systems Manager. AWS-Services Verwenden Sie diese Zertifikate, um Anrufe an andere AWS-Services zu verschlüsseln.

Ein TLS-Zertifikat ist bereits standardmäßig auf jeder EC2 Amazon-Instance installiert, die aus any Amazon Machine Image (AMI) erstellt wurde. Die meisten modernen Betriebssysteme enthalten das erforderliche TLS-Zertifikat von Amazon Trust Services CAs in ihrem Trust Store.

Um zu überprüfen, ob das erforderliche Zertifikat auf Ihrer Instance installiert ist, führen Sie den folgenden Befehl basierend auf dem Betriebssystem Ihrer Instance aus. Achten Sie darauf, den *region* Teil der URL durch den Teil zu ersetzen, AWS-Region in dem sich Ihre verwaltete Instance befindet.

Linux & macOS

```
curl -L https://ssm.region.amazonaws.com
```

Windows

```
Invoke-WebRequest -Uri https://ssm.region.amazonaws.com
```

Der Befehl sollte einen `UnknownOperationException`-Fehler zurückgeben. Wenn Sie stattdessen eine SSL/TLS Fehlermeldung erhalten, ist das erforderliche Zertifikat möglicherweise nicht installiert.

Wenn Sie feststellen, AMIs dass die erforderlichen CA-Zertifikate von Amazon Trust Services nicht auf Ihren Basisbetriebssystemen, auf Instances installiert sind, die nicht von Amazon bereitgestellt wurden, oder auf Ihren eigenen lokalen Servern VMs, müssen Sie ein Zertifikat von Amazon Trust Services installieren und zulassen oder AWS Certificate Manager (ACM) verwenden, um Zertifikate für einen unterstützten integrierten Service zu erstellen und zu verwalten.

Auf jeder Ihrer verwalteten Instances muss eines der folgenden Transport Layer Security (TLS)-Zertifikate installiert sein.

- Amazon Root CA 1
- Starfield Services Root Certificate Authority – G2
- Starfield Class 2 Certificate Authority

Informationen zur Verwendung von ACM finden Sie im AWS Certificate Manager -Benutzerhandbuch.

Wenn Zertifikate in Ihrer Datenverarbeitungsumgebung von einem Gruppenrichtlinienobjekt (GPO) verwaltet werden, dann müssen Sie möglicherweise die Gruppenrichtlinie so konfigurieren, dass eines dieser Zertifikate enthalten ist.

Weitere Informationen zu den Amazon Root- und Starfield-Zertifikaten finden Sie im Blogbeitrag How to Prepare for AWS's Move to Its Own Certificate Authority.

Problembehandlung bei der Verfügbarkeit von verwalteten Knoten mit **ssm-cli**

Die `ssm-cli` ist ein eigenständiges Befehlszeilentool, das in der SSM Agent-Installation enthalten ist. Wenn Sie SSM Agent 3.1.501.0 oder höher auf einem Computer installieren, können Sie `ssm-cli`-Befehle auf diesem Computer ausführen. Mithilfe der Ausgabe dieser Befehle können Sie feststellen, ob die Maschine die Mindestanforderungen für eine Amazon-EC2-Instance oder eine Nicht-EC2-Maschine erfüllt, die von AWS Systems Manager verwaltet und daher zu Listen verwalteter Knoten in Systems Manager hinzugefügt werden soll. (SSM Agent-Version 3.1.501.0 wurde im November 2021 veröffentlicht.)

Mindestanforderungen

Damit eine Amazon-EC2-Instance oder Nicht-EC2-Maschine von AWS Systems Manager verwaltet werden kann und in den Listen der verwalteten Knoten verfügbar ist, muss sie drei Hauptanforderungen erfüllen:

- SSM Agent muss auf einer Maschine mit einem unterstützten Betriebssystem installiert sein und laufen.

  Einige von AWS verwaltete Amazon Machine Images (AMIs) für EC2 sind so konfiguriert, dass sie Instances mit vorinstalliertem SSM Agent launchen. (Sie können auch ein benutzerdefiniertes AMI zur Vorinstallation von SSM Agent konfigurieren). Weitere Informationen finden Sie unter Finden Sie AMIs mit dem vorinstallierten SSM Agent.
- Der Maschine muss ein AWS Identity and Access Management (IAM)-Instance-Profil (für EC2-Instances) oder eine IAM-Servicerolle (für Nicht-EC2-Maschinen) zugeordnet werden, die die erforderlichen Berechtigungen für die Kommunikation mit dem Systems-Manager-Service bereitstellt.
- SSM Agent muss in der Lage sein, eine Verbindung zu einem Systems-Manager-Endpunkt herzustellen, um sich beim Service anzumelden. Danach muss der verwaltete Knoten für den Service verfügbar sein, was vom Service bestätigt wird, der alle fünf Minuten ein Signal sendet, um den Zustand des verwalteten Knoten zu überprüfen.

Vorkonfigurierte Befehle in **ssm-cli**

Es sind vorkonfigurierte Befehle enthalten, die die erforderlichen Informationen sammeln, um Ihnen bei der Diagnose zu helfen, warum eine Maschine, von der Sie bestätigt haben, dass sie läuft, nicht in Ihrer Liste der verwalteten Knoten in Systems Manager enthalten ist. Diese Befehle werden ausgeführt, wenn Sie die `get-diagnostics`-Option angeben.

Führen Sie auf der Maschine den folgenden Befehl aus, um `ssm-cli` für die Problembehebung in Bezug auf die Verfügbarkeit der verwalteten Knoten zu verwenden.

Linux & macOS

```
ssm-cli get-diagnostics --output table
```

Windows

In Windows Server-Maschinen müssen Sie zum `C:\Program Files\Amazon\SSM`-Verzeichnis navigieren, bevor Sie den Befehl ausführen.

```
ssm-cli.exe get-diagnostics --output table
```

PowerShell

Auf Windows Server-Maschinen müssen Sie zum `C:\Program Files\Amazon\SSM`-Verzeichnis navigieren, bevor Sie den Befehl ausführen.

```
.\ssm-cli.exe get-diagnostics --output table
```

Der Befehl liefert eine Ausgabe in Form einer Tabelle ähnlich der folgenden.

> **Note**
>
> Konnektivitätsprüfungen an den Endpunkten `ssmmessages`, `s3`, `kms`, `logs` und `monitoring` sind für zusätzliche optionale Features wie Session Manager, die sich bei Amazon Simple Storage Service (Amazon S3) oder Amazon CloudWatch Logs anmelden und AWS Key Management Service (AWS KMS)-Verschlüsselung verwenden.

Linux & macOS

```
[root@instance]# ssm-cli get-diagnostics --output table
##################################################################################################
# Check                                  # Status  # Note
                                        #
##################################################################################################
# EC2 IMDS                               # Success # IMDS is accessible and has
 instance id i-0123456789abcdefa in Region  #
```

```
#                                       #         # us-east-2
                                       #
#############################################################################################
# Hybrid instance registration          # Skipped # Instance does not have hybrid
 registration                           #
#############################################################################################
# Connectivity to ssm endpoint          # Success # ssm.us-east-2.amazonaws.com is
 reachable                              #
#############################################################################################
# Connectivity to ec2messages endpoint  # Success # ec2messages.us-
east-2.amazonaws.com is reachable                  #
#############################################################################################
# Connectivity to ssmmessages endpoint  # Success # ssmmessages.us-
east-2.amazonaws.com is reachable                  #
#############################################################################################
# Connectivity to s3 endpoint           # Success # s3.us-east-2.amazonaws.com is
 reachable                              #
#############################################################################################
# Connectivity to kms endpoint          # Success # kms.us-east-2.amazonaws.com is
 reachable                              #
#############################################################################################
# Connectivity to logs endpoint         # Success # logs.us-east-2.amazonaws.com is
 reachable                             #
#############################################################################################
# Connectivity to monitoring endpoint   # Success # monitoring.us-
east-2.amazonaws.com is reachable                   #
#############################################################################################
# AWS Credentials                       # Success # Credentials are for
                                       #
#                                       #         #
 arn:aws:sts::123456789012:assumed-role/Fullaccess/i-0123456789abcdefa #
#                                       #         # and will expire at 2021-08-17
 18:47:49 +0000 UTC                     #
#############################################################################################
# Agent service                         # Success # Agent service is running and is
 running as expected user              #
#############################################################################################
# Proxy configuration                   # Skipped # No proxy configuration detected
                                       #
#############################################################################################
# SSM Agent version                     # Success # SSM Agent version is 3.0.1209.0,
 latest available agent version is     #
#                                       #         # 3.1.192.0
                                       #
```

```
################################################################################################
```

Windows Server and PowerShell

```
PS C:\Program Files\Amazon\SSM> .\ssm-cli.exe get-diagnostics --output table
################################################################################################
# Check                                 # Status  # Note
                                      #
################################################################################################
# EC2 IMDS                              # Success # IMDS is accessible and has
 instance id i-0123456789EXAMPLE in       #
#                                       #         # Region us-east-2
                                      #
################################################################################################
# Hybrid instance registration          # Skipped # Instance does not have hybrid
 registration                          #
################################################################################################
# Connectivity to ssm endpoint          # Success # ssm.us-east-2.amazonaws.com is
 reachable                            #
################################################################################################
# Connectivity to ec2messages endpoint  # Success # ec2messages.us-
east-2.amazonaws.com is reachable                   #
################################################################################################
# Connectivity to ssmmessages endpoint  # Success # ssmmessages.us-
east-2.amazonaws.com is reachable                   #
################################################################################################
# Connectivity to s3 endpoint           # Success # s3.us-east-2.amazonaws.com is
 reachable                            #
################################################################################################
# Connectivity to kms endpoint          # Success # kms.us-east-2.amazonaws.com is
 reachable                            #
################################################################################################
# Connectivity to logs endpoint         # Success # logs.us-east-2.amazonaws.com is
 reachable                           #
################################################################################################
# Connectivity to monitoring endpoint   # Success # monitoring.us-
east-2.amazonaws.com is reachable                    #
################################################################################################
# AWS Credentials                       # Success # Credentials are for
                                      #
#                                       #         #
 arn:aws:sts::123456789012:assumed-role/SSM-Role/i-123abc45EXAMPLE  #
```

```
#                                     #         #  and will expire at 2021-09-02
 13:24:42 +0000 UTC                  #
#############################################################################################
# Agent service                       # Success # Agent service is running and is
 running as expected user            #
#############################################################################################
# Proxy configuration                 # Skipped # No proxy configuration detected
                                     #
#############################################################################################
# Windows sysprep image state         # Success # Windows image state value is at
 desired value IMAGE_STATE_COMPLETE  #
#############################################################################################
# SSM Agent version                   # Success # SSM Agent version is 3.2.815.0,
 latest agent version in us-east-2   #
#                                     #         # is 3.2.985.0
                                     #
#############################################################################################
```

Die folgende Tabelle enthält zusätzliche Details für jede der von `ssm-cli` ausgeführten Überprüfungen.

`ssm-cli`-Diagnoseprüfungen

Check	Details
Amazon-EC2-Instance-Metadaten-Service	Gibt an, ob der verwaltete Knoten den Metadaten-Service erreichen kann. Ein fehlgeschlagener Test deutet auf ein Konnektivitätsproblem zu `http://169.254.169.254` hin, das durch das lokale Routing, den Proxy oder die Firewall- und Proxy-Konfigurationen des Betriebssystems verursacht werden kann.
Hybrid-Instance-Registrierung	Zeigt an, ob SSM Agent über eine Hybrid-Aktivierung registriert ist.
Konnektivität mit `ssm`-Endpunkt	Zeigt an, ob der Knoten in der Lage ist, die Service-Endpunkte für Systems Manager auf TCP-Port 443 zu erreichen. Ein fehlgesch

Check	Details
	lagener Test weist auf Konnektivitätsprobleme mit `https://ssm.` *region* `.amazonaws.com` hin, je nach der AWS-Region, in der sich der Knoten befindet. Konnektivitätsprobleme können durch die VPC-Konfiguration verursacht werden, einschließlich Sicherheitsgruppen, Netzwerkzugriffs-Kontrolllisten, Routing-Tabellen oder OS-Firewalls und Proxys.
Konnektivität mit `ec2messages` -Endpunkt	Zeigt an, ob der Knoten in der Lage ist, die Service-Endpunkte für Systems Manager auf TCP-Port 443 zu erreichen. Ein fehlgeschlagener Test weist auf Konnektivitätsprobleme mit `https://ec2messages.` *region* `.amazonaws.com` hin, je nach der AWS-Region, in der sich der Knoten befindet. Konnektivitätsprobleme können durch die VPC-Konfiguration verursacht werden, einschließlich Sicherheitsgruppen, Netzwerkzugriffs-Kontrolllisten, Routing-Tabellen oder OS-Firewalls und Proxys.
Konnektivität mit `ssmmessages` -Endpunkt	Zeigt an, ob der Knoten in der Lage ist, die Service-Endpunkte für Systems Manager auf TCP-Port 443 zu erreichen. Ein fehlgeschlagener Test weist auf Konnektivitätsprobleme mit `https://ssmmessages.` *region* `.amazonaws.com` hin, je nach der AWS-Region, in der sich der Knoten befindet. Konnektivitätsprobleme können durch die VPC-Konfiguration verursacht werden, einschließlich Sicherheitsgruppen, Netzwerkzugriffs-Kontrolllisten, Routing-Tabellen oder OS-Firewalls und Proxys.

Check	Details
Konnektivität mit s3-Endpunkt	Zeigt an, ob der Knoten in der Lage ist, den Service-Endpunkt für Amazon Simple Storage Service auf TCP-Port 443 zu erreichen. Ein fehlgeschlagener Test weist auf Konnektivitätsprobleme mit `https://s3.` *`region`*`.amazonaws.com` hin, je nach der AWS-Region, in der sich der Knoten befindet. Eine Verbindung zu diesem Endpunkt ist nicht erforderlich, damit ein Knoten in Ihrer Liste der verwalteten Knoten erscheint.
Konnektivität mit kms-Endpunkt	Zeigt an, ob der Knoten in der Lage ist, den Service-Endpunkt für AWS Key Management Service auf TCP-Port 443 zu erreichen. Ein fehlgeschlagener Test weist auf Konnektivitätsprobleme mit `https://kms.` *`region`*`.amazonaws.com` hin, je nach der AWS-Region, in der sich der Knoten befindet. Eine Verbindung zu diesem Endpunkt ist nicht erforderlich, damit ein Knoten in Ihrer Liste der verwalteten Knoten erscheint.
Konnektivität mit logs-Endpunkt	Zeigt an, ob der Knoten in der Lage ist, den Service-Endpunkt für Amazon CloudWatch Logs am TCP-Port 443 zu erreichen. Ein fehlgeschlagener Test weist auf Konnektivitätsprobleme mit `https://logs.` *`region`*`.amazonaws.com` hin, je nach der AWS-Region, in der sich der Knoten befindet. Eine Verbindung zu diesem Endpunkt ist nicht erforderlich, damit ein Knoten in Ihrer Liste der verwalteten Knoten erscheint.

Check	Details
Konnektivität mit `monitoring` -Endpunkt	Zeigt an, ob der Knoten in der Lage ist, den Service-Endpunkt für Amazon CloudWatch auf TCP-Port 443 zu erreichen. Ein fehlgeschlagener Test weist auf Konnektivitätsprobleme mit `https://monitoring.`*`region`*`.amazonaws.com` hin, je nach der AWS-Region, in der sich der Knoten befindet. Eine Verbindung zu diesem Endpunkt ist nicht erforderlich, damit ein Knoten in Ihrer Liste der verwalteten Knoten erscheint.
AWS-Anmeldedaten	Zeigt an, ob SSM Agent über die erforderlichen Anmeldeinformationen auf der Grundlage des IAM-Instance-Profils (für EC2-Instances) oder der IAM-Servicerolle (für Nicht-EC2-Maschinen) verfügt, die mit der Maschine verbunden sind. Ein fehlgeschlagener Test zeigt an, dass der Maschine kein IAM-Instance-Profil oder keine IAM-Servicerolle zugeordnet ist, oder dass sie nicht die erforderlichen Berechtigungen für Systems Manager enthält.
Agent-Service	Zeigt an, ob der SSM Agent-Service läuft und ob der Service als Root für Linux oder macOS bzw. als SYSTEM für Windows Server läuft. Ein fehlgeschlagener Test zeigt an, dass der SSM Agent-Service nicht ausgeführt oder nicht als Root oder SYSTEM ausgeführt wird.
Proxykonfiguration	Gibt an, ob SSM Agent konfiguriert ist, einen Proxy zu verwenden.

Check	Details
Sysprep-Image-Status (nur Windows)	Zeigt den Status von `Sysprep` auf dem Knoten an. SSM Agent wird nicht auf dem Knoten gestartet, wenn der `Sysprep`-Status einen anderen Wert als `IMAGE_STATE_COMPLETE` hat.
SSM Agent-Version	Zeigt an, ob die neueste verfügbare Version von SSM Agent installiert ist.

AWS Systems Manager Hybride Aktivierungen

Um EC2 Maschinen für die Verwendung AWS Systems Manager in einer Hybrid- und Multicloud-Umgebung zu konfigurieren, erstellen Sie eine Hybrid-Aktivierung. Als verwaltete Knoten werden u. a. folgende Typen unterstützt, die keine EC2 Maschinen sind:

- Server in Ihren eigenen Räumlichkeiten (On-Premises-Server)
- AWS IoT Greengrass Kerngeräte
- AWS IoT und Geräte, die nicht zu den AWS Edge-Geräten gehören
- Virtuelle Maschinen (VMs), auch VMs in anderen Cloud-Umgebungen

Wenn Sie den Befehl create-activation ausführen, um einen Hybrid-Aktivierungsprozess zu starten, erhalten Sie in der Befehlsantwort einen Aktivierungscode und eine ID. Anschließend fügen Sie den Aktivierungscode und die ID dem Befehl zur Installation von SSM Agent auf der Maschine bei, wie in Schritt 3 von Installation von SSM Agent auf hybriden Linux-Knoten und Schritt 4 von Installation von SSM Agent auf hybriden Windows Server-Knoten beschrieben.

Dieser Aktivierungsprozess gilt für alle Typen, die keine EC2 Maschinen sind, mit Ausnahme von AWS IoT Greengrass Kerngeräten. Weitere Informationen zum Konfigurieren von AWS IoT Greengrass -Core-Geräten für Systems Manager finden Sie unter Verwalten von Edge-Geräten mit Systems Manager.

> **Note**
>
> Support wird derzeit nicht für EC2 macOS Maschinen bereitgestellt.

Üner Systems Manager Instances-Kontingente

AWS Systems Manager bietet eine Stufe „Standard-Instances“ und eine Stufe „Advanced-Instances“. Beide unterstützen verwaltete Knoten in Ihrer Hybrid- und Multi-Cloud-Umgebung. Die Stufe „Standard-Instances“ ermöglicht es Ihnen, maximal 1.000 Maschinen pro Person zu registrieren. AWS-Konto AWS-Region Wenn Sie mehr als 1 000 Maschinen in einem einzigen Konto und einer Region anmelden müssen, verwenden Sie das Advanced-Instances-Kontingent. Sie können im Advanced-Instances-Kontingent so viele verwaltete Knoten erstellen, wie Sie möchten. Alle verwalteten Knoten, die für Systems Manager konfiguriert sind, werden auf pay-per-use Basis von Preisen berechnet. Weitere Informationen über das Aktivieren des Advanced-Instances-Kontingent finden Sie unter Aktivieren des Kontingents für erweiterte Instances. Weitere Informationen über die Preise finden Sie unter AWS Systems Manager – Preise.

Beachten Sie die folgenden zusätzlichen Informationen zur Ebene für Standard-Instances und zur Ebene für erweiterte Instances:

- Mithilfe von Advanced Instances können Sie in einer Hybrid- und Multi-Cloud-Umgebung auch eine Verbindung zu Ihren EC2 Nicht-Nodes AWS Systems Manager Session Manager herstellen. Session Managerbietet interaktiven Shell-Zugriff auf Ihre Instanzen. Weitere Informationen finden Sie unter AWS Systems Manager Session Manager.
- Das Kontingent für Standardinstanzen gilt auch für EC2 Instanzen, die eine lokale Systems Manager Manager-Aktivierung verwenden (was kein übliches Szenario ist).
- Um von Microsoft veröffentlichte Anwendungen auf lokalen Instanzen virtueller Maschinen (VMs) zu patchen, aktivieren Sie die Stufe „Advanced-Instances“. Die Nutzung des Advanced-Instances-Kontingents ist kostenpflichtig. Für Patch-Anwendungen, die von Microsoft auf Amazon Elastic Compute Cloud (Amazon EC2) -Instances veröffentlicht wurden, fallen keine zusätzlichen Gebühren an. Weitere Informationen finden Sie unter Informationen zum Patchen von Anwendungen, die von Microsoft unter Windows Server veröffentlicht wurden.

AWS Systems Manager-Bestand

AWS Systems Manager Inventar bietet Einblick in Ihre AWS Computerumgebung. Mit Inventory können Sie Metadaten aus Ihren verwalteten Knoten erfassen. Sie können diese Metadaten in einem zentralen Amazon Simple Storage Service (Amazon S3)-Bucket speichern und dann die integrierten Tools nutzen, um Daten abzufragen und schnell zu ermitteln, welche Knoten die Software ausführen, welche Konfigurationen im Rahmen Ihrer Software-Richtlinie erforderlich sind und welche Knoten aktualisiert werden müssen. Sie können Inventory mit nur einem Klick für all Ihre verwalteten Knoten

konfigurieren. Sie können mithilfe von Amazon Athena auch Inventardaten von mehreren AWS-Konten Geräten konfigurieren AWS-Regionen und anzeigen. Um mit Inventory zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Inventory.

Wenn die vorkonfigurierten, von Systems Manager Inventory Inventory erfassten Metadaten nicht Ihren Anforderungen entsprechen, können Sie einen benutzerdefinierten Bestand erstellen. Beim benutzerdefinierten Bestand handelt es sich lediglich um eine JSON-Datei mit Informationen, die Sie bereitstellen und zum verwalteten Knoten in einem bestimmten Verzeichnis hinzufügen. Bei der Datenerfassung erfasst Systems Manager Inventory diese Daten für den benutzerdefinierten Bestand. Beispiel: Wenn Sie ein großes Rechenzentrum betreiben, können Sie die Rack-Standorte der einzelnen Server als benutzerdefinierten Bestand angeben. Anschließend können Sie beim Anzeigen anderer Bestandsdaten die Daten im Rack-Bereich anzeigen.

> **Important**
>
> Systems Manager Inventory erfasst nur Metadaten von Ihren verwalteten Knoten. Inventory greift nicht auf proprietäre Informationen oder Daten zu.

In der folgenden Tabelle werden die Arten von Daten beschrieben, die Sie mit Systems Manager Inventory erfassen können. Außerdem werden darin verschiedene Angebote für die gezielte Erfassung von Knoten sowie die Erfassungsintervalle beschrieben, die angegeben werden können.

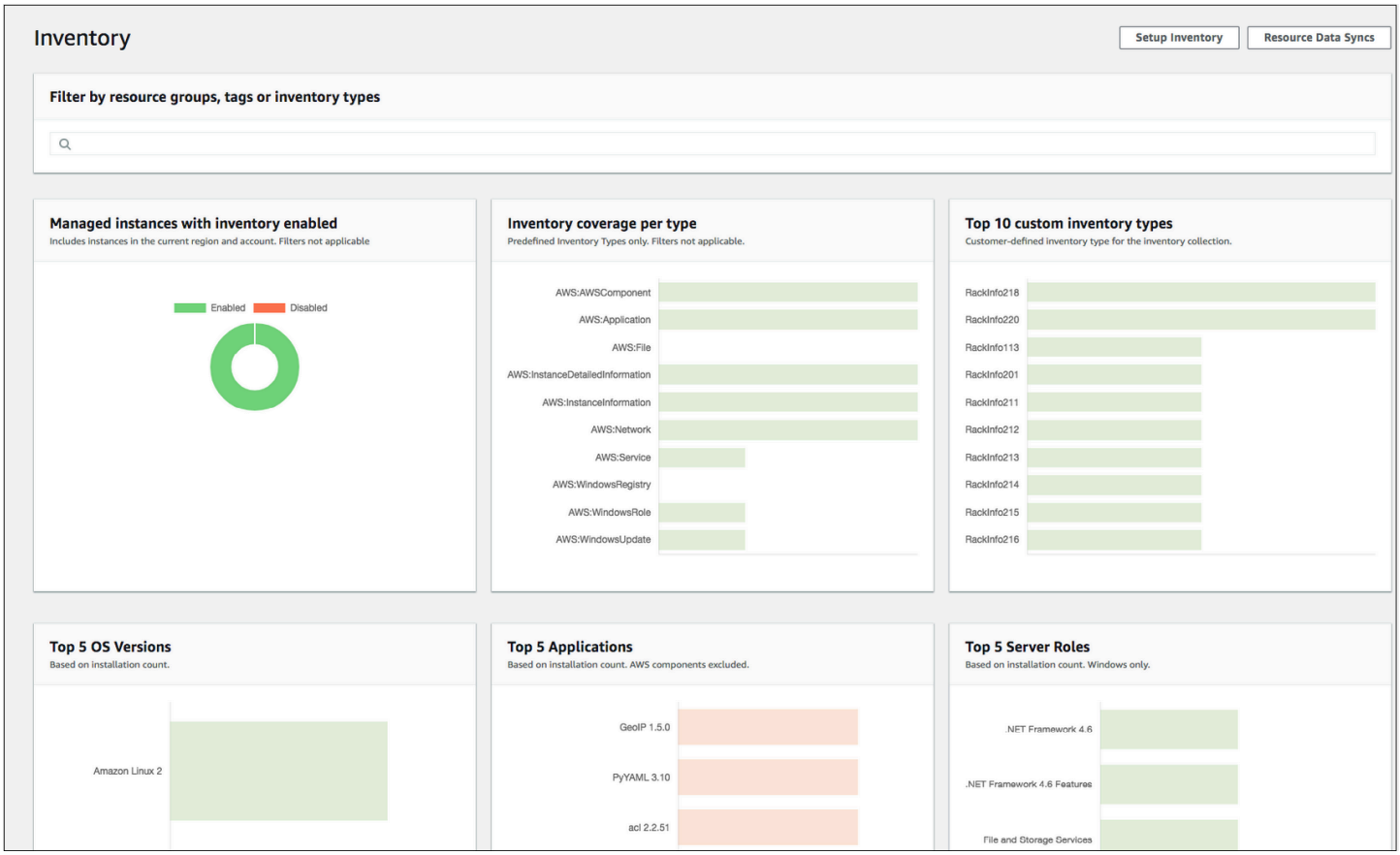
Konfiguration	Details
Metadatentypen	Sie können Inventory so konfigurieren, dass die folgenden Typen von Daten erfasst werden: • Anwendungen: Anwendungsnamen, Herausgeber, Versionen usw. • AWS Komponenten: EC2 Treiber, Agenten, Versionen usw. • Dateien: Name, Größe, Version, Installat ionsdatum, Änderung und Zeitpunkt der letzten Zugriffe usw. • Netzwerkkonfiguration: IP-Adresse, MAC-Adresse, DNS-Gateway, Subnetzmaske usw.

Konfiguration	Details
	• Windows-Updates: Hotfix-ID, installiert durch, Installationsdatum usw. • Instanzdetails: CPUModel CPUCores, CPUs, CPUSpeedMHz, CPUSockets, CPUHyperThreadEnabled, OSService Paket usw. • Services: Name, Anzeigename, Status, abhängige Services, Servicetyp, Starttyp usw. • Tags: Tags, die Ihren Knoten zugewiesen werden. • Windows-Registry: Registry-Schlüsselpfad, Wertname, Werttyp und Wert. • Windows-Rollen: Name, Anzeigename, Pfad, Funktionstyp, Installationsstatus usw. • Custom inventory (Benutzerdefinierter Bestand): Metadaten, die einem verwalteten Knoten zugewiesen wurden, wie in Arbeiten mit benutzerdefiniertem Bestand beschrieben. **Note** Wie Sie eine Liste aller durch Inventory erfassten Metadaten anzeigen, lesen Sie nach unter Metadaten-Erfassung durch Inventory

Konfiguration	Details
Zu erfassende Knoten	Sie können wählen, ob Sie alle verwaltet en Knoten in Ihren AWS-Konto individue ll ausgewählten Knoten oder Zielgruppen von Knoten mithilfe von Tags inventarisieren möchten. Weitere Informationen zur Erfassung von Bestandsdaten aller verwalteten Knoten finden Sie unter Inventarisieren Sie alle verwalteten Knoten in Ihrem AWS-Konto.
Wann Daten erfasst werden sollen	Sie können ein Intervall für die Erfassung in Minuten, Stunden und Tagen angeben. Das kürzeste mögliche Erfassungsintervall ist alle 30 Minuten.

> **Note**
>
> Abhängig von der Menge der erfassten Daten kann es einige Minuten in Anspruch nehmen, bis die Daten vom System in der Ausgabe bereitgestellt werden können, die Sie angegeben haben. Nachdem die Informationen gesammelt wurden, werden die Daten über einen sicheren HTTPS-Kanal an einen AWS Klartext-Speicher gesendet, auf den nur von Ihrem aus zugegriffen werden kann. AWS-Konto

Sie können die Daten in der Systems Manager-Konsole auf der Seite Inventory anzeigen. Diese enthält mehrere vordefinierte Karten zur Datenabfrage.



> **Note**
>
> Inventarkarten filtern automatisch von Amazon EC2 verwaltete Instances mit dem Status „Beendet“ und „Gestoppt“ heraus. Bei Knoten, die vor Ort und über AWS IoT Greengrass zentrale Geräte verwaltet werden, filtern Inventarkarten automatisch Knoten mit dem Status „Beendet“ heraus.

Wenn Sie eine Resource Data Sync zum Synchronisieren und Speichern aller Daten in einem einzelnen Amazon S3-Bucket erstellen, können Sie die Daten auf der Seite Inventory Detailed View (Detailansicht zum Bestand) detailliert anzeigen. Weitere Informationen finden Sie unter Abfragen von Bestandsdaten aus mehreren Regionen und Konten.

EventBridge Unterstützung

Dieses Systems Manager Manager-Tool wird in den EventBridge Amazon-Regeln als Ereignistyp unterstützt. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Ereignissen

mit Amazon EventBridge und Referenz: EventBridge Amazon-Ereignistypen und -muster für Systems Manager.

Inhalt

- Weitere Informationen über Systems Manager Inventory
- Einrichten von Systems Manager Inventory
- Konfigurieren der Bestandserfassung
- Abfragen von Bestandsdaten aus mehreren Regionen und Konten
- Abfragen der Bestandserfassung mithilfe von Filtern
- Aggregieren von Bestandsdaten
- Arbeiten mit benutzerdefiniertem Bestand
- Anzeigen von Bestandsverlauf und Änderungsnachverfolgung
- Anhalten der Datenerfassung und Löschen von Bestandsdaten
- Zuweisen benutzerdefinierter Bestands-Metadaten zu einem verwalteten Knoten
- Verwenden von AWS CLI , um die Inventardatenerfassung zu konfigurieren
- Walkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten
- Fehlerbehebung bei Problemen mit Systems Manager Inventory

Weitere Informationen über Systems Manager Inventory

Wenn Sie AWS Systems Manager Inventory konfigurieren, geben Sie den Typ der zu erfassenden Metadaten, die verwalteten Knoten, aus denen die Metadaten erfasst werden sollen, und einen Zeitplan für die Erfassung der Metadaten an. Diese Konfigurationen werden als AWS Systems Manager State Manager-Zuordnung in Ihrem AWS-Konto gespeichert. Eine Zuordnung ist einfach eine Konfiguration.

> **Note**
>
> Inventory erfasst nur Metadaten. Es werden keine personenbezogenen oder vertraulichen Daten erfasst.

Themen

- Metadaten-Erfassung durch Inventory

- Arbeiten mit Datei- und Windows-Registrierungsbestand

Metadaten-Erfassung durch Inventory

Das folgende Beispiel zeigt die vollständige Liste der Metadaten von jedem AWS Systems Manager Inventar-Plugin erfasst wurden.

```
{
    "typeName": "AWS:InstanceInformation",
    "version": "1.0",
    "attributes":[
      { "name": "AgentType",                                "dataType" : "STRING"},
      { "name": "AgentVersion",                            "dataType" : "STRING"},
      { "name": "ComputerName",                            "dataType" : "STRING"},
      { "name": "InstanceId",                              "dataType" : "STRING"},
      { "name": "IpAddress",                               "dataType" : "STRING"},
      { "name": "PlatformName",                            "dataType" : "STRING"},
      { "name": "PlatformType",                            "dataType" : "STRING"},
      { "name": "PlatformVersion",                         "dataType" : "STRING"},
      { "name": "ResourceType",                            "dataType" : "STRING"},
      { "name": "AgentStatus",                             "dataType" : "STRING"},
      { "name": "InstanceStatus",                          "dataType" : "STRING"}
    ]
  },
  {
    "typeName" : "AWS:Application",
    "version": "1.1",
    "attributes":[
      { "name": "Name",               "dataType": "STRING"},
      { "name": "ApplicationType",    "dataType": "STRING"},
      { "name": "Publisher",          "dataType": "STRING"},
      { "name": "Version",            "dataType": "STRING"},
      { "name": "Release",            "dataType": "STRING"},
      { "name": "Epoch",              "dataType": "STRING"},
      { "name": "InstalledTime",      "dataType": "STRING"},
      { "name": "Architecture",       "dataType": "STRING"},
      { "name": "URL",                "dataType": "STRING"},
      { "name": "Summary",            "dataType": "STRING"},
      { "name": "PackageId",          "dataType": "STRING"}
    ]
  },
  {
    "typeName" : "AWS:File",
```

```
    "version": "1.0",
    "attributes":[
      { "name": "Name",               "dataType": "STRING"},
      { "name": "Size",           "dataType": "STRING"},
      { "name": "Description",        "dataType": "STRING"},
      { "name": "FileVersion",        "dataType": "STRING"},
      { "name": "InstalledDate",      "dataType": "STRING"},
      { "name": "ModificationTime",   "dataType": "STRING"},
      { "name": "LastAccessTime",     "dataType": "STRING"},
      { "name": "ProductName",        "dataType": "STRING"},
      { "name": "InstalledDir",       "dataType": "STRING"},
      { "name": "ProductLanguage",    "dataType": "STRING"},
      { "name": "CompanyName",        "dataType": "STRING"},
      { "name": "ProductVersion",       "dataType": "STRING"}
    ]
  },
  {
    "typeName": "AWS:AWSComponent",
    "version": "1.0",
    "attributes":[
      { "name": "Name",               "dataType": "STRING"},
      { "name": "ApplicationType",    "dataType": "STRING"},
      { "name": "Publisher",          "dataType": "STRING"},
      { "name": "Version",            "dataType": "STRING"},
      { "name": "InstalledTime",      "dataType": "STRING"},
      { "name": "Architecture",       "dataType": "STRING"},
      { "name": "URL",                "dataType": "STRING"}
    ]
  },
  {
    "typeName": "AWS:WindowsUpdate",
    "version":"1.0",
    "attributes":[
      { "name": "HotFixId",           "dataType": "STRING"},
      { "name": "Description",        "dataType": "STRING"},
      { "name": "InstalledTime",      "dataType": "STRING"},
      { "name": "InstalledBy",        "dataType": "STRING"}
    ]
  },
  {
    "typeName": "AWS:Network",
    "version":"1.0",
    "attributes":[
      { "name": "Name",               "dataType": "STRING"},
```

```
      { "name": "SubnetMask",         "dataType": "STRING"},
      { "name": "Gateway",            "dataType": "STRING"},
      { "name": "DHCPServer",         "dataType": "STRING"},
      { "name": "DNSServer",          "dataType": "STRING"},
      { "name": "MacAddress",         "dataType": "STRING"},
      { "name": "IPV4",               "dataType": "STRING"},
      { "name": "IPV6",               "dataType": "STRING"}
    ]
  },
  {
    "typeName": "AWS:PatchSummary",
    "version":"1.0",
    "attributes":[
      { "name": "PatchGroup",                      "dataType": "STRING"},
      { "name": "BaselineId",                      "dataType": "STRING"},
      { "name": "SnapshotId",                      "dataType": "STRING"},
      { "name": "OwnerInformation",                "dataType": "STRING"},
      { "name": "InstalledCount",                  "dataType": "NUMBER"},
      { "name": "InstalledPendingRebootCount",     "dataType": "NUMBER"},
      { "name": "InstalledOtherCount",             "dataType": "NUMBER"},
      { "name": "InstalledRejectedCount",          "dataType": "NUMBER"},
      { "name": "NotApplicableCount",              "dataType": "NUMBER"},
      { "name": "UnreportedNotApplicableCount",    "dataType": "NUMBER"},
      { "name": "MissingCount",                    "dataType": "NUMBER"},
      { "name": "FailedCount",                     "dataType": "NUMBER"},
      { "name": "OperationType",                   "dataType": "STRING"},
      { "name": "OperationStartTime",              "dataType": "STRING"},
      { "name": "OperationEndTime",                "dataType": "STRING"},
      { "name": "InstallOverrideList",             "dataType": "STRING"},
      { "name": "RebootOption",                    "dataType": "STRING"},
      { "name": "LastNoRebootInstallOperationTime", "dataType": "STRING"},
      { "name": "ExecutionId",                     "dataType": "STRING",
   "isOptional": "true"},
      { "name": "NonCompliantSeverity",            "dataType": "STRING",
   "isOptional": "true"},
      { "name": "SecurityNonCompliantCount",       "dataType": "NUMBER",
   "isOptional": "true"},
      { "name": "CriticalNonCompliantCount",       "dataType": "NUMBER",
   "isOptional": "true"},
      { "name": "OtherNonCompliantCount",          "dataType": "NUMBER",
   "isOptional": "true"}
    ]
  },
  {
```

```
    "typeName": "AWS:PatchCompliance",
    "version":"1.0",
    "attributes":[
      { "name": "Title",                       "dataType": "STRING"},
      { "name": "KBId",                        "dataType": "STRING"},
      { "name": "Classification",              "dataType": "STRING"},
      { "name": "Severity",                    "dataType": "STRING"},
      { "name": "State",                       "dataType": "STRING"},
      { "name": "InstalledTime",               "dataType": "STRING"}
    ]
  },
  {
    "typeName": "AWS:ComplianceItem",
    "version":"1.0",
    "attributes":[
      { "name": "ComplianceType",              "dataType": "STRING",
"isContext": "true"},
      { "name": "ExecutionId",                 "dataType": "STRING",
"isContext": "true"},
      { "name": "ExecutionType",               "dataType": "STRING",
"isContext": "true"},
      { "name": "ExecutionTime",               "dataType": "STRING",
"isContext": "true"},
      { "name": "Id",                          "dataType": "STRING"},
      { "name": "Title",                       "dataType": "STRING"},
      { "name": "Status",                      "dataType": "STRING"},
      { "name": "Severity",                    "dataType": "STRING"},
      { "name": "DocumentName",                "dataType": "STRING"},
      { "name": "DocumentVersion",             "dataType": "STRING"},
      { "name": "Classification",              "dataType": "STRING"},
      { "name": "PatchBaselineId",             "dataType": "STRING"},
      { "name": "PatchSeverity",               "dataType": "STRING"},
      { "name": "PatchState",                  "dataType": "STRING"},
      { "name": "PatchGroup",                  "dataType": "STRING"},
      { "name": "InstalledTime",               "dataType": "STRING"},
      { "name": "InstallOverrideList",         "dataType": "STRING",
"isOptional": "true"},
      { "name": "DetailedText",                "dataType": "STRING",
"isOptional": "true"},
      { "name": "DetailedLink",                "dataType": "STRING",
"isOptional": "true"},
      { "name": "CVEIds",                      "dataType": "STRING",
"isOptional": "true"}
    ]
```

```
  },
  {
    "typeName": "AWS:ComplianceSummary",
    "version":"1.0",
    "attributes":[
      { "name": "ComplianceType",                "dataType": "STRING"},
      { "name": "PatchGroup",                    "dataType": "STRING"},
      { "name": "PatchBaselineId",               "dataType": "STRING"},
      { "name": "Status",                        "dataType": "STRING"},
      { "name": "OverallSeverity",               "dataType": "STRING"},
      { "name": "ExecutionId",                   "dataType": "STRING"},
      { "name": "ExecutionType",                 "dataType": "STRING"},
      { "name": "ExecutionTime",                 "dataType": "STRING"},
      { "name": "CompliantCriticalCount",        "dataType": "NUMBER"},
      { "name": "CompliantHighCount",            "dataType": "NUMBER"},
      { "name": "CompliantMediumCount",          "dataType": "NUMBER"},
      { "name": "CompliantLowCount",             "dataType": "NUMBER"},
      { "name": "CompliantInformationalCount",   "dataType": "NUMBER"},
      { "name": "CompliantUnspecifiedCount",     "dataType": "NUMBER"},
      { "name": "NonCompliantCriticalCount",     "dataType": "NUMBER"},
      { "name": "NonCompliantHighCount",         "dataType": "NUMBER"},
      { "name": "NonCompliantMediumCount",       "dataType": "NUMBER"},
      { "name": "NonCompliantLowCount",          "dataType": "NUMBER"},
      { "name": "NonCompliantInformationalCount", "dataType": "NUMBER"},
      { "name": "NonCompliantUnspecifiedCount",  "dataType": "NUMBER"}
    ]
  },
  {
    "typeName": "AWS:InstanceDetailedInformation",
    "version":"1.0",
    "attributes":[
      { "name": "CPUModel",                   "dataType": "STRING"},
      { "name": "CPUCores",                   "dataType": "NUMBER"},
      { "name": "CPUs",                       "dataType": "NUMBER"},
      { "name": "CPUSpeedMHz",                "dataType": "NUMBER"},
      { "name": "CPUSockets",                 "dataType": "NUMBER"},
      { "name": "CPUHyperThreadEnabled",      "dataType": "STRING"},
      { "name": "OSServicePack",              "dataType": "STRING"}
    ]
   },
   {
     "typeName": "AWS:Service",
     "version":"1.0",
     "attributes":[
```

```
        { "name": "Name",                          "dataType": "STRING"},
        { "name": "DisplayName",                   "dataType": "STRING"},
        { "name": "ServiceType",                   "dataType": "STRING"},
        { "name": "Status",                        "dataType": "STRING"},
        { "name": "DependentServices",             "dataType": "STRING"},
        { "name": "ServicesDependedOn",            "dataType": "STRING"},
        { "name": "StartType",                     "dataType": "STRING"}
     ]
    },
    {
      "typeName": "AWS:WindowsRegistry",
      "version":"1.0",
      "attributes":[
        { "name": "KeyPath",                          "dataType": "STRING"},
        { "name": "ValueName",                        "dataType": "STRING"},
        { "name": "ValueType",                        "dataType": "STRING"},
        { "name": "Value",                            "dataType": "STRING"}
      ]
    },
    {
      "typeName": "AWS:WindowsRole",
      "version":"1.0",
      "attributes":[
        { "name": "Name",                          "dataType": "STRING"},
        { "name": "DisplayName",                   "dataType": "STRING"},
        { "name": "Path",                          "dataType": "STRING"},
        { "name": "FeatureType",                   "dataType": "STRING"},
        { "name": "DependsOn",                     "dataType": "STRING"},
        { "name": "Description",                   "dataType": "STRING"},
        { "name": "Installed",                     "dataType": "STRING"},
        { "name": "InstalledState",                "dataType": "STRING"},
        { "name": "SubFeatures",                   "dataType": "STRING"},
        { "name": "ServerComponentDescriptor",     "dataType": "STRING"},
        { "name": "Parent",                        "dataType": "STRING"}
      ]
    },
    {
      "typeName": "AWS:Tag",
      "version":"1.0",
      "attributes":[
        { "name": "Key",                     "dataType": "STRING"},
        { "name": "Value",                   "dataType": "STRING"}
      ]
    },
```

```
    {
      "typeName": "AWS:ResourceGroup",
      "version":"1.0",
      "attributes":[
        { "name": "Name",                    "dataType": "STRING"},
        { "name": "Arn",                     "dataType": "STRING"}
      ]
    },
    {
      "typeName": "AWS:BillingInfo",
      "version": "1.0",
      "attributes": [
        { "name": "BillingProductId",        "dataType": "STRING"}
      ]
    }
```

Note

- Für`"typeName": "AWS:InstanceInformation"` kann der `InstanceStatus` einer der folgenden sein: Active, ConnectionLost, Stopped, Terminated (Aktiv, Verbindung abgebrochen, angehalten, beendet).
- Mit der Veröffentlichung von Version 2.5 ersetzt der RPM-Paketmanager ersetzt das Serial-Attribut durch Epoch. Das Epoch-Attribut ist eine monoton zunehmende Ganzzahl wie Serial. Wenn Sie eine Bestandsaufnahme unter Verwendung des `AWS:Application`-Typs durchführen, bedeutet ein größerer Wert für Epoch eine neuere Version. Wenn Epoch-Werte gleich oder leer sind, verwenden Sie den Wert des Version- oder Release-Attributs, um die neuere Version zu bestimmen.
- Einige Metadaten sind in Linux-Instances nicht verfügbar. Insbesondere für „typeName“: „AWS:Network“ werden die folgenden Metadatentypen für Linux-Instances noch nicht unterstützt. Sie WERDEN für Windows unterstützt.
  - { "name": "SubnetMask", "dataType": "STRING"},
  - { "name": "DHCPServer", "dataType": "STRING"},
  - { "name": "DNSServer", "dataType": "STRING"},
  - { "name": "Gateway", "dataType": "STRING"},

Arbeiten mit Datei- und Windows-Registrierungsbestand

AWS Systems Manager Inventory ermöglicht das Suchen und Inventarisieren von Dateien auf Windows Server Linux- und macOS Betriebssystemen. Sie können auch die Windows-Registry durchsuchen und inventarisieren.

Dateien: Sie können Metadaten-Informationen zu Dateien erfassen, einschließlich Dateinamen, der Erstellungszeit der Dateien, der letzten Änderungs- und Zugriffszeit der Dateien oder Dateigrößen, um nur ein paar zu nennen. Um mit der Erfassung eines Dateibestands zu beginnen, geben Sie einen Dateipfad an, in dem Sie die Inventarisierung durchführen möchten, ein oder mehrere Muster, die definieren, welche Dateitypen inventarisiert werden soll, und ob der Pfad rekursiv durchsucht werden soll. Systems Manager inventarisiert alle Datei-Metadaten für Dateien im angegebenen Pfad, die dem Muster entsprechen. Die Dateiinventarisierung verwendet die folgenden Eingangsparameter.

```
{
"Path": string,
"Pattern": array[string],
"Recursive": true,
"DirScanLimit" : number // Optional
}
```

- Path: Der Verzeichnispfad, in dem Dateien inventarisiert werden sollen. Für Windows können Sie Umgebungsvariablen wie `%PROGRAMFILES%` verwenden, solange die Variable einem einzigen Verzeichnispfad zugeordnet ist. Wenn Sie beispielsweise %PATH% verwenden, das auf mehreren Verzeichnispfade abgebildet wird, wirft Inventory einen Fehler auf.
- Pattern: Ein Array mit zu identifizierenden Mustern.
- Recursive: Ein Boolescher Wert, der angibt, ob Inventory die Verzeichnisse rekursiv durchlaufen soll.
- DirScanLimit: Ein optionaler Wert, der angibt, wie viele Verzeichnisse gescannt werden sollen. Verwenden Sie diesen Parameter, um die Leistung Ihrer verwalteten Knoten möglichst wenig zu beeinträchtigen. Inventory scannt maximal 5.000 Verzeichnisse.

> **Note**
>
> Inventory erfasst Metadaten für maximal 500 Dateien für alle angegebenen Pfade.

Hier finden Sie einige Beispiele, wie Sie die Parameter angeben, wenn Sie eine Inventarisierung von Dateien vornehmen wollen.

- Unter Linux und macOS erfassen Sie Metadaten von .sh-Dateien im Verzeichnis `/home/ec2-user` ohne die Unterverzeichnisse.

```
[{"Path":"/home/ec2-user","Pattern":["*.sh", "*.sh"],"Recursive":false}]
```

- Unter Windows erfassen Sie rekursiv Metadaten aller „.exe“-Dateien im Ordner Programme, einschließlich der Unterverzeichnisse.

```
[{"Path":"C:\Program Files","Pattern":["*.exe"],"Recursive":true}]
```

- Unter Windows erfassen Sie Metadaten bestimmter Log-Muster.

```
[{"Path":"C:\ProgramData\Amazon","Pattern":["*amazon*.log"],"Recursive":true}]
```

- Beschränken Sie die Anzahl der Verzeichnisse, wenn Sie eine rekursive Erfassung durchführen.

```
[{"Path":"C:\Users","Pattern":["*.ps1"],"Recursive":true, "DirScanLimit": 1000}]
```

Windows Registry: Sie können Schlüssel und Werte der Windows Registry erfassen. Sie können einen Schlüssel-Pfad auswählen und alle Schlüssel und Werte rekursiv erfassen. Sie können auch einen bestimmten Registrierungsschlüssel und seinen Wert für einen bestimmten Pfad erfassen. Inventory erfasst den Schlüsselpfad, den Namen, Typ und Wert.

```
{
"Path": string,
"Recursive": true,
"ValueNames": array[string] // optional
}
```

- Path: Der Pfad zum Registry-Schlüssel.
- Recursive: Ein Boolescher Wert, der angibt, ob Inventory die Registry-Pfade rekursiv durchlaufen soll.
- ValueNames: Eine Reihe von Wertnamen für die Inventarisierung von Registrierungsschlüsseln. Wenn Sie diesen Parameter verwenden, inventarisiert Systems Manager nur die angegebenen Wertenamen für den angegebenen Pfad.

> **Note**
>
> Inventory erfasst Metadaten für maximal 250 Registry-Schlüsselwerte für alle angegebenen Pfade.

Hier finden Sie einige Beispiele, wie Sie die Parameter angeben, wenn Sie eine Inventarisierung der Windows Registry vornehmen wollen.

- Erfassen Sie alle Schlüssel und Werte rekursiv für einen bestimmten Pfad.

```
[{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Amazon","Recursive": true}]
```

- Erfassen Sie alle Schlüssel und Werte für einen bestimmten Pfad (die rekursive Suche ist deaktiviert).

```
[{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Intel\PSIS\PSIS_DECODER", "Recursive": false}]
```

- Erfasst einen bestimmten Schlüssel unter Verwendung der Option `ValueNames`.

```
{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Amazon\MachineImage","ValueNames":["AMIName"]}
```

Einrichten von Systems Manager Inventory

Bevor Sie AWS Systems Manager Inventory zum Sammeln von Metadaten über die Anwendungen, Services, AWS-Komponenten und mehr verwenden, die auf Ihren verwalteten Knoten ausgeführt werden, empfehlen wir, die Ressourcen-Datensynchronisierung zu konfigurieren, um die Speicherung Ihrer Bestandsdaten in einem einzigen Amazon Simple Storage Service (Amazon S3)-Bucket zu zentralisieren. Wir empfehlen außerdem, die Amazon EventBridge-Überwachung von Bestandsereignissen zu konfigurieren. Diese Prozesse erleichtern das Anzeigen und Verwalten von Bestandsdaten und -sammlungen.

Themen

- Erstellen einer Resource Data Sync für Inventory
- Verwenden von EventBridge zum Überwachen von Inventory-Ereignissen

Erstellen einer Resource Data Sync für Inventory

In diesem Thema wird beschrieben, wie Sie die Ressourcendatensynchronisierung für AWS Systems Manager -Inventar einrichten und konfigurieren. Informationen zu Resource Data Sync für Systems Manager Explorer finden Sie unter Einrichten von Systems Manager Explorer, um Daten aus mehreren Konten und Regionen anzuzeigen.

Über Resource Data Sync

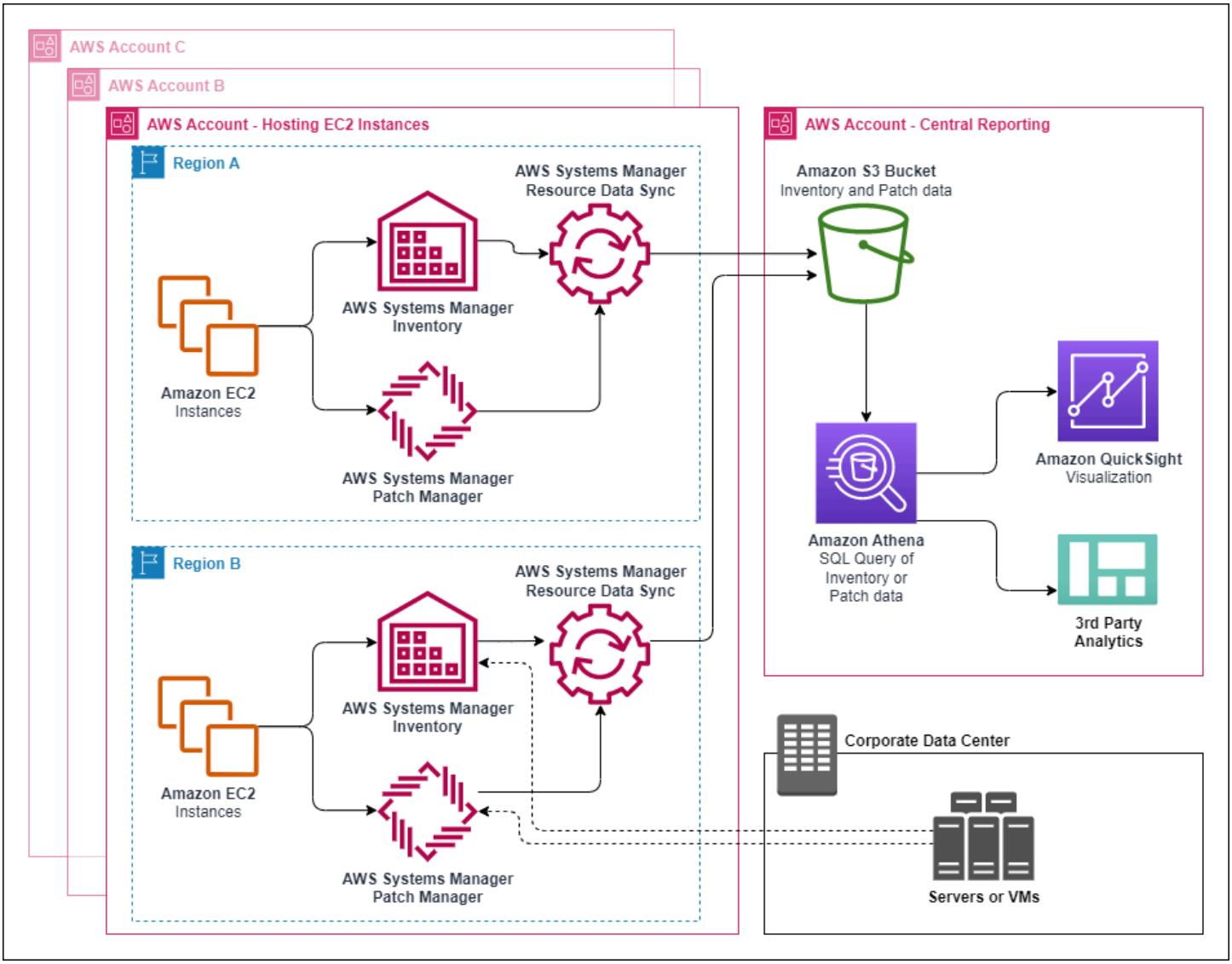
Sie können mit der Ressourcen-Datensynchronisierung von Systems Manager Bestandsdaten aus allen Ihren verwalteten Knoten an einen einzelnen Amazon Simple Storage Service (Amazon S3)-Bucket senden. Resource Data Sync aktualisiert die Daten dann automatisch, wenn neue Bestandsdaten erfasst werden. Da alle Inventardaten in einem Amazon S3 S3-Ziel-Bucket gespeichert sind, können Sie Dienste wie Amazon Athena und Amazon Quick Suite verwenden, um die aggregierten Daten abzufragen und zu analysieren.

Sie können Inventory z. B. so konfigurieren, dass Daten über das Betriebssystem (OS) und die Anwendungen erfasst werden, die auf eine Flotte von 150 verwalteten Knoten ausgeführt werden. Einige dieser Knoten befinden sich in einem lokalen Rechenzentrum, andere laufen in Amazon Elastic Compute Cloud (Amazon EC2) über mehrere AWS-Regionen. Wenn Sie die Ressourcen-Datensynchronisierung nicht konfiguriert haben, müssen Sie die erfassten Bestandsdaten für jeden verwalteten Knoten manuell einholen oder Sie müssen entsprechende Skripts erstellen, die diese Informationen sammeln. Anschließend müssten Sie die Daten in eine Anwendung portieren, um Abfragen ausführen und diese analysieren zu können.

Mit der Ressourcen-Datensynchronisierung führen Sie eine einmalige Operation aus, die alle Bestandsdaten von allen Ihren verwalteten Knoten synchronisiert. Wenn die Synchronisierung erfolgreich erstellt wurde, erstellt Systems Manager eine Ausgangsbasis mit allen Bestandsdaten und speichert diese in dem jeweiligen Amazon S3-Ziel-Bucket. Wenn neue Bestandsdaten erfasst werden, aktualisiert Systems Manager die Daten in dem Amazon S3-Bucket automatisch. Anschließend können Sie die Daten schnell und kostengünstig nach Amazon Athena und Amazon Quick Suite portieren.

Abbildung 1 zeigt, wie Resource Data Sync Inventardaten von Amazon EC2 und anderen Maschinentypen in einer Hybrid- und Multi-Cloud-Umgebung in einem Amazon S3 S3-Ziel-Bucket aggregiert. Dieses Diagramm zeigt auch, wie die Ressourcendatensynchronisierung mit mehreren AWS-Konten und funktioniert. AWS-Regionen

Diagramm 1: Synchronisieren von Ressourcendaten mit mehreren AWS-Konten und AWS-Regionen



Wenn Sie einen verwalteten Knoten löschen, behält die Ressourcen-Datensynchronisierung die Bestandsdatei für den gelöschten Knoten bei. Für ausgeführte Knoten überschreibt die Ressourcen-Datensynchronisierung die alte Bestandsdatei jedoch automatisch, wenn neue Dateien erstellt und in den Amazon-S3-Bucket geschrieben werden. Wenn Sie Inventaränderungen im Laufe der Zeit verfolgen möchten, können Sie den AWS Config Dienst verwenden, um den `SSM:ManagedInstanceInventory` Ressourcentyp zu verfolgen. Weitere Informationen finden Sie unter Erste Schritte mit AWS Config.

Gehen Sie wie in diesem Abschnitt beschrieben vor, um mithilfe von Amazon S3 und AWS Systems Manager Konsolen eine Ressourcendatensynchronisierung für Inventory zu erstellen. Sie können sie auch verwenden AWS CloudFormation , um eine Ressourcendatensynchronisierung zu erstellen oder zu löschen. Um sie zu verwenden CloudFormation, fügen Sie die

`AWS::SSM::ResourceDataSync`Ressource zu Ihrer CloudFormation Vorlage hinzu. Informationen dazu finden Sie in einer der folgenden Dokumentationsressourcen:

- AWS CloudFormation Ressource für die Synchronisation von Ressourcendaten in AWS Systems Manager (Blog)
- Arbeiten mit AWS CloudFormation -Vorlagen im AWS CloudFormation -Benutzerhandbuch

> **Note**
>
> Sie können AWS Key Management Service (AWS KMS) verwenden, um Inventardaten im Amazon S3 S3-Bucket zu verschlüsseln. Ein Beispiel dafür, wie Sie mithilfe von AWS Command Line Interface (AWS CLI) eine verschlüsselte Synchronisation erstellen und wie Sie mit den zentralisierten Daten in Amazon Athena und Amazon Quick Suite arbeiten, finden Sie unterWalkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten.

Bevor Sie beginnen

Bevor Sie eine Ressourcendatensynchronisierung erstellen, verwenden Sie das folgende Verfahren, um einen zentralen Amazon S3-Bucket zum Speichern aggregierter Bestandsdaten zu erstellen. Das Verfahren beschreibt, wie Sie eine Bucket-Richtlinie zuweisen, die es Systems Manager ermöglicht, Bestandsdaten aus mehreren Konten in den Bucket zu schreiben. Wenn Sie bereits über einen Amazon S3-Bucket verfügen, den Sie zum Aggregieren von Bestandsdaten für die Ressourcendatensynchronisierung verwenden möchten, müssen Sie den Bucket so konfigurieren, dass die Richtlinie im folgenden Verfahren verwendet wird.

> **Note**
>
> Systems Manager Inventory kann keine Daten zu einem angegebenen Amazon S3-Bucket hinzufügen, wenn dieser Bucket für die Verwendung von Object Lock konfiguriert ist. Stellen Sie sicher, dass der Amazon S3-Bucket, den Sie für die Resource Data Sync erstellen oder auswählen, nicht für die Verwendung von Amazon S3 Object Lock konfiguriert ist. Weitere Informationen finden Sie unter Funktionsweise von Amazon S3 Object Lock im Benutzerhandbuch zu Amazon Simple Storage Service.

So erstellen und konfigurieren Sie einen Amazon S3-Bucket für Resource Data Sync

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Erstellen Sie einen Bucket zum Speichern der aggregierten Bestandsdaten. Weitere Informationen finden Sie unter Erstellen eines Buckets im Benutzerhandbuch zu Amazon Simple Storage Service. Notieren Sie sich den Namen des Buckets und den AWS-Region Ort, an dem Sie ihn erstellt haben.
3. Klicken Sie auf die Registerkarte Permissions (Berechtigungen) und anschließend auf Bucket Policy (Bucket-Richtlinie).
4. Kopieren Sie die folgende Bucket-Richtlinie in den Richtlinien-Editor. *amzn-s3-demo-bucket*Ersetzen Sie ihn durch den Namen des S3-Buckets, den Sie erstellt haben. *account_ID_number*Ersetzen Sie es durch eine gültige AWS-Konto ID-Nummer.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "SSMBucketPermissionsCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
        },
        {
            "Sid": " SSMBucketDelivery",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*/accountid=111122223333/
*",
                "arn:aws:s3:::amzn-s3-demo-bucket/*/accountid=444455556666/
*",
```

```
                "arn:aws:s3:::amzn-s3-demo-bucket/*/accountid=123456789012/
*",
                "arn:aws:s3:::amzn-s3-demo-bucket/*/accountid=777788889999/*"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:ssm:*:111122223333:resource-
data-sync/*"
                }
            }
        }
    ]
}
```

5. Speichern Sie Ihre Änderungen.

Erstellen einer Resource Data Sync für Inventory

Führen Sie die folgenden Schritte aus, um mit der Systems Manager-Konsole eine Ressource Data Sync for Systems Manager Inventory zu erstellen. Informationen zum Erstellen einer Ressourcendatensynchronisierung mithilfe von finden Sie unterVerwenden von AWS CLI , um die Inventardatenerfassung zu konfigurieren. AWS CLI

So erstellen Sie eine Ressourcendaten-Synchronisierung

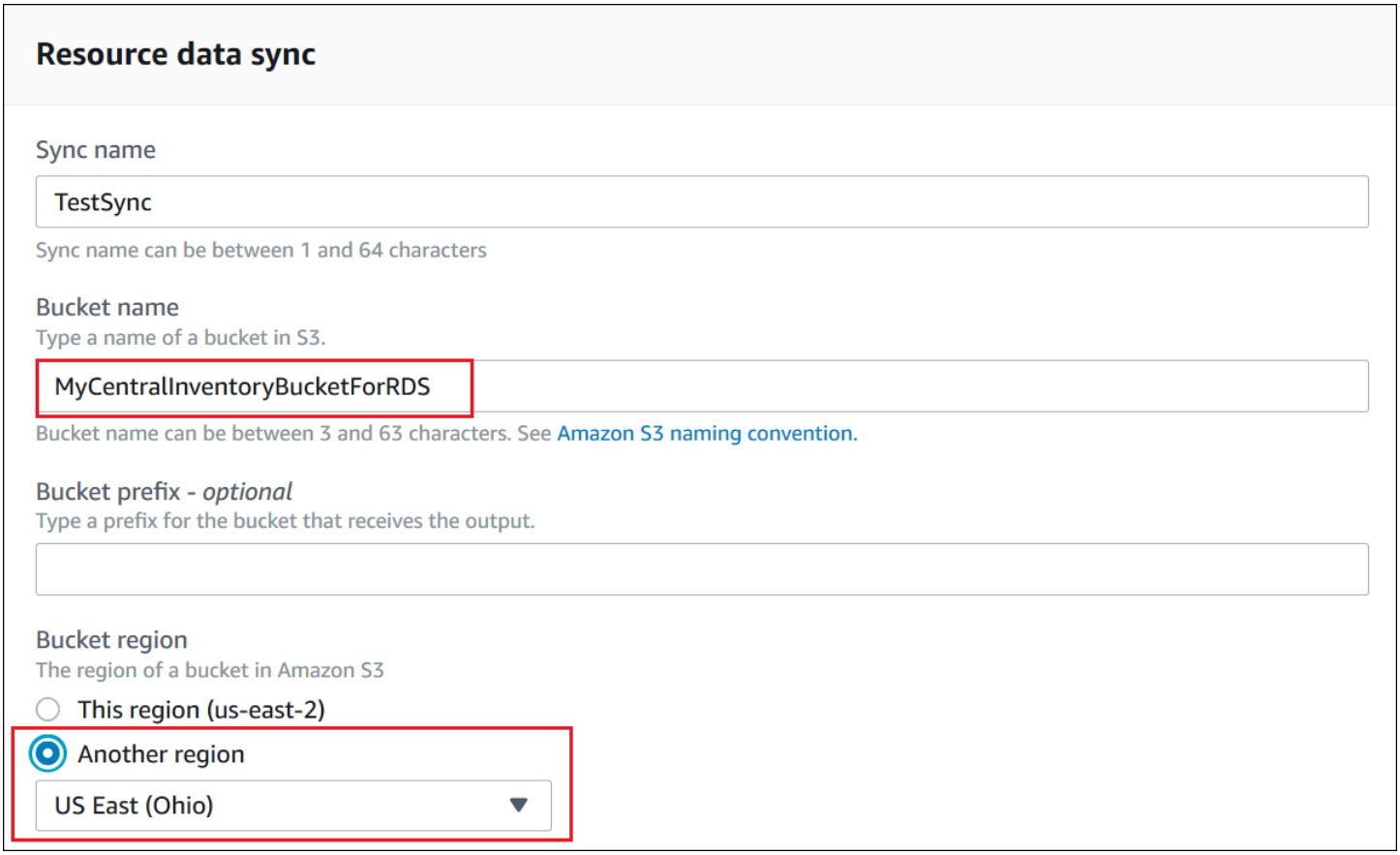
1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie im Menü Account management (Kontoverwaltung) die Option Resource Data Sync (Ressourcendaten).
4. Wählen Sie Ressourcen-Datensynchronisierung erstellen.
5. Geben Sie im Feld Sync name einen Namen für die Synchronisierungskonfiguration ein.
6. Geben Sie in das Feld Bucket name (Bucket-name) den Namen des Amazon S3-Buckets ein, das Sie mit dem Verfahren To create and configure an Amazon S3 bucket for resource data sync (Erstellen und Konfigurieren eines Amazon S3-Buckets für Ressource Data Sync) erstellt haben.

7. (Optional) Geben Sie im Feld Bucket prefix (Bucket-Präfix) den Namen eines Amazon S3-Bucket-Präfixes (Unterverzeichnis) an.
8. Wählen Sie im Feld Bucket region (Bucket-Region) die Option This region (Diese Region) aus, wenn sich der erstellte Amazon S3-Bucket in der aktuellen AWS-Region befindet. Befindet sich der erstellte Bucket in einer anderen AWS-Region, wählen Sie Another region (Andere Region) aus und geben den Namen der Region an.

   > **Note**
   >
   > Wenn sich der Synchronisierungs- und der Amazon S3-Ziel-Bucket in verschiedenen Regionen befinden, können Kosten für Datenübertragung anfallen. Weitere Informationen finden Sie unter Amazon S3 – Preise.

9. (Optional) Geben Sie im Feld KMS Key ARN (KMS-Schlüssel-ARN) einen KMS-Schlüssel-ARN zum Verschlüsseln von Bestandsdaten in Amazon S3 ein oder fügen Sie einen ein.
10. Wählen Sie Erstellen aus.

Um Inventardaten aus mehreren zu synchronisieren AWS-Regionen, müssen Sie in jeder Region eine Ressourcendatensynchronisierung einrichten. Wiederholen Sie diesen Vorgang an jedem AWS-Region Ort, an dem Sie Inventardaten sammeln und an den zentralen Amazon S3 S3-Bucket senden möchten. Wenn Sie die Synchronisierung in jeder Region erstellen, geben Sie den zentralen Amazon S3-Bucket im Bucket name (Bucket-Name) an. Verwenden Sie dann die Option Bucket region (Bucket-Region), um die Region auszuwählen, in der Sie den zentralen Amazon S3-Bucket erstellt haben, wie im folgenden Screenshot gezeigt. Wenn die Zuordnung zu Erfassen von Bestandsdaten das nächste Mal ausgeführt wird, speichert Systems Manager die Daten im zentralen Amazon S3-Bucket.



Erstellen einer Inventory Resource Data Sync für mehrere Konten, die in AWS Organizations definiert sind

Sie können Inventardaten von AWS-Konten Defined in AWS Organizations mit einem zentralen Amazon S3 S3-Bucket synchronisieren. Nachdem Sie das folgende Verfahren abgeschlossen haben, werden Bestandsdaten mit einzelnen Amazon S3-Schlüsselpräfixen im zentralen Bucket synchronisiert. Jedes key prefix steht für eine andere AWS-Konto ID.

Bevor Sie beginnen

Bevor Sie beginnen, stellen Sie sicher, dass Sie AWS-Konten in eingerichtet und konfiguriert haben AWS Organizations. Weitere Informationen finden Sie unter im AWS Organizations - Benutzerhandbuch.

Beachten Sie außerdem, dass Sie die organisationsbasierte Ressourcendatensynchronisierung für jede Ressource erstellen müssen AWS-Region und dass unter AWS-Konto definiert ist. AWS Organizations

Erstellen eines zentralen Amazon S3-Buckets

Verwenden Sie das folgende Verfahren, um einen zentralen Amazon S3-Bucket zum Speichern aggregierter Bestandsdaten zu erstellen. Das Verfahren beschreibt, wie Sie eine Bucket-Richtlinie zuweisen, die es Systems Manager ermöglicht, Bestandsdaten aus Ihrer AWS Organizations -Konto-ID in den Bucket zu schreiben. Wenn Sie bereits über einen Amazon S3-Bucket verfügen, den Sie zum Aggregieren von Bestandsdaten für die Ressourcendatensynchronisierung verwenden möchten, müssen Sie den Bucket so konfigurieren, dass die Richtlinie im folgenden Verfahren verwendet wird.

Um einen Amazon S3 S3-Bucket für die Ressourcendatensynchronisierung für mehrere Konten zu erstellen und zu konfigurieren, definiert in AWS Organizations

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Erstellen Sie einen Bucket zum Speichern der aggregierten Bestandsdaten. Weitere Informationen finden Sie unter Erstellen eines Buckets im Benutzerhandbuch zu Amazon Simple Storage Service. Notieren Sie sich den Namen des Buckets und den AWS-Region Ort, an dem Sie ihn erstellt haben.
3. Klicken Sie auf die Registerkarte Permissions (Berechtigungen) und anschließend auf Bucket Policy (Bucket-Richtlinie).
4. Kopieren Sie die folgende Bucket-Richtlinie in den Richtlinien-Editor. Ersetzen Sie *amzn-s3-demo-bucket* und *organization-id* durch den Namen des Amazon S3 S3-Buckets, den Sie erstellt haben, und durch eine gültige AWS Organizations Konto-ID.

   Optional können Sie es *bucket-prefix* durch den Namen eines Amazon S3 S3-Präfix (Unterverzeichnis) ersetzen. Wenn Sie kein Präfix erstellt haben, entfernen Sie in der folgenden Richtlinie*bucket-prefix*/aus dem ARN.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "SSMBucketPermissionsCheck",
      "Effect": "Allow",
      "Principal": {
        "Service": "ssm.amazonaws.com"
      },
      "Action": "s3:GetBucketAcl",
```

```
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
    },
    {
      "Sid": " SSMBucketDelivery",
      "Effect": "Allow",
      "Principal": {
        "Service": "ssm.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/bucket-prefix/*/accountid=*/*"
      ],
      "Condition": {
        "StringEquals": {
          "s3:x-amz-acl": "bucket-owner-full-control",
          "aws:SourceOrgID": "organization-id"
                  }
      }
    },
    {
      "Sid": " SSMBucketDeliveryTagging",
      "Effect": "Allow",
      "Principal": {
        "Service": "ssm.amazonaws.com"
      },
      "Action": "s3:PutObjectTagging",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/bucket-prefix/*/accountid=*/*"
      ]
    }
  ]
}
```

Erstellen einer Bestands-Resource Data Sync für Konten, die in AWS Organizations definiert sind

Das folgende Verfahren beschreibt, wie Sie mithilfe von eine Ressourcendatensynchronisierung für Konten erstellen, die in definiert sind AWS Organizations. AWS CLI Sie müssen den verwenden AWS CLI , um diese Aufgabe auszuführen. Sie müssen dieses Verfahren auch für jedes AWS-Region Verfahren ausführen, das AWS-Konto unter definiert ist AWS Organizations.

So erstellen Sie eine Ressourcendatensynchronisierung für ein Konto, das in AWS Organizations (AWS CLI) definiert ist

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob keine anderen auf AWS Organizations basierten Resource Data Syncs vorhanden sind. Sie können mehrere Standardsynchronisierungen haben, einschließlich mehrerer Standardsynchronisierungen und einer organisationsbasierten Synchronisierung. Sie können jedoch nur eine organisationsbasierte Ressourcendatensynchronisierung durchführen.

   ```
   aws ssm list-resource-data-sync
   ```

   Wenn der Befehl andere organisationsbasierte Ressourcendatensynchronisierungen zurückgibt, müssen Sie diese löschen oder sich dafür entscheiden, keine neue zu erstellen.

3. Führen Sie den folgenden Befehl aus, um eine Ressourcendatensynchronisierung für ein Konto zu erstellen, das in AWS Organizations definiert ist. Geben Sie für amzn-s3-demo-bucket den Namen des Amazon-S3-Buckets an, den Sie zuvor in diesem Thema erstellt haben. Wenn Sie ein Präfix (Unterverzeichnis) für Ihren Bucket erstellt haben, geben Sie diese Information für *prefix-name* an.

   ```
   aws ssm create-resource-data-sync --sync-name name --s3-
   destination "BucketName=amzn-s3-demo-bucket,Prefix=prefix-
   name,SyncFormat=JsonSerDe,Region=AWS-Region, for example us-
   east-2,DestinationDataSharing={DestinationDataSharingType=Organization}"
   ```

4. Wiederholen Sie die Schritte 2 und 3 für alle AWS-Region Bereiche AWS-Konto , in denen Sie Daten mit dem zentralen Amazon S3 S3-Bucket synchronisieren möchten.

Ressourcendatensynchronisierungen verwalten

In jedem AWS-Konto Fall können 5 Ressourcendatensynchronisierungen durchgeführt werden. AWS-Region Sie können die AWS Systems Manager Fleet Manager Konsole verwenden, um Ihre Ressourcendatensynchronisierungen zu verwalten.

Um Ressourcendatensynchronisierungen anzuzeigen

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie im Menü-Dropdown Kontoverwaltung die Option Ressourcendatensynchronisierung.
4. Wählen Sie eine Ressourcendatensynchronisierung aus der Tabelle aus, und klicken Sie dann auf Details anzeigen, um Informationen zu Ihrer Ressourcendatensynchronisierung anzuzeigen.

So löschen Sie eine Ressourcendaten-Synchronisierung

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie im Menü-Dropdown Kontoverwaltung die Option Ressourcendatensynchronisierung.
4. Wählen Sie eine Ressourcendatensynchronisierung aus der Tabelle aus, und klicken Sie dann auf Löschen.

Verwenden von EventBridge zum Überwachen von Inventory-Ereignissen

Sie können eine Regel in Amazon EventBridge konfigurieren, um ein Ereignis als Reaktion auf AWS Systems Manager-Inventory-Ressourcenstatusänderungen zu erstellen. EventBridge unterstützt Ereignisse für die folgenden Inventory-Statusänderungen. Alle Ereignisse werden auf bestmögliche Weise ausgegeben.

Benutzerdefinierter Bestandstyp für eine bestimmte Instance gelöscht: Wenn eine Regel für die Überwachung dieses Ereignisses konfiguriert ist, erstellt EventBridge ein Ereignis, wenn ein benutzerdefinierter Bestandstyp für eine bestimmte verwaltete Instance gelöscht wird. EventBridge sendet ein Ereignis pro Knoten pro benutzerdefiniertem Bestandstyp. Hier ist ein Beispielereignismuster.

```
{
    "timestampMillis": 1610042981103,
    "source": "SSM",
    "account": "123456789012",
    "type": "INVENTORY_RESOURCE_STATE_CHANGE",
    "startTime": "Jan 7, 2021 6:09:41 PM",
```

```
    "resources": [
        {
            "arn": "arn:aws:ssm:us-east-1:123456789012:managed-instance/i-12345678"
        }
    ],
    "body": {
        "action-status": "succeeded",
        "action": "delete",
        "resource-type": "managed-instance",
        "resource-id": "i-12345678",
        "action-reason": "",
        "type-name": "Custom:MyCustomInventoryType"
    }
}
```

Benutzerdefiniertes Bestandstyp-Löschereignis für alle Instances: Wenn eine Regel für die Überwachung dieses Ereignisses konfiguriert ist, erstellt EventBridge ein Ereignis, wenn ein benutzerdefinierter Bestandstyp für alle verwalteten Knoten gelöscht wird. Hier ist ein Beispielereignismuster.

```
{
    "timestampMillis": 1610042904712,
    "source": "SSM",
    "account": "123456789012",
    "type": "INVENTORY_RESOURCE_STATE_CHANGE",
    "startTime": "Jan 7, 2021 6:08:24 PM",
    "resources": [

    ],
    "body": {
        "action-status": "succeeded",
        "action": "delete-summary",
        "resource-type": "managed-instance",
        "resource-id": "",
        "action-reason": "The delete for type name Custom:SomeCustomInventoryType
 was completed. The deletion summary is: {\"totalCount\":1,\"remainingCount\":0,
\"summaryItems\":[{\"version\":\"1.1\",\"count\":1,\"remainingCount\":0}]}",
        "type-name": "Custom:MyCustomInventoryType"
    }
}
```

PutInventory ruft ein altes Schemaversionsereignis auf: Wenn eine Regel für die Überwachung dieses Ereignisses konfiguriert ist, erstellt EventBridge ein Ereignis, wenn ein PutInventory-Aufruf durchgeführt wird, der eine Schemaversion verwendet, die niedriger als das aktuelle Schema ist. Dieses Ereignis gilt für alle Inventararten. Hier ist ein Beispielereignismuster.

```
{
    "timestampMillis": 1610042629548,
    "source": "SSM",
    "account": "123456789012",
    "type": "INVENTORY_RESOURCE_STATE_CHANGE",
    "startTime": "Jan 7, 2021 6:03:49 PM",
    "resources": [
        {
            "arn": "arn:aws:ssm:us-east-1:123456789012:managed-instance/i-12345678"
        }
    ],
    "body": {
        "action-status": "failed",
        "action": "put",
        "resource-type": "managed-instance",
        "resource-id": "i-01f017c1b2efbe2bc",
        "action-reason": "The inventory item with type name
 Custom:MyCustomInventoryType was sent with a disabled schema verison 1.0. You must
 send a version greater than 1.0",
        "type-name": "Custom:MyCustomInventoryType"
    }
}
```

Informationen zum Konfigurieren von EventBridge für die Überwachung dieser Ereignisse finden Sie unter Konfigurieren von EventBridge für Systems-Manager-Ereignisse.

Konfigurieren der Bestandserfassung

In diesem Abschnitt wird beschrieben, wie Sie die AWS Systems Manager Inventarerfassung auf einem oder mehreren verwalteten Knoten mithilfe der Systems Manager Manager-Konsole konfigurieren. Ein Beispiel für die Konfiguration der Inventarerfassung mithilfe von AWS Command Line Interface (AWS CLI) finden Sie unterVerwenden von AWS CLI , um die Inventardatenerfassung zu konfigurieren.

Wenn Sie die Inventarerfassung konfigurieren, erstellen Sie zunächst eine AWS Systems Manager State Manager Zuordnung. Systems Manager erfasst die Bestandsdaten, wenn der

Zuordnungsstatus ausgeführt wird. Wenn Sie die Zuordnung nicht zuerst erstellen und versuchen, das `aws:softwareInventory` Plug-in beispielsweise mithilfe von aufzurufen, gibt das System den folgenden Fehler zurück: AWS Systems Manager Run Command `The aws:softwareInventory plugin can only be invoked via ssm-associate.`

> **Note**
>
> Beachten Sie das folgende Verhalten, wenn Sie mehrere Bestandszuordnungen für einen verwalteten Knoten erstellen:
>
> - Jedem Knoten kann eine Inventarzuordnung zugewiesen werden, die auf alle Knoten abzielt (--targets „Key=InstanceIds, Values=*“).
> - Jedem Knoten kann auch eine bestimmte Assoziation zugewiesen werden, die entweder key/value Tag-Paare oder eine Ressourcengruppe verwendet. AWS
> - Wenn einem Knoten mehrere Bestandszuordnungen zugewiesen sind, zeigt der Status Skipped (Übersprungen) für die Zuordnung an, die nicht ausgeführt wurde. Die zuletzt durchgeführte Zuordnung zeigt den aktuellen Status der Bestandszuordnung an.
> - Wenn einem Knoten mehrere Inventarzuordnungen zugewiesen sind und jede ein key/value Tag-Paar verwendet, können diese Inventarzuordnungen aufgrund des Tag-Konflikts nicht auf dem Knoten ausgeführt werden. Die Zuordnung wird immer noch auf Knoten ausgeführt, bei denen der key/value Tagkonflikt nicht vorliegt.

Bevor Sie beginnen

Führen Sie die folgenden Aufgaben aus, bevor Sie die Bestandserfassung konfigurieren.

- Aktualisieren Sie AWS Systems Manager SSM Agent die Knoten, die Sie inventarisieren möchten. Durch Ausführen der neuesten Version von SSM Agent stellen Sie sicher, dass Sie Metadaten für alle unterstützten Bestandstypen sammeln können. Informationen zur Aktualisierung von SSM Agent mithilfe von State Manager finden Sie unter Exemplarische Vorgehensweise: Automatisches Update SSM Agent mit AWS CLI.
- Stellen Sie sicher, dass Sie die Einrichtungsvoraussetzungen für Ihre Amazon Elastic Compute Cloud (Amazon EC2) -Instances und EC2 Nicht-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung erfüllt haben. Weitere Informationen finden Sie unter Einrichten von verwalteten Knoten für AWS Systems Manager.

- Stellen Sie für Windows Server Microsoft-Knoten sicher, dass Ihr verwalteter Knoten mit Windows PowerShell 3.0 (oder höher) konfiguriert ist. SSM Agentverwendet das `ConvertTo-Json` Cmdlet in PowerShell , um die Windows Update-Inventardaten in das erforderliche Format zu konvertieren.
- (Optional) Erstellen Sie eine Resource Data Sync, um Bestandsdaten zentral in einem Amazon S3-Bucket zu speichern. Resource Data Sync aktualisiert die Daten dann automatisch, wenn neue Bestandsdaten erfasst werden. Weitere Informationen finden Sie unter Walkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten.
- (Optional) Erstellen Sie eine JSON-Datei für das Erfassen des benutzerdefinierten Bestands. Weitere Informationen finden Sie unter Arbeiten mit benutzerdefiniertem Bestand.

Inventarisieren Sie alle verwalteten Knoten in Ihrem AWS-Konto

Sie können alle verwalteten Knoten in Ihrem inventarisieren, AWS-Konto indem Sie eine globale Inventarzuordnung erstellen. Eine globale Bestandszuordnung führt die folgenden Aktionen aus:

- Wendet die globale Inventarkonfiguration (Zuordnung) automatisch auf alle vorhandenen verwalteten Knoten in Ihrem an AWS-Konto. Verwaltete Knoten, die bereits über eine Bestandszuordnung verfügen, werden übersprungen, wenn die globale Bestandszuordnung angewendet wurde und ausgeführt wird. Wenn ein Knoten ausgelassen wird, zeigt die detaillierte Statusmeldung `Overridden By Explicit Inventory Association` an. Diese Knoten werden von der globalen Zuordnung übersprungen, sie melden aber immer noch den Bestand, wenn sie ihre zugewiesene Bestandszuordnung ausführen.
- Fügt neue Knoten, die in Ihrem erstellt wurden, automatisch AWS-Konto zur globalen Inventarzuordnung hinzu.

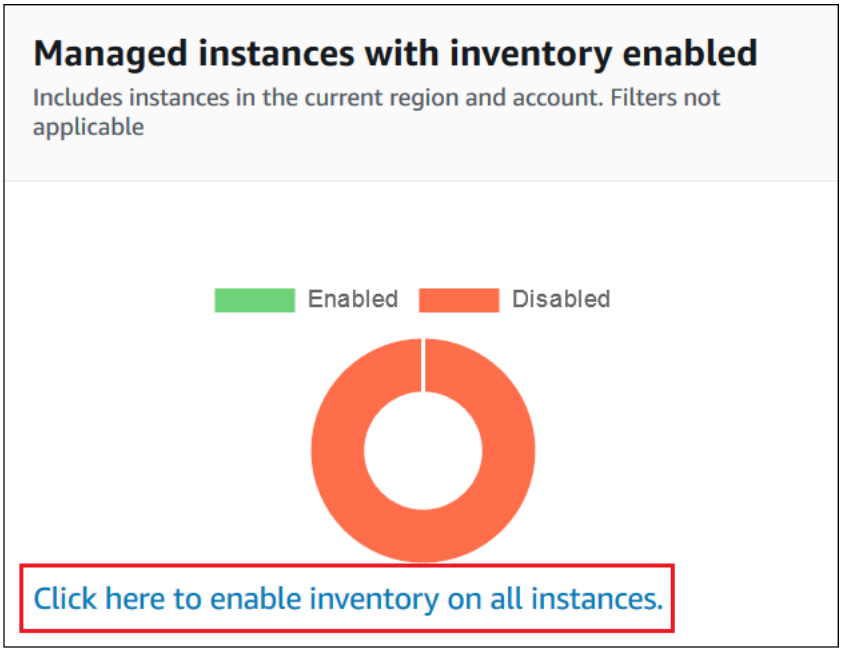
> **Note**
>
> - Wenn ein verwalteter Knoten für die globale Bestandszuordnung konfiguriert ist und Sie diesem Knoten eine bestimmte Zuordnung zuweisen, dann stellt Systems Manager Inventory die globale Zuordnung zurück und wendet die spezifische Zuordnung an.
> - Globale Bestandszuordnungen sind in SSM Agent-Version 2.0.790.0 oder höher verfügbar. Weitere Informationen zur Aktualisierung von SSM Agent auf Ihren Knoten finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command.

Konfigurieren der Bestandserfassung mit einem Klick (Konsole)

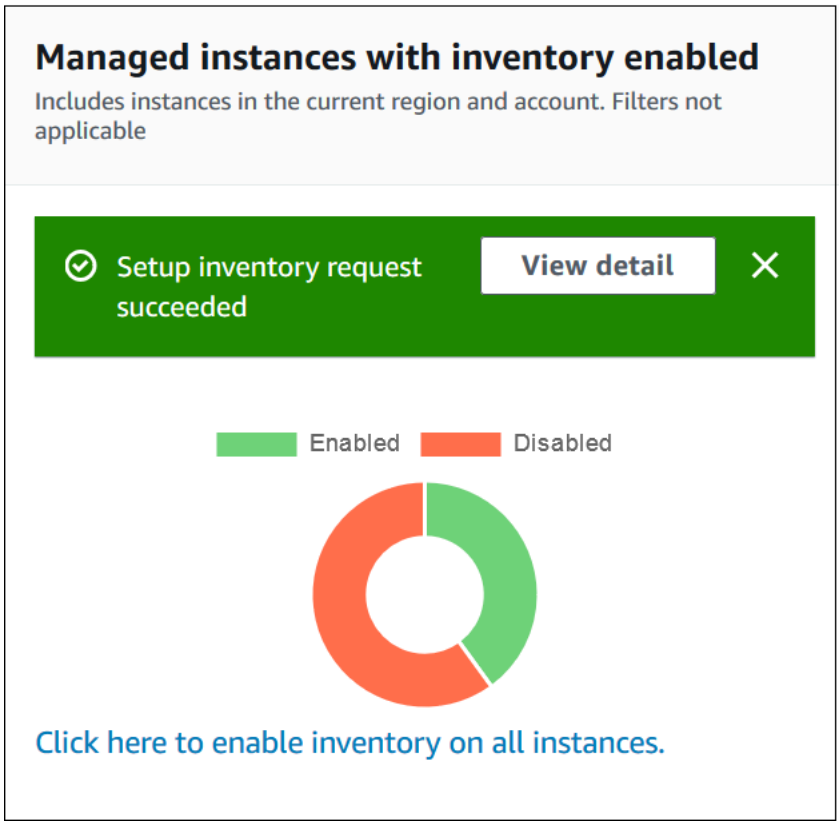
Gehen Sie wie folgt vor, um Systems Manager Inventory für alle verwalteten Knoten in Ihrem AWS-Konto und in einem einzigen zu konfigurieren AWS-Region.

So konfigurieren Sie all Ihre verwalteten Knoten in der aktuellen Region für Systems Manager Inventory

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Inventory.
3. Wählen Sie auf der Karte Managed instances with inventory enabled (Verwaltete Instances mit aktiviertem Bestand) die Option Click here to enable inventory on all instances (Klicken Sie hier, um den Bestand für alle Instances zu aktivieren) aus.



Wurde dieser Vorgang erfolgreich abgeschlossen, zeigt die Konsole die folgende Meldung an.



Je nach Anzahl der verwalteten Knoten in Ihrem Konto kann es einige Minuten dauern, bis die globale Bestandszuordnung angewendet wird. Warten Sie ein paar Minuten und aktualisieren Sie dann die Seite. Überprüfen Sie, ob die Konfiguration des Bestands auf all Ihren verwalteten Knoten in der Grafik entsprechend angezeigt wird.

Konfigurieren der Erfassung über die Konsole

Dieser Abschnitt enthält Informationen zum Konfigurieren von Systems Manager Inventory für das Sammeln von Metadaten aus Ihren verwalteten Knoten mithilfe der Systems-Manager-Konsole. Sie können schnell Metadaten von allen Knoten in einem bestimmten AWS-Konto (und allen future Knoten, die möglicherweise in diesem Konto erstellt werden) sammeln oder Sie können Inventardaten selektiv mithilfe von Tags oder Knoten IDs sammeln.

> **Note**
>
> Bevor Sie dieses Verfahren ausführen, prüfen Sie, ob bereits eine globale Bestandszuordnung existiert. Wenn bereits eine globale Bestandszuordnung vorhanden ist,

wird diese bei jedem Start einer neuen Instance auf diese angewendet und die neue Instance wird inventarisiert.

So konfigurieren Sie die Bestandserfassung

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Inventory.
3. Wählen Sie die Option Setup Inventory.
4. Identifizieren Sie im Abschnitt Targets (Ziele) die Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie eine der folgenden Optionen auswählen.

   - Auswählen aller verwalteten Instances in diesem Konto – Diese Option wählt alle verwalteten Knoten aus, für die es keine Bestandszuordnung gibt. Wenn Sie diese Option auswählen, werden Knoten, für die bereits Bestandszuordnungen durchgeführt wurden, während der Bestandserfassung übersprungen und mit dem Status Skipped (Übersprungen) in den Bestandsergebnissen angezeigt. Weitere Informationen finden Sie unter Inventarisieren Sie alle verwalteten Knoten in Ihrem AWS-Konto.
   - Specifying a tag (Angeben eines Tags) – Mit dieser Option können Sie ein einzelnes Tag zum Identifizieren von Knoten in Ihrem Konto angeben, aus denen Sie den Bestand erfassen möchten. Wenn Sie ein Tag verwenden, wird jeder in der Zukunft mit demselben Tag erstellte Knoten den Bestand ebenfalls melden. Wenn bereits eine Bestandszuordnung für alle Knoten besteht, überschreibt die Verwendung eines Tags zum Auswählen bestimmter Knoten als Ziel für einen anderen Bestand die Knoten-Mitgliedschaft in der Zielgruppe Alle verwalteten Instances. Verwaltete Knoten mit dem angegebenen Tag werden bei der künftigen Bestandserfassung von Allen verwalteten Instances übersprungen.
   - Manually selecting instances (Manuelles Auswählen von Instances) – Mit dieser Option können Sie bestimmte verwaltete Knoten in Ihrem Konto auswählen. Die explizite Auswahl bestimmter Knoten überschreibt bei Verwendung dieser Option die Bestandszuordnungen im Ziel Alle verwalteten Instances. Der Knoten wird bei der künftigen Bestandserfassung von Alle verwalteten Instances übersprungen.

> **Note**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

5. Wählen Sie im Bereich Schedule (Planung) aus, wie oft das System die Bestand-Metadaten in Ihren Knoten erfassen soll.
6. Verwenden Sie die Listen im Bereich Parameters, um die verschiedenen Typen der Bestandserfassung zu aktivieren oder zu deaktivieren. Weitere Informationen zur Erfassung von Datei- und Windows-Registry-Bestand finden Sie unter Arbeiten mit Datei- und Windows-Registrierungsbestand.
7. Wählen Sie im Bereich Advanced (Erweitert) die Option Sync inventory execution logs to an Amazon S3 bucket (Bestandsausführungsprotokolle mit einem Amazon S3-Bucket synchronisieren), wenn Sie den Ausführungsstatus der Zuordnung in einem S3-Bucket speichern möchten.
8. Wählen Sie die Option Setup Inventory. Systems Manager erstellt eine State Manager-Zuordnung und führt sofort Inventory auf den Knoten aus.
9. Wählen Sie im Navigationsbereich State Manager aus. Überprüfen Sie, ob eine neue Zuordnung erstellt wurde, die das **`AWS-GatherSoftwareInventory`**-Dokument verwendet. Der Zuordnungszeitplan verwendet einen Rate-Ausdruck. Überprüfen Sie auch, ob das Feld Status den Wert Success enthält. Wenn Sie die Option Sync inventory execution logs to an S3 bucket (Inventory-Ausführungsprotokolle mit einem S3-Bucket synchronisieren) ausgewählt haben, können Sie die Protokolldaten nach einigen Minuten in Amazon S3 anzeigen. Wenn Sie Bestandsdaten für einen bestimmten Knoten anzeigen möchten, klicken Sie auf Managed Instances (Verwaltete Instances) im Navigationsbereich.
10. Wählen Sie einen Knoten und anschließend View details (Details anzeigen).
11. Wählen Sie auf der Knoten-Detailseite Inventory aus. Verwenden Sie die Inventory type-Listen, um den Bestand zu filtern.

Abfragen von Bestandsdaten aus mehreren Regionen und Konten

AWS Systems Manager Inventory ist in Amazon Athena integriert, sodass Sie Inventardaten von mehreren AWS-Regionen und AWS-Konten abfragen können. Die Athena-Integration verwendet die

Ressourcendatensynchronisierung, sodass Sie Inventardaten von all Ihren verwalteten Knoten auf der Detailansichtsseite in der AWS Systems Manager Konsole anzeigen können.

> **Important**
>
> Diese Funktion wird verwendet AWS Glue , um die Daten in Ihrem Amazon Simple Storage Service (Amazon S3) -Bucket zu crawlen, und Amazon Athena, um die Daten abzufragen. Abhängig davon, wie viele Daten durchsucht und abgefragt werden, werden Ihnen diese Services in Rechnung gestellt. Mit AWS Glue zahlen Sie einen Stundensatz, der sekundengenau abgerechnet wird, für Crawler (Erkennung von Daten) und ETL-Jobs (Verarbeitung und Laden von Daten). Bei Athena richtet sich die Gebühr nach der Menge der pro Abfrage durchsuchten Daten. Wir empfehlen Ihnen, die Preisrichtlinien für diese Services zu lesen, bevor Sie die Amazon Athena-Integration mit Systems Manager Inventory verwenden. Weitere Informationen finden Sie unter Amazon Athena – Preise und AWS Glue - Preise.

Sie können Inventory-Daten auf der Seite Detailed View (Detailsansicht in allen AWS-Regionen anzeigen, in denen Amazon Athena verfügbar ist. Eine Liste der unterstützten Regionen finden Sie unter Amazon Athena-Service-Endpunkte im Allgemeine Amazon Web Services-Referenz.

Bevor Sie beginnen

Die Athena-Integration verwendet Resource Data Sync. Sie müssen Resource Data Sync einrichten und konfigurieren, um dieses Feature zu verwenden. Weitere Informationen finden Sie unter Walkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten.

Beachten Sie außerdem, dass die Detailed View (Detailansicht Bestandsdaten für den Besitzer des zentralen Amazon S3-Buckets anzeigt, der von Resource Data Sync verwendet wird. Wenn Sie nicht der Besitzer des zentralen Amazon S3-Buckets sind, werden Ihnen auf der Seite Detailed View (Detailansicht) keine Bestandsdaten angezeigt.

Zugriff konfigurieren

Bevor Sie Daten aus mehreren Konten und Regionen auf der Seite Detailsansicht in der Systems-Manager-Konsole abfragen und anzeigen können, müssen Sie Ihre (IAM)-Entität mit Berechtigungen zur Ansicht der Daten konfigurieren.

Wenn die Inventardaten in einem Amazon S3 S3-Bucket gespeichert sind, der die Verschlüsselung AWS Key Management Service (AWS KMS) verwendet, müssen Sie auch Ihre IAM-Entität und die `Amazon-GlueServiceRoleForSSM` Servicerolle für die AWS KMS Verschlüsselung konfigurieren.

Themen

- Konfigurieren Ihrer IAM-Entität für den Zugriff auf die Seite Detailansicht
- (Optional) Konfigurieren Sie Berechtigungen für die Anzeige AWS KMS verschlüsselter Daten

Konfigurieren Ihrer IAM-Entität für den Zugriff auf die Seite Detailansicht

Im Folgenden werden die Mindestberechtigungen beschrieben, die zum Anzeigen von Bestandsdaten auf der Seite Detailansicht erforderlich sind.

Die von **`AWSQuicksightAthenaAccess`** verwaltete Richtlinie

Die folgende `PassRole` und der zusätzliche erforderliche Berechtigungsblock

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowGlue",
            "Effect": "Allow",
            "Action": [
                "glue:GetCrawler",
                "glue:GetCrawlers",
                "glue:GetTables",
                "glue:StartCrawler",
                "glue:CreateCrawler"
            ],
            "Resource": "*"
        },
        {
            "Sid": "iamPassRole",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::111122223333:role/SSMInventoryGlueRole",
                "arn:aws:iam::111122223333:role/SSMInventoryServiceRole"
```

```
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "glue.amazonaws.com"
                }
            }
        },
        {
            "Sid": "iamRoleCreation",
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:AttachRolePolicy"
            ],
            "Resource": "arn:aws:iam::111122223333:role/*"
        },
        {
            "Sid": "iamPolicyCreation",
            "Effect": "Allow",
            "Action": "iam:CreatePolicy",
            "Resource": "arn:aws:iam::111122223333:policy/*"
        }
    ]
}
```

(Optional) Wenn der Amazon S3 S3-Bucket, der zum Speichern von Inventardaten verwendet wird AWS KMS, mithilfe von verschlüsselt ist, müssen Sie der Richtlinie auch den folgenden Block hinzufügen.

```
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:Region:account_ID:key/key_ARN"
    ]
}
```

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center -Benutzerhandbuch.

- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

  Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anleitung unter Eine Rolle für einen externen Identitätsanbieter (Verbund) erstellen im IAM-Benutzerhandbuch.

- IAM-Benutzer:
  - Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Befolgen Sie die Anleitung unter Eine Rolle für einen IAM-Benutzer erstellen im IAM-Benutzerhandbuch.
  - (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter Hinzufügen von Berechtigungen zu einem Benutzer (Konsole) im IAM-Benutzerhandbuch.

(Optional) Konfigurieren Sie Berechtigungen für die Anzeige AWS KMS verschlüsselter Daten

Wenn der Amazon S3 S3-Bucket, der zum Speichern von Inventardaten verwendet wird, mithilfe von AWS Key Management Service (AWS KMS) verschlüsselt ist, müssen Sie Ihre IAM-Entität und die GlueServiceRoleForAmazon-SSM-Rolle mit `kms:Decrypt` Berechtigungen für den AWS KMS Schlüssel konfigurieren.

Bevor Sie beginnen

Um die `kms:Decrypt` Berechtigungen für den AWS KMS Schlüssel bereitzustellen, fügen Sie Ihrer IAM-Entität den folgenden Richtlinienblock hinzu:

```
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:Region:account_ID:key/key_ARN"
    ]
}
```

Falls Sie dies noch nicht getan haben, schließen Sie dieses Verfahren ab und fügen Sie `kms:Decrypt` Berechtigungen für den AWS KMS Schlüssel hinzu.

Gehen Sie wie folgt vor, um die GlueServiceRoleForAmazon-SSM-Rolle mit den `kms:Decrypt` Berechtigungen für den AWS KMS Schlüssel zu konfigurieren.

So konfigurieren Sie die GlueServiceRoleForAmazon-SSM-Rolle mit Berechtigungen **`kms:Decrypt`**

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen aus, und verwenden Sie dann das Suchfeld, um die GlueServiceRoleForAmazon-SSM-Rolle zu suchen. Die Seite Summary (Übersicht) wird geöffnet.
3. Verwenden Sie das Suchfeld, um die GlueServiceRoleForAmazon-SSM-Rolle zu finden. Wählen Sie den Rollennamen aus. Die Seite Summary (Übersicht) wird geöffnet.
4. Wählen Sie den Rollennamen aus. Die Seite Summary (Übersicht) wird geöffnet.
5. Wählen Sie Inline-Richtlinie hinzufügen. Die Seite Create policy (Richtlinie erstellen) wird geöffnet.
6. Wählen Sie den Tab JSON.
7. Löschen Sie den vorhandenen JSON-Text im Editor, kopieren Sie die folgende Richtlinie und fügen Sie sie in den JSON-Editor ein.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/key_ARN"
            ]
        }
    ]
}
```

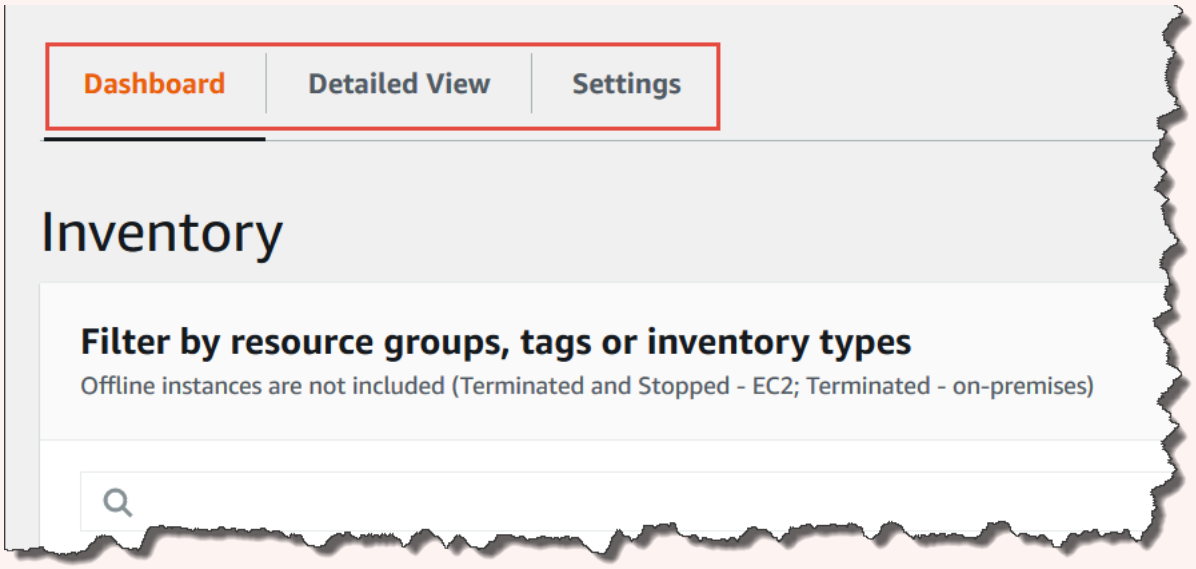
8. Wählen Sie Review policy (Richtlinie überprüfen) aus.
9. Geben Sie auf der Seite Review Policy (Richtlinie überprüfen) im Feld Name einen Namen ein.
10. Wählen Sie Richtlinie erstellen aus.

Abfragen von Daten auf der Seite „Inventory Detailed View (Detaillierte Bestandsansicht)“

Gehen Sie wie folgt vor, um Inventardaten von mehreren AWS-Regionen und AWS-Konten auf der Seite Inventar Detailed View von Systems Manager anzuzeigen.

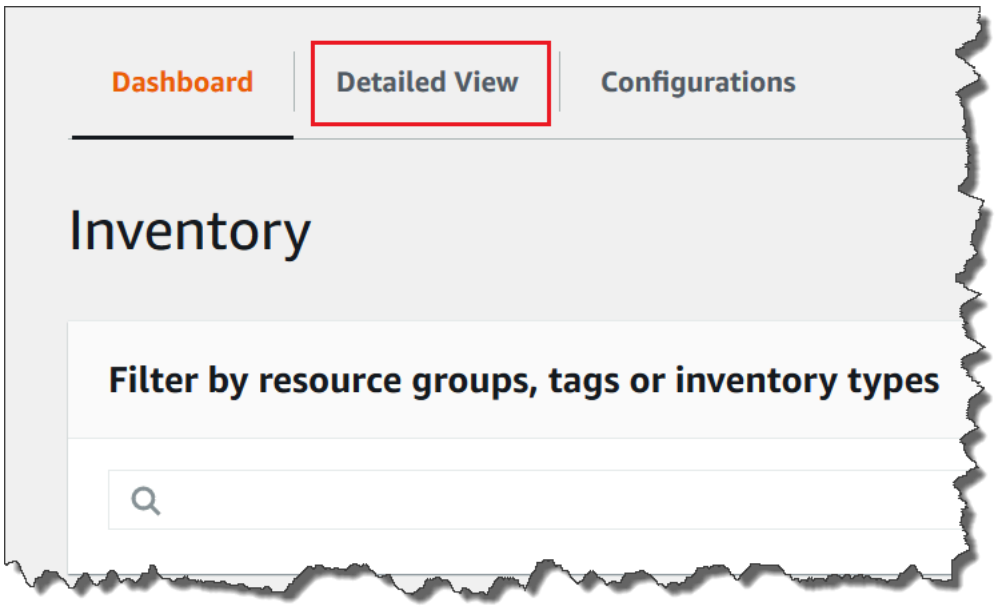
Important

Die Seite Inventory Detailed View (Detailansicht) ist nur in AWS-Regionen verfügbar, die Amazon Athena anbieten. Wenn die folgenden Registerkarten nicht auf der Seite Systems Manager Inventory angezeigt werden, bedeutet dies, dass Athena nicht in der Region verfügbar ist und Sie die Detailansicht nicht verwenden können, um Daten abzufragen.

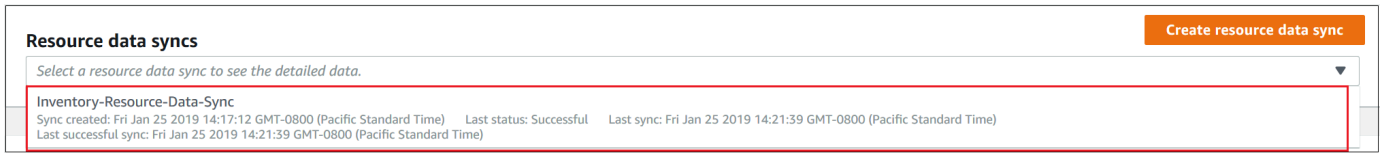


Bestandsdaten aus mehreren Regionen und Konten in der AWS Systems Manager -Konsole anzeigen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Inventory.
3. Wählen Sie die Registerkarte Detailed View (Detaillierte Ansicht) aus.

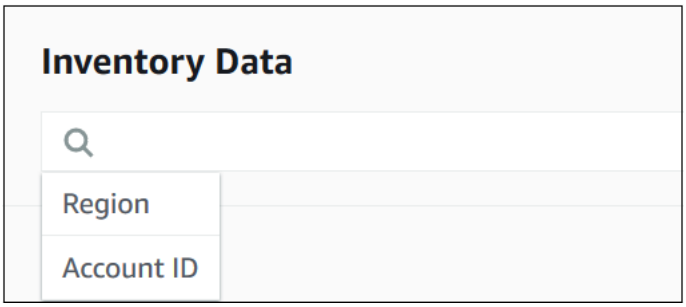


4. Wählen Sie die Resource Data Sync aus, für die Sie Daten abfragen möchten.



5. Wählen Sie in der Liste Inventory Type (Bestandstyp) den Typ der Bestandsdaten aus, die Sie abfragen möchten, und drücken Sie dann Enter.

6. Um die Daten zu filtern, wählen Sie die Filterleiste aus und wählen Sie dann eine Filteroption aus.



Sie können die Schaltfläche Export to CSV (Exportieren in CSV) verwenden, um die aktuelle Abfrage in einem Tabellenkalkulationsprogramm wie Microsoft Excel anzuzeigen. Sie können auch die Schaltflächen Query History (Abfrageverlauf) und Run Advanced Queries (Erweiterte Abfragen ausführen) verwenden, um mit Ihren Daten in Amazon Athena zu interagieren.

Bearbeiten des Zeitplans für den AWS Glue -Crawler

AWS Glue crawlt standardmäßig zweimal täglich die Inventardaten im zentralen Amazon S3 S3-Bucket. Wenn Sie häufig die Arten der auf Ihren Knoten zu erfassenden Daten ändern, möchten Sie möglicherweise Sie die Daten häufiger durchsuchen, wie im folgenden Verfahren beschrieben.

> **Important**
>
> AWS Glue AWS-Konto berechnet Ihnen für Crawler (Erkennung von Daten) und ETL-Jobs (Verarbeitung und Laden von Daten) einen Stundensatz, der sekundenweise abgerechnet wird. Bevor Sie den Crawler-Zeitplan anzeigen, rufen Sie die AWS Glue -Preisliste auf.

So ändern Sie den Bestandsdatencrawler-Zeitplan

1. Öffnen Sie die AWS Glue Konsole unter. https://console.aws.amazon.com/glue/
2. Wählen Sie im Navigationsbereich Crawlers (Crawler) aus.
3. Wählen Sie in der Liste der Crawler die Option neben dem Systems Manager Inventory-Crawler aus. Der Crawler-Name verwendet das folgende Format:

   `AWSSystemsManager-`*`s3-bucket-name`*`-`*`Region`*`-`*`account_ID`*
4. Wählen Sie Action (Aktion) und Edit crawler (Crawler bearbeiten) aus.
5. Wählen Sie im Navigationsbereich Schedule (Zeitplan) aus.
6. Geben Sie im Feld Cron expression (cron-Ausdruck) einen neuen Zeitplan mit einem Cron-Format an. Weitere Informationen zum Cron-Format finden Sie unter Zeitpläne für Aufträge und Crawler im AWS Glue Developer Guide.

> **Important**
>
> Sie können den Crawler anhalten, damit keine Gebühren mehr von anfallen. AWS Glue Wenn Sie den Crawler aussetzen oder die Häufigkeit ändern, damit die Daten weniger häufig

durchsucht werden, zeigt Inventory Detailed View (Detaillierte Ansicht) möglicherweise Daten an, die nicht aktuell sind.

Abfragen der Bestandserfassung mithilfe von Filtern

Nachdem Sie Bestandsdaten erfasst haben, können Sie den Filter-Funktionen in AWS Systems Manager nutzen, um eine Liste der verwalteten Knoten abzufragen, die bestimmte Filterkriterien erfüllen.

So fragen Sie Knoten basierend auf Bestandsfiltern ab

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Inventory.
3. Wählen Sie im Abschnitt Filter by resource groups, tags or inventory types die Filteroption. Eine Liste vordefinierter Filter wird angezeigt.
4. Wählen Sie ein Attribut, nach dem gefiltert werden soll. Wählen Sie zum Beispiel **`AWS:Application`** aus. Wenn Sie dazu aufgefordert werden, wählen Sie ein sekundäres Attribut, nach dem gefiltert werden soll. Wählen Sie zum Beispiel **`AWS:Application.Name`** aus.
5. Wählen Sie eine Begrenzung in der Liste aus. Wählen Sie z. B. Begin with. Im Filter wird ein Textfeld angezeigt.
6. Geben Sie einen Wert in das Textfeld ein. Geben Sie beispielsweise Amazon ein (SSM Agent ist als Amazon SSM Agent benannt).
7. Drücken Sie Enter. Das System gibt eine Liste der verwalteten Knoten zurück, die einen Anwendungsnamen haben, der mit dem Wort Amazon beginnt.

> **Note**
>
> Sie können mehrere Filter kombinieren, um die Suche zu verfeinern.

Aggregieren von Bestandsdaten

Nachdem Sie Ihre verwalteten Knoten für AWS Systems Manager Inventar konfiguriert haben, können Sie die aggregierte Anzahl der Inventardaten anzeigen. Nehmen wir beispielsweise an, Sie haben Dutzende oder Hunderte von verwalteten Knoten zur Erfassung des Bestandstyps `AWS:Application` konfiguriert. Mithilfe der Informationen in diesem Abschnitt können Sie die genaue Zahl der Knoten anzeigen, die zum Erfassen dieser Daten konfiguriert sind.

Sie können außerdem spezifische Bestandsdetails durch Aggregieren eines Datentyps sehen. Beispiel: Der Bestandstyp `AWS:InstanceInformation` erfasst Betriebssystem-Plattforminformationen mit dem Datentyp `Platform`. Durch die Aggregierung von Daten für den Datentyp `Platform` können Sie schnell sehen, auf wie vielen Knoten Windows Server, Linux und macOS ausgeführt wird.

Die Verfahren in diesem Abschnitt beschreiben, wie Sie die aggregierte Anzahl von Inventardaten mithilfe von AWS Command Line Interface ()AWS CLI anzeigen können. Sie können vorkonfigurierte aggregierte Zählungen auch in der AWS Systems Manager Konsole auf der Inventarseite anzeigen. Diese vorkonfigurierten Dashboards werden als Inventory Insights (Bestandseinblicke) bezeichnet. Sie bieten eine 1-Klick-Wiederherstellung Ihrer Bestand-Konfigurationsprobleme.

Beachten Sie die folgenden wichtigen Details zu den Aggregationszählungen von Bestandsdaten:

- Wenn Sie einen verwalteten Knoten beenden, der zum Sammeln von Bestandsdaten konfiguriert ist, behält Systems Manager die Bestandsdaten 30 Tage lang bei und löscht sie anschließend. Für ausgeführte Knoten löscht das System alle Bestandsdaten, die älter als 30 Tage sind. Wenn Sie Inventardaten länger als 30 Tage speichern müssen, können AWS Config Sie den Verlauf aufzeichnen oder die Daten regelmäßig abfragen und in einen Amazon Simple Storage Service (Amazon S3) -Bucket hochladen.
- Wenn ein Knoten zuvor konfiguriert wurde, um einen spezifischen Bestandsdatentyp zu melden, z. B. `AWS:Network`, und Sie die Konfiguration später so ändern, dass die Daten dieses Typs nicht mehr erfasst werden, zeigt die Aggregationszählung nach wie vor `AWS:Network`-Daten an, bis der Knoten beendet wurde und 30 Tage vergangen sind.

Informationen zur schnellen Konfiguration und Erfassung von Inventardaten von allen Knoten in einem bestimmten AWS-Konto (und allen future Knoten, die möglicherweise in diesem Konto erstellt werden) finden Sie unter Inventarisieren Sie alle verwalteten Knoten in Ihrem AWS-Konto.

Themen

- Aggregieren von Bestandsdaten zum Anzeigen der Anzahl von Knoten, die bestimmte Arten von Daten erfassen
- Aggregieren von Bestandsdaten mit Gruppen, um zu sehen, welche Knoten zur Erfassung eines Bestandstyps konfiguriert sind und welche nicht

Aggregieren von Bestandsdaten zum Anzeigen der Anzahl von Knoten, die bestimmte Arten von Daten erfassen

Sie können den AWS Systems Manager GetInventoryAPI-Vorgang verwenden, um die aggregierte Anzahl von Knoten anzuzeigen, die einen oder mehrere Inventar- und Datentypen erfassen. Mit dem `AWS:InstanceInformation` Inventartyp können Sie beispielsweise eine Zusammenfassung von Betriebssystemen anzeigen, indem Sie den GetInventory API-Vorgang mit dem `AWS:InstanceInformation.PlatformType` Datentyp verwenden. Hier finden Sie ein Beispiel für den AWS CLI -Befehl und die Ausgabe.

```
aws ssm get-inventory --aggregators "Expression=AWS:InstanceInformation.PlatformType"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "Entities":[
      {
         "Data":{
            "AWS:InstanceInformation":{
               "Content":[
                  {
                     "Count":"7",
                     "PlatformType":"windows"
                  },
                  {
                     "Count":"5",
                     "PlatformType":"linux"
                  }
               ]
            }
         }
      }
   ]
}
```

Erste Schritte

Bestimmen Sie die Bestands- und Datentypen, für die Sie Zählungen anzeigen möchten. Sie können eine Liste der Bestandstypen und Datentypen anzeigen, die die Aggregation unterstützen, indem Sie den folgenden Befehl im Fenster AWS CLI ausführen.

```
aws ssm get-inventory-schema --aggregator
```

Der Befehl gibt eine JSON-Liste mit Bestands- und Datentypen zurück, die die Aggregation unterstützen. Das TypeNameFeld zeigt die unterstützten Inventartypen. Das Feld Name zeigt die einzelnen Datentypen. Der Bestandstyp `AWS:Application` in der folgenden Listen enthält z. B. Datentypen für `Name` und `Version`.

```
{
    "Schemas": [
        {
            "TypeName": "AWS:Application",
            "Version": "1.1",
            "DisplayName": "Application",
            "Attributes": [
                {
                    "DataType": "STRING",
                    "Name": "Name"
                },
                {
                    "DataType": "STRING",
                    "Name": "Version"
                }
            ]
        },
        {
            "TypeName": "AWS:InstanceInformation",
            "Version": "1.0",
            "DisplayName": "Platform",
            "Attributes": [
                {
                    "DataType": "STRING",
                    "Name": "PlatformName"
                },
                {
                    "DataType": "STRING",
                    "Name": "PlatformType"
```

```
                },
                {
                    "DataType": "STRING",
                    "Name": "PlatformVersion"
                }
            ]
        },
        {
            "TypeName": "AWS:ResourceGroup",
            "Version": "1.0",
            "DisplayName": "ResourceGroup",
            "Attributes": [
                {
                    "DataType": "STRING",
                    "Name": "Name"
                }
            ]
        },
        {
            "TypeName": "AWS:Service",
            "Version": "1.0",
            "DisplayName": "Service",
            "Attributes": [
                {
                    "DataType": "STRING",
                    "Name": "Name"
                },
                {
                    "DataType": "STRING",
                    "Name": "DisplayName"
                },
                {
                    "DataType": "STRING",
                    "Name": "ServiceType"
                },
                {
                    "DataType": "STRING",
                    "Name": "Status"
                },
                {
                    "DataType": "STRING",
                    "Name": "StartType"
                }
            ]
```

```
        },
        {
            "TypeName": "AWS:WindowsRole",
            "Version": "1.0",
            "DisplayName": "WindowsRole",
            "Attributes": [
                {
                    "DataType": "STRING",
                    "Name": "Name"
                },
                {
                    "DataType": "STRING",
                    "Name": "DisplayName"
                },
                {
                    "DataType": "STRING",
                    "Name": "FeatureType"
                },
                {
                    "DataType": "STRING",
                    "Name": "Installed"
                }
            ]
        }
    ]
}
```

Sie können Daten für einen der gelisteten Bestandstypen aggregieren, indem Sie einen Befehl erstellen, der die folgende Syntax verwendet.

```
aws ssm get-inventory --aggregators "Expression=InventoryType.DataType"
```

Hier sind einige Beispiele.

Beispiel 1

Dieses Beispiel aggregiert eine Zählung der Windows-Rollen, die von Ihren Knoten verwendet werden.

```
aws ssm get-inventory --aggregators "Expression=AWS:WindowsRole.Name"
```

Beispiel 2

Dieses Beispiel aggregiert eine Zählung der Anwendungen, die auf Ihren Knoten installiert sind.

```
aws ssm get-inventory --aggregators "Expression=AWS:Application.Name"
```

Kombinieren von mehreren Aggregatoren

Sie können auch mehrere Bestands- und Datentypen in einem Befehl kombinieren, um die Daten besser zu verstehen. Hier sind einige Beispiele.

Beispiel 1

Dieses Beispiel aggregiert eine Zählung der Betriebssystemtypen, die von Ihren Knoten verwendet werden. Außerdem wird der spezifische Name der Betriebssysteme zurückgegeben.

```
aws ssm get-inventory --aggregators '[{"Expression":
 "AWS:InstanceInformation.PlatformType", "Aggregators":[{"Expression":
 "AWS:InstanceInformation.PlatformName"}]}]'
```

Beispiel 2

Dieses Beispiel aggregiert eine Zählung der Anwendungen, die auf Ihren Knoten ausgeführt werden, sowie die spezifische Version der einzelnen Anwendungen.

```
aws ssm get-inventory --aggregators '[{"Expression": "AWS:Application.Name",
 "Aggregators":[{"Expression": "AWS:Application.Version"}]}]'
```

Wenn Sie möchten, können Sie einen Aggregationsausdruck mit einem oder mehreren Bestands- und Datentypen in einer JSON-Datei erstellen und die Datei über die AWS CLI aufrufen. Die JSON-Datei muss die folgende Syntax verwenden.

```
[
       {
           "Expression": "string",
           "Aggregators": [
               {
                  "Expression": "string"
               }
           ]
       }
```

```
]
```

Sie müssen die Datei mit der Erweiterung „.json” speichern.

Im folgenden Beispiel werden mehrere Bestands- und Datentypen verwendet.

```
[
       {
           "Expression": "AWS:Application.Name",
           "Aggregators": [
               {
                   "Expression": "AWS:Application.Version",
                   "Aggregators": [
                     {
                     "Expression": "AWS:InstanceInformation.PlatformType"
                     }
                   ]
               }
           ]
       }
]
```

Rufen Sie die Datei mit folgendem Befehl über die AWS CLI auf.

```
aws ssm get-inventory --aggregators file://file_name.json
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{"Entities":
 [
   {"Data":
     {"AWS:Application":
       {"Content":
         [
           {"Count": "3",
            "PlatformType": "linux",
            "Version": "2.6.5",
            "Name": "audit-libs"},
           {"Count": "2",
            "PlatformType": "windows",
            "Version": "2.6.5",
            "Name": "audit-libs"},
           {"Count": "4",
```

```
            "PlatformType": "windows",
            "Version": "6.2.8",
            "Name": "microsoft office"},
           {"Count": "2",
            "PlatformType": "windows",
            "Version": "2.6.5",
            "Name": "chrome"},
           {"Count": "1",
            "PlatformType": "linux",
            "Version": "2.6.5",
            "Name": "chrome"},
           {"Count": "2",
            "PlatformType": "linux",
            "Version": "6.3",
            "Name": "authconfig"}
         ]
       }
     },
    "ResourceType": "ManagedInstance"}
 ]
}
```

Aggregieren von Bestandsdaten mit Gruppen, um zu sehen, welche Knoten zur Erfassung eines Bestandstyps konfiguriert sind und welche nicht

Gruppen in Systems Manager Inventory ermöglichen es Ihnen, schnell eine Anzahl der verwalteten Knoten zu sehen, die für das Erfassen einer oder mehrerer Bestandstypen konfiguriert bzw. nicht konfiguriert sind. Mit Gruppen geben Sie einen oder mehrere Bestandstypen sowie einen Filter an, der den `exists`-Operator verwendet.

Beispiel: Angenommen, Sie haben vier verwaltete Knoten zum Erfassen der folgenden Bestandstypen konfiguriert:

- Knoten 1: `AWS:Application`
- Knoten 2: `AWS:File`
- Knoten 3: `AWS:Application`, `AWS:File`
- Knoten 4: `AWS:Network`

Sie können den folgenden Befehl vom ausführen, AWS CLI um zu sehen, wie viele Knoten so konfiguriert sind, dass sie `AWS:Application` sowohl die `AWS:File inventory` Typen als auch

erfassen. Die Antwort gibt auch die Anzahl der Knoten zurück, die nicht zum Erfassen dieser beiden Bestandstypen konfiguriert sind.

```
aws ssm get-inventory --aggregators
 'Groups=[{Name=ApplicationAndFile,Filters=[{Key=TypeName,Values=[AWS:Application],Type=Exists}
{Key=TypeName,Values=[AWS:File],Type=Exists}]}]'
```

Die Befehlsantwort zeigt, dass nur ein verwalteter Knoten zum Erfassen der beiden Bestandstypen `AWS:Application` und `AWS:File` konfiguriert ist.

```
{
   "Entities":[
      {
         "Data":{
            "ApplicationAndFile":{
               "Content":[
                  {
                     "notMatchingCount":"3"
                  },
                  {
                     "matchingCount":"1"
                  }
               ]
            }
         }
      }
   ]
}
```

> **Note**
>
> Gruppen geben keine Datentypzahlen zurück. Außerdem können Sie die Ergebnisse nicht detailliert aufschlüsseln, um zu sehen, welche Knoten für die Erfassung IDs des Inventartyps konfiguriert sind oder nicht.

Wenn Sie möchten, können Sie einen Aggregationsausdruck mit einem oder mehreren Bestandstypen in einer JSON-Datei erstellen und die Datei über die AWS CLI aufrufen. Die JSON-Datei muss die folgende Syntax verwenden:

```
{
```

```
   "Aggregators":[
      {
         "Groups":[
            {
               "Name":"Name",
               "Filters":[
                  {
                     "Key":"TypeName",
                     "Values":[
                        "Inventory_type"
                     ],
                     "Type":"Exists"
                  },
                  {
                     "Key":"TypeName",
                     "Values":[
                        "Inventory_type"
                     ],
                     "Type":"Exists"
                  }
               ]
            }
         ]
      }
   ]
}
```

Sie müssen die Datei mit der Erweiterung „.json” speichern.

Rufen Sie die Datei mit folgendem Befehl über die AWS CLI auf.

```
aws ssm get-inventory --cli-input-json file://file_name.json
```

Weitere Beispiele

Die folgenden Beispiele zeigen Ihnen, wie Sie Bestandsdaten aggregieren, um zu sehen, welche verwalteten Knoten zum Erfassen der angegebenen Bestandstypen konfiguriert bzw. nicht konfiguriert sind. Diese Beispiele verwenden die AWS CLI. Jedes Beispiel enthält einen vollständigen Befehl mit Filtern, die Sie über die Befehlszeile ausführen können, und eine input.json-Beispieldatei, falls Sie es vorziehen, die Informationen in eine Datei einzugeben.

Beispiel 1

Dieses Beispiel aggregiert eine Zählung der Knoten, die zum Erfassen entweder des Bestandstyps `AWS:Application` oder des Typs `AWS:File` konfiguriert bzw. nicht konfiguriert sind.

Führen Sie den folgenden Befehl über die AWS CLI aus.

```
aws ssm get-inventory --aggregators
 'Groups=[{Name=ApplicationORFile,Filters=[{Key=TypeName,Values=[AWS:Application,
 AWS:File],Type=Exists}]}]'
```

Wenn Sie eine Datei verwenden möchten, kopieren Sie das folgende Beispiel in eine Datei und speichern Sie sie unter dem Namen input.json.

```
{
   "Aggregators":[
      {
         "Groups":[
            {
               "Name":"ApplicationORFile",
               "Filters":[
                  {
                     "Key":"TypeName",
                     "Values":[
                        "AWS:Application",
                        "AWS:File"
                     ],
                     "Type":"Exists"
                  }
               ]
            }
         ]
      }
   ]
}
```

Führen Sie den folgenden Befehl über die AWS CLI aus.

```
aws ssm get-inventory --cli-input-json file://input.json
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
```

```
   "Entities":[
      {
         "Data":{
            "ApplicationORFile":{
               "Content":[
                  {
                     "notMatchingCount":"1"
                  },
                  {
                     "matchingCount":"3"
                  }
               ]
            }
         }
      }
   ]
}
```

Beispiel 2

Dieses Beispiel aggregiert eine Zählung der Knoten, die zum Erfassen des Bestandstyps `AWS:Application`, `AWS:File` und `AWS:Network` konfiguriert bzw. nicht konfiguriert sind.

Führen Sie den folgenden Befehl über die AWS CLI aus.

```
aws ssm get-inventory --aggregators
 'Groups=[{Name=Application,Filters=[{Key=TypeName,Values=[AWS:Application],Type=Exists}]},
 {Name=File,Filters=[{Key=TypeName,Values=[AWS:File],Type=Exists}]},
 {Name=Network,Filters=[{Key=TypeName,Values=[AWS:Network],Type=Exists}]}]'
```

Wenn Sie eine Datei verwenden möchten, kopieren Sie das folgende Beispiel in eine Datei und speichern Sie sie unter dem Namen input.json.

```
{
   "Aggregators":[
      {
         "Groups":[
            {
               "Name":"Application",
               "Filters":[
                  {
                     "Key":"TypeName",
                     "Values":[
```

```
                        "AWS:Application"
                     ],
                     "Type":"Exists"
                  }
               ]
            },
            {
               "Name":"File",
               "Filters":[
                  {
                     "Key":"TypeName",
                     "Values":[
                        "AWS:File"
                     ],
                     "Type":"Exists"
                  }
               ]
            },
            {
               "Name":"Network",
               "Filters":[
                  {
                     "Key":"TypeName",
                     "Values":[
                        "AWS:Network"
                     ],
                     "Type":"Exists"
                  }
               ]
            }
         ]
      }
   ]
}
```

Führen Sie den folgenden Befehl über die AWS CLI aus.

```
aws ssm get-inventory --cli-input-json file://input.json
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
   "Entities":[
```

```
      {
         "Data":{
            "Application":{
               "Content":[
                  {
                     "notMatchingCount":"2"
                  },
                  {
                     "matchingCount":"2"
                  }
               ]
            },
            "File":{
               "Content":[
                  {
                     "notMatchingCount":"2"
                  },
                  {
                     "matchingCount":"2"
                  }
               ]
            },
            "Network":{
               "Content":[
                  {
                     "notMatchingCount":"3"
                  },
                  {
                     "matchingCount":"1"
                  }
               ]
            }
         }
      }
   ]
}
```

Arbeiten mit benutzerdefiniertem Bestand

Sie können Ihren Knoten beliebige Metadaten zuweisen, indem Sie ein benutzerdefiniertes AWS Systems Manager Inventar erstellen. Nehmen wir z. B. an, dass Sie eine große Anzahl von Servern in Racks in Ihrem Rechenzentrum verwalten; und diese Server als von Systems Manager verwaltete Knoten konfiguriert wurden. Derzeit speichern Sie die Informationen zu den Standorten der Server-

Racks in einer Tabelle. Mit einem benutzerdefinierten Bestand können Sie die Rack-Standorte der einzelnen Knoten als Metadaten auf dem Knoten angeben. Wenn Sie den Bestand mit Systems Manager erfassen, werden die Metadaten zusammen mit den anderen Bestandsmetadaten erfasst. Anschließend können Sie alle Bestandsmetadaten in einen zentralen Amazon S3-Bucket portieren, indem Sie Resource Data Sync (Ressourcendaten-Synchronisation) verwenden und die Daten abfragen.

> **Note**
>
> Systems Manager unterstützt maximal 20 benutzerdefinierte Bestandstypen pro AWS-Konto.

Um einem Knoten ein benutzerdefiniertes Inventar zuzuweisen, können Sie entweder den Systems Manager PutInventoryAPI-Vorgang verwenden, wie unter beschriebenZuweisen benutzerdefinierter Bestands-Metadaten zu einem verwalteten Knoten. Oder Sie können eine JSON-Datei für den benutzerdefinierten Bestand erstellen und diese auf den Knoten hochladen. In diesem Abschnitt wird beschrieben, wie Sie die JSON-Datei erstellen.

Die folgende Beispiel-JSON-Datei mit benutzerdefiniertem Bestand gibt Rack-Informationen zu einem On-Premises-Server an. Dieses Beispiel gibt einen Typ von benutzerdefinierten Bestandsdaten (`"TypeName": "Custom:RackInformation"`) an, mit mehreren Einträgen unter `Content`, die die Daten beschreiben.

```
{
    "SchemaVersion": "1.0",
    "TypeName": "Custom:RackInformation",
    "Content": {
        "Location": "US-EAST-02.CMH.RACK1",
        "InstalledTime": "2016-01-01T01:01:01Z",
        "vendor": "DELL",
        "Zone" : "BJS12",
        "TimeZone": "UTC-8"
      }
 }
```

Sie können wie im folgenden Beispiel gezeigt auch verschiedene Einträge im Abschnitt `Content` angeben.

```
{
```

```
"SchemaVersion": "1.0",
"TypeName": "Custom:PuppetModuleInfo",
    "Content": [{
        "Name": "puppetlabs/aws",
        "Version": "1.0"
      },
      {
        "Name": "puppetlabs/dsc",
        "Version": "2.0"
      }
    ]
}
```

Das JSON-Schema für den benutzerdefinierten Bestand erfordert die Abschnitte `SchemaVersion`, `TypeName` und `Content`, aber Sie können die Informationen in diesen Abschnitten definieren.

```
{
    "SchemaVersion": "user_defined",
    "TypeName": "Custom:user_defined",
    "Content": {
        "user_defined_attribute1": "user_defined_value1",
        "user_defined_attribute2": "user_defined_value2",
        "user_defined_attribute3": "user_defined_value3",
        "user_defined_attribute4": "user_defined_value4"
      }
 }
```

Der `TypeName`-Wert ist auf 100 Zeichen begrenzt. Außerdem muss der `TypeName`-Wert mit dem großgeschriebenen Wort `Custom` beginnen. Beispiel, `Custom:PuppetModuleInfo`. Daher würden die folgenden Beispiele zu einer Ausnahme führen: `CUSTOM:PuppetModuleInfo`, `custom:PuppetModuleInfo`.

Der `Content` Abschnitt umfasst Attribute und*`data`*. Beachten Sie, dass bei diesen Elementen Groß- und Kleinschreibung nicht berücksichtigt wird. Wenn Sie jedoch ein Attribut definieren (z. B: `"Vendor": "DELL"`) müssen Sie dieses Attribut in Ihren Dateien für den benutzerdefinierten Bestand konsistent referenzieren. Wenn Sie in einer Datei `"Vendor": "DELL"` (mit einem großen „V” in `vendor`) und in einer anderen Datei `"vendor": "DELL"` (mit einem kleinen „v” in `vendor`) angeben, gibt das System ein Fehler zurück.

> **Note**
>
> Sie müssen die Datei mit der Erweiterung `.json` speichern und die von Ihnen definierte Bestandsliste darf nur aus Zeichenfolgenwerten bestehen.

Wenn Sie die Datei erstellt haben, müssen Sie sie auf dem Knoten speichern. In der folgenden Tabelle wird der jeweilige Speicherort angezeigt, an dem die JSON-Dateien für den benutzerdefinierten Bestand auf dem Knoten gespeichert werden müssen.

Betriebssystem	Pfad
Linux	/var/lib/amazon/ssm/*node-id*/inventory/benutzerdefiniert
macOS	`/opt/aws/ssm/data/` *node-id*`/inventory/custom`
Windows Server	%SystemDrive%\ProgramData\ Amazon\ SSM \\InstanceData\ Inventar*node-id*\ Benutzerdefiniert

Ein Beispiel für die Verwendung von benutzerdefiniertem Inventar finden Sie unter Ermitteln der Festplattenauslastung Ihrer Flotte mithilfe von benutzerdefinierten EC2 Systems Manager Manager-Inventartypen.

Löschen eines benutzerdefinierten Bestands

Sie können den DeleteInventoryAPI-Vorgang verwenden, um einen benutzerdefinierten Inventartyp und die mit diesem Typ verknüpften Daten zu löschen. Sie rufen den Befehl delete-inventory unter Verwendung der AWS Command Line Interface (AWS CLI) auf, um alle Daten für einen Bestandstyp zu löschen. Sie rufen den Befehl delete-inventory mit der `SchemaDeleteOption` auf, um einen benutzerdefinierten Bestandstyp zu löschen.

> **Note**
>
> Ein Bestandstyp wird auch als Bestandsschema bezeichnet.

Der Parameter `SchemaDeleteOption` umfasst die folgenden Optionen:

- DeleteSchema: Diese Option löscht den angegebenen benutzerdefinierten Typ und alle damit verknüpften Daten. Sie können das Schema später bei Bedarf erneut erstellen.
- DisableSchema: Wenn Sie diese Option wählen, schaltet das System die aktuelle Version aus, löscht alle zugehörigen Daten und ignoriert alle neuen Daten, wenn die Version kleiner oder gleich der ausgeschalteten Version ist. Sie können diesen Inventartyp wieder zulassen, indem Sie die PutInventoryAktion für eine Version aufrufen, die höher ist als die ausgeschaltete Version.

Um das benutzerdefinierte Inventar zu löschen oder zu deaktivieren, verwenden Sie AWS CLI

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus, um mit der Option `dry-run` anzuzeigen, welche Daten aus dem System gelöscht werden. Mit diesem Befehl werden keine Daten gelöscht.

   ```
   aws ssm delete-inventory --type-name "Custom:custom_type_name" --dry-run
   ```

   Das System gibt unter anderem folgende Informationen zurück

   ```
   {
      "DeletionSummary":{
         "RemainingCount":3,
         "SummaryItems":[
            {
               "Count":2,
               "RemainingCount":2,
               "Version":"1.0"
            },
            {
               "Count":1,
               "RemainingCount":1,
               "Version":"2.0"
            }
         ],
         "TotalCount":3
      },
   ```

```
   "TypeName":"Custom:custom_type_name"
}
```

Informationen zur Interpretation der Übersicht für gelöschten Bestand finden Sie unter Interpretieren der Übersicht für gelöschten Bestand.

3. Führen Sie den folgenden Befehl aus, um alle Daten für einen benutzerdefinierten Bestandstyp zu löschen.

```
aws ssm delete-inventory --type-name "Custom:custom_type_name"
```

> **Note**
>
> Der Fortschritt des Löschvorgangs wird in der Ausgabe dieses Befehls nicht angezeigt. Aus diesem Grund sind TotalCount und Verbleibende Anzahl immer gleich, da das System noch nichts gelöscht hat. Sie können den describe-inventory-deletions Befehl verwenden, um den Fortschritt des Löschvorgangs anzuzeigen, wie später in diesem Thema beschrieben.

Das System gibt unter anderem folgende Informationen zurück

```
{
   "DeletionId":"system_generated_deletion_ID",
   "DeletionSummary":{
      "RemainingCount":3,
      "SummaryItems":[
         {
            "Count":2,
            "RemainingCount":2,
            "Version":"1.0"
         },
         {
            "Count":1,
            "RemainingCount":1,
            "Version":"2.0"
         }
      ],
      "TotalCount":3
   },
   "TypeName":"custom_type_name"
```

```
}
```

Das System löscht alle Daten für den angegebenen benutzerdefinierten Bestandstyp aus dem Systems Manager Inventory-Service.

4. Führen Sie den folgenden Befehl aus. Der Befehl führt die folgenden Aktionen für die aktuelle Version des Bestandstyps aus: Deaktivieren der aktuellen Version, Löschen aller Daten daraus und Ignorieren aller neuen Daten, wenn die Version kleiner oder gleich der deaktivierten Version ist.

```
aws ssm delete-inventory --type-name "Custom:custom_type_name" --schema-delete-
option "DisableSchema"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "DeletionId":"system_generated_deletion_ID",
   "DeletionSummary":{
      "RemainingCount":3,
      "SummaryItems":[
         {
            "Count":2,
            "RemainingCount":2,
            "Version":"1.0"
         },
         {
            "Count":1,
            "RemainingCount":1,
            "Version":"2.0"
         }
      ],
      "TotalCount":3
   },
   "TypeName":"Custom:custom_type_name"
}
```

Sie können einen deaktivierten Bestandstyp mit dem folgenden Befehl anzeigen.

```
aws ssm get-inventory-schema --type-name Custom:custom_type_name
```

5. Führen Sie den folgenden Befehl aus, um einen Bestandstyp zu löschen.

```
aws ssm delete-inventory --type-name "Custom:custom_type_name" --schema-delete-
option "DeleteSchema"
```

Das System löscht das Schema und alle Bestandsdaten für den angegebenen benutzerdefinierten Typ.

Das System gibt unter anderem folgende Informationen zurück

```
{
   "DeletionId":"system_generated_deletion_ID",
   "DeletionSummary":{
      "RemainingCount":3,
      "SummaryItems":[
         {
            "Count":2,
            "RemainingCount":2,
            "Version":"1.0"
         },
         {
            "Count":1,
            "RemainingCount":1,
            "Version":"2.0"
         }
      ],
      "TotalCount":3
   },
   "TypeName":"Custom:custom_type_name"
}
```

Anzeigen des Löschstatus

Sie können den Status eines Löschvorgangs mithilfe des `describe-inventory-deletions` AWS CLI Befehls überprüfen. Sie können eine Lösch-ID angeben, um den Status eines bestimmten Löschvorgangs anzuzeigen. Wenn Sie keine Lösch-ID angeben, wird eine Liste aller Löschvorgänge der letzten 30 Tage angezeigt.

1. Führen Sie den folgenden Befehl aus, um den Status eines Löschvorgangs anzuzeigen. Das System gibt in der Übersicht über gelöschten Bestand die Lösch-ID zurück.

```
aws ssm describe-inventory-deletions --deletion-id system_generated_deletion_ID
```

Das System gibt den aktuellen Status zurück. Der Löschvorgang ist möglicherweise noch nicht abgeschlossen. Das System gibt unter anderem folgende Informationen zurück

```
{"InventoryDeletions":
  [
    {"DeletionId": "system_generated_deletion_ID",
     "DeletionStartTime": 1521744844,
     "DeletionSummary":
      {"RemainingCount": 1,
       "SummaryItems":
        [
          {"Count": 1,
           "RemainingCount": 1,
           "Version": "1.0"}
        ],
       "TotalCount": 1},
     "LastStatus": "InProgress",
     "LastStatusMessage": "The Delete is in progress",
     "LastStatusUpdateTime": 1521744844,
     "TypeName": "Custom:custom_type_name"}
  ]
}
```

Wenn der Löschvorgang abgeschlossen ist, wird in `LastStatusMessage` die Meldung "Deletion is successful" (Löschvorgang erfolgreich) angezeigt.

```
{"InventoryDeletions":
  [
    {"DeletionId": "system_generated_deletion_ID",
     "DeletionStartTime": 1521744844,
     "DeletionSummary":
      {"RemainingCount": 0,
       "SummaryItems":
        [
          {"Count": 1,
           "RemainingCount": 0,
           "Version": "1.0"}
        ],
       "TotalCount": 1},
```

```
     "LastStatus": "Complete",
     "LastStatusMessage": "Deletion is successful",
     "LastStatusUpdateTime": 1521745253,
     "TypeName": "Custom:custom_type_name"}
  ]
}
```

2. Führen Sie den folgenden Befehl aus, um eine Liste aller Löschvorgänge der letzten 30 Tage anzuzeigen.

```
aws ssm describe-inventory-deletions --max-results a number
```

```
{"InventoryDeletions":
  [
    {"DeletionId": "system_generated_deletion_ID",
     "DeletionStartTime": 1521682552,
     "DeletionSummary":
      {"RemainingCount": 0,
       "SummaryItems":
        [
          {"Count": 1,
           "RemainingCount": 0,
           "Version": "1.0"}
        ],
       "TotalCount": 1},
     "LastStatus": "Complete",
     "LastStatusMessage": "Deletion is successful",
     "LastStatusUpdateTime": 1521682852,
     "TypeName": "Custom:custom_type_name"},
    {"DeletionId": "system_generated_deletion_ID",
     "DeletionStartTime": 1521744844,
     "DeletionSummary":
      {"RemainingCount": 0,
       "SummaryItems":
        [
          {"Count": 1,
           "RemainingCount": 0,
           "Version": "1.0"}
        ],
       "TotalCount": 1},
     "LastStatus": "Complete",
     "LastStatusMessage": "Deletion is successful",
```

```
     "LastStatusUpdateTime": 1521745253,
     "TypeName": "Custom:custom_type_name"},
    {"DeletionId": "system_generated_deletion_ID",
     "DeletionStartTime": 1521680145,
     "DeletionSummary":
      {"RemainingCount": 0,
       "SummaryItems":
        [
          {"Count": 1,
           "RemainingCount": 0,
           "Version": "1.0"}
        ],
       "TotalCount": 1},
     "LastStatus": "Complete",
     "LastStatusMessage": "Deletion is successful",
     "LastStatusUpdateTime": 1521680471,
     "TypeName": "Custom:custom_type_name"}
  ],
 "NextToken": "next-token"
```

Interpretieren der Übersicht für gelöschten Bestand

Sehen Sie sich das folgende Beispiel an, um den Inhalt der Übersicht für gelöschten Bestand besser zu verstehen. Ein Benutzer hat drei Knoten Benutzerdefiniert: RackSpace Inventar zugewiesen. Die Inventarelemente 1 und 2 verwenden den benutzerdefinierten Typ Version 1.0 (“ SchemaVersion „:"1.0"). Für Inventarartikel 3 wird der benutzerdefinierte Typ Version 2.0 (“ SchemaVersion „:"2.0") verwendet.

RackSpace benutzerdefiniertes Inventar 1

```
{
   "CaptureTime":"2018-02-19T10:48:55Z",
   "TypeName":"CustomType:RackSpace",
   "InstanceId":"i-1234567890",
   "SchemaVersion":"1.0"   "Content":[
      {
         content of custom type omitted
      }
   ]
}
```

RackSpace benutzerdefiniertes Inventar 2

```
{
   "CaptureTime":"2018-02-19T10:48:55Z",
   "TypeName":"CustomType:RackSpace",
   "InstanceId":"i-1234567891",
   "SchemaVersion":"1.0"   "Content":[
      {
         content of custom type omitted
      }
   ]
}
```

RackSpace benutzerdefiniertes Inventar 3

```
{
   "CaptureTime":"2018-02-19T10:48:55Z",
   "TypeName":"CustomType:RackSpace",
   "InstanceId":"i-1234567892",
   "SchemaVersion":"2.0"   "Content":[
      {
         content of custom type omitted
      }
   ]
}
```

Der Benutzer führt den folgenden Befehl aus, um eine Vorschau der zu löschenden Daten anzuzeigen.

```
aws ssm delete-inventory --type-name "Custom:RackSpace" --dry-run
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "DeletionId":"1111-2222-333-444-66666",
   "DeletionSummary":{
      "RemainingCount":3,
      "TotalCount":3,
                TotalCount and RemainingCount are the number of items that would be
 deleted if this was not a dry run. These numbers are the same because the system
 didn't delete anything.
      "SummaryItems":[
         {
```

```
            "Count":2,             The system found two items that use SchemaVersion
 1.0. Neither item was deleted.
            "RemainingCount":2,
            "Version":"1.0"
         },
         {
            "Count":1,             The system found one item that uses SchemaVersion
 1.0. This item was not deleted.
            "RemainingCount":1,
            "Version":"2.0"
         }
      ],

   },
   "TypeName":"Custom:RackSpace"
}
```

Der Benutzer führt den folgenden Befehl aus, um das benutzerdefinierte RackSpace Inventar zu löschen.

> **Note**
>
> Der Fortschritt des Löschvorgangs wird in der Ausgabe dieses Befehls nicht angezeigt. Daher sind `TotalCount` und `RemainingCount` immer identisch, da das System noch nichts gelöscht hat. Sie können den `describe-inventory-deletions`-Befehl verwenden, um den Fortschritt des Löschvorgangs anzuzeigen.

```
aws ssm delete-inventory --type-name "Custom:RackSpace"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "DeletionId":"1111-2222-333-444-7777777",
   "DeletionSummary":{
      "RemainingCount":3,           There are three items to delete
      "SummaryItems":[
         {
            "Count":2,             The system found two items that use SchemaVersion
 1.0.
            "RemainingCount":2,
```

```
            "Version":"1.0"
         },
         {
            "Count":1,              The system found one item that uses SchemaVersion
 2.0.
            "RemainingCount":1,
            "Version":"2.0"
         }
      ],
      "TotalCount":3
   },
   "TypeName":"RackSpace"
}
```

Aktionen zum Löschen von Inventar anzeigen in EventBridge

Sie können Amazon so konfigurieren EventBridge , dass jedes Mal, wenn ein Benutzer benutzerdefiniertes Inventar löscht, ein Ereignis erstellt wird. EventBridge bietet drei Arten von Ereignissen für benutzerdefinierte Vorgänge zum Löschen von Inventar:

- Löschaktion für eine Instance: Ob der benutzerdefinierte Bestand für einen bestimmten verwalteten Knoten erfolgreich gelöscht wurde oder nicht.
- Löschaktion-Übersicht: Eine Übersicht über die Löschaktion.
- Warnung für deaktivierten benutzerdefinierten Inventartyp: Ein Warnungsereignis, wenn ein Benutzer den PutInventoryAPI-Vorgang für eine Version des benutzerdefinierten Inventartyps aufgerufen hat, die zuvor deaktiviert war.

Hier finden Sie Beispiele für jedes Ereignis.

Löschaktion für eine Instance

```
{
   "version":"0",
   "id":"998c9cde-56c0-b38b-707f-0411b3ff9d11",
   "detail-type":"Inventory Resource State Change",
   "source":"aws.ssm",
   "account":"478678815555",
   "time":"2018-05-24T22:24:34Z",
   "region":"us-east-1",
   "resources":[
```

```
      "arn:aws:ssm:us-east-1:478678815555:managed-instance/i-0a5feb270fc3f0b97"
   ],
   "detail":{
      "action-status":"succeeded",
      "action":"delete",
      "resource-type":"managed-instance",
      "resource-id":"i-0a5feb270fc3f0b97",
      "action-reason":"",
      "type-name":"Custom:MyInfo"
   }
}
```

Löschaktion-Übersicht

```
{
   "version":"0",
   "id":"83898300-f576-5181-7a67-fb3e45e4fad4",
   "detail-type":"Inventory Resource State Change",
   "source":"aws.ssm",
   "account":"478678815555",
   "time":"2018-05-24T22:28:25Z",
   "region":"us-east-1",
   "resources":[

   ],
   "detail":{
      "action-status":"succeeded",
      "action":"delete-summary",
      "resource-type":"managed-instance",
      "resource-id":"",
      "action-reason":"The delete for type name Custom:MyInfo was completed. The
 deletion summary is: {\"totalCount\":2,\"remainingCount\":0,\"summaryItems\":
[{\"version\":\"1.0\",\"count\":2,\"remainingCount\":0}]}",
      "type-name":"Custom:MyInfo"
   }
}
```

Warnung für einen deaktivierten benutzerdefinierten Bestandstyp

```
{
   "version":"0",
   "id":"49c1855c-9c57-b5d7-8518-b64aeeef5e4a",
   "detail-type":"Inventory Resource State Change",
```

```
   "source":"aws.ssm",
   "account":"478678815555",
   "time":"2018-05-24T22:46:58Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-east-1:478678815555:managed-instance/i-0ee2d86a2cfc371f6"
   ],
   "detail":{
      "action-status":"failed",
      "action":"put",
      "resource-type":"managed-instance",
      "resource-id":"i-0ee2d86a2cfc371f6",
      "action-reason":"The inventory item with type name Custom:MyInfo was sent with a
 disabled schema version 1.0. You must send a version greater than 1.0",
      "type-name":"Custom:MyInfo"
   }
}
```

Gehen Sie wie folgt vor, um eine EventBridge Regel für benutzerdefinierte Inventarlöschvorgänge zu erstellen. In diesem Verfahren wird gezeigt, wie Sie eine Regel erstellen, die Benachrichtigungen für Löschvorgänge an benutzerdefiniertem Bestand an ein Amazon SNS-Thema sendet. Bevor Sie beginnen, stellen Sie sicher, dass Sie ein Amazon SNS-Thema haben oder ein neues erstellen. Weitere Informationen dazu erhalten Sie unter Erste Schritte im Amazon Simple Notification Service Entwicklerhandbuch.

So konfigurieren Sie EventBridge das Löschen von Inventarvorgängen

1. Öffnen Sie die EventBridge Amazon-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Regeln aus.
3. Wählen Sie Regel erstellen aus.
4. Geben Sie einen Namen und eine Beschreibung für die Regel ein.

   Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben Region und auf demselben Event Bus haben.
5. Wählen Sie für Event Bus den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel auf übereinstimmende Ereignisse reagiert, die von Ihnen selbst stammen AWS-Konto, wählen Sie Standard. Wenn ein AWS-Service in Ihrem Konto ein Ereignis ausgibt, wird es immer an den Standard-Event-Bus Ihres Kontos weitergeleitet.
6. Bei Regeltyp wählen Sie Regel mit einem Ereignismuster aus.

7. Wählen Sie Weiter aus.
8. Wählen Sie als Eventquelle AWS Events oder EventBridge Partnerevents aus.
9. Wählen Sie im Abschnitt Ereignismuster die Option Ereignismusterformular aus.
10. Als Event source (Ereignisquelle) wählen Sie AWS -Services aus.
11. Wählen Sie für AWS service (-Service), die Option Systems Manager aus.
12. Wählen Sie Inventory für Event type (Ereignistyp).
13. Für Specific detail type(s) (Spezifische(r) Detail-Typ(en)), wählen Sie Inventory Resource State Change (Inventar-Ressourcen-Statusänderung).
14. Wählen Sie Weiter aus.
15. Bei Zieltypen wählen Sie AWS -Service aus.
16. Wählen Sie für Select a target (Ziel auswählen), die Option SNS topic (SNS-Thema), und dann für Topic (Thema), Ihr Thema aus.
17. Vergewissern Sie sich, dass im Abschnitt Additional settings (Zusätzliche Einstellungen) für Configure target input (Zieleingabe konfigurieren) die Option Matched event (Übereinstimmendes Ereignis) ausgewählt ist.
18. Wählen Sie Weiter aus.
19. (Optional) Geben Sie ein oder mehrere Tags für die Regel ein. Weitere Informationen finden Sie unter Tagging Your Amazon EventBridge Resources im EventBridge Amazon-Benutzerhandbuch.
20. Wählen Sie Weiter aus.
21. Überprüfen Sie die Details der Regel und wählen Sie dann Regel erstellen aus.

Anzeigen von Bestandsverlauf und Änderungsnachverfolgung

Sie können den AWS Systems Manager Inventarverlauf und die Änderungsnachverfolgung für alle Ihre verwalteten Knoten anzeigen, indem Sie AWS Config AWS Config bietet einen detaillierten Überblick über die Konfiguration der AWS Ressourcen in Ihrem AWS-Konto. Dazu gehört auch, wie die Ressourcen jeweils zueinander in Beziehung stehen und wie sie in der Vergangenheit konfiguriert wurden, damit Sie sehen können, wie sich die Konfigurationen und Beziehungen im Laufe der Zeit verändern. Um den Bestandsverlauf und die Änderungsverfolgung anzuzeigen, müssen Sie die folgenden Ressourcen in AWS Config aktivieren:

- SSM: ManagedInstanceInventory

- SSM: PatchCompliance
- SSM: AssociationCompliance
- SSM: FileData

> **Note**
>
> Beachten Sie die folgenden wichtigen Hinweise zum Inventory-Verlauf und der Änderungsverfolgung:
>
> - Wenn Sie Änderungen in Ihrem System nachverfolgen AWS Config möchten, müssen Sie Systems Manager Inventory so konfigurieren, dass `AWS:File` Metadaten erfasst werden, sodass Sie Dateiänderungen in AWS Config (`SSM:FileData`) anzeigen können. Wenn Sie das nicht tun, dann verfolgt AWS Config keine Dateiänderungen auf Ihrem System.
> - Wenn Sie SSM: PatchCompliance und SSM: aktivierenAssociationCompliance, können Sie den Compliance-Verlauf und die Änderungsverfolgung von Systems Patch Manager Manager-Patches und Systems State Manager Manager-Verknüpfungen einsehen. Weitere Informationen über die Compliance-Verwaltung für diese Ressourcen finden Sie unter Erfahren Sie mehr über Compliance.

Im folgenden Verfahren wird beschrieben, wie Sie den Inventarverlauf und die Aufzeichnung von Änderungen mithilfe AWS Config von () aktivieren. AWS Command Line Interface AWS CLI Weitere Informationen zur Auswahl und Konfiguration dieser Ressourcen finden Sie unter Auswahl welcher AWS Config Ressourceneinträge im AWS Config Entwicklerhandbuch. AWS Config Informationen zu AWS Config -Preisen erhalten Sie unter Pricing (Preise für WAF).

Bevor Sie beginnen

AWS Config benötigt AWS Identity and Access Management (IAM) -Berechtigungen, um Konfigurationsdetails zu Systems Manager Manager-Ressourcen abzurufen. Im folgenden Verfahren müssen Sie einen Amazon-Ressourcennamen (ARN) für eine IAM-Rolle angeben, die AWS Config Berechtigungen für Systems Manager Manager-Ressourcen erteilt. Sie können die verwaltete `AWS_ConfigRole`-Richtlinie der IAM-Rolle hinzufügen, die Sie AWS Config zuweisen. Weitere Informationen zu dieser Rolle finden Sie unter AWS managed policy: AWS_ ConfigRole im AWS Config Developer Guide. Weitere Informationen zum Erstellen einer IAM-Rolle und dem Zuweisen der `AWS_ConfigRole`-verwalteten Richtlinie zu dieser Rolle finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

So aktivieren Sie den Inventarverlauf und die Aufzeichnung von Änderungen in AWS Config

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Kopieren Sie das folgende JSON-Beispiel in einen einfachen Texteditor und speichern Sie die Datei unter dem Namen recordingGroup.json.

   ```
   {
      "allSupported":false,
      "includeGlobalResourceTypes":false,
      "resourceTypes":[
         "AWS::SSM::AssociationCompliance",
         "AWS::SSM::PatchCompliance",
         "AWS::SSM::ManagedInstanceInventory",
         "AWS::SSM::FileData"
      ]
   }
   ```

3. Führen Sie den folgenden Befehl aus, um die Datei recordingGroup.json in AWS Config zu laden.

   ```
   aws configservice put-configuration-recorder --configuration-recorder
    name=myRecorder,roleARN=arn:aws:iam::123456789012:role/myConfigRole --recording-
   group file://recordingGroup.json
   ```

4. Führen Sie den folgenden Befehl aus, um die Erfassung des Bestandsverlaufs und der Änderungsnachverfolgung zu starten.

   ```
   aws configservice start-configuration-recorder --configuration-recorder-
   name myRecorder
   ```

Nachdem Sie die Verlaufs- und Änderungsverfolgung konfiguriert haben, können Sie weitere Details des Verlaufs für einen bestimmten verwalteten Knoten mit der Schaltfläche AWS Config in der Systems-Manager-Konsole anzeigen. Sie können entweder von der Seite Managed Instances (Verwaltete Instances) oder der Inventory(Inventory)-Seite aus auf die Schaltfläche AWS Config

zugreifen. Je nach Bildschirmgröße müssen Sie möglicherweise einen Bildlauf nach rechts auf der Seite ausführen, um die Schaltfläche anzuzeigen.

Anhalten der Datenerfassung und Löschen von Bestandsdaten

Wenn Sie AWS Systems Manager-Inventory nicht mehr verwenden möchten, um Metadaten über Ihre AWS-Ressourcen anzuzeigen, können Sie die Datenerfassung beenden und bereits erfasste Daten löschen. Dieser Abschnitt enthält folgende Informationen.

Themen

- Beenden der Datensammlung
- Löschen einer Inventory Resource Data Sync

Beenden der Datensammlung

Wenn Sie Systems Manager zunächst so konfigurieren, dass Bestandsdaten erfasst werden, erstellt das System eine State Manager-Zuordnung, die den Zeitplan und die Ressourcen definiert, aus denen Metadaten gesammelt werden sollen. Sie können die Datenerfassung stoppen, indem Sie alle State Manager-Zuordnungen löschen, die das `AWS-GatherSoftwareInventory`-Dokument verwenden.

Löschen einer Bestandszuordnung

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager aus.
3. Wählen Sie eine Zuordnung aus, die das `AWS-GatherSoftwareInventory`-Dokument nutzt und wählen Sie dann Delete (Löschen).
4. Wiederholen Sie Schritt drei für alle verbleibenden Zuordnungen, die das `AWS-GatherSoftwareInventory`-Dokument verwenden.

Löschen einer Inventory Resource Data Sync

Wenn Sie AWS Systems Manager-Inventory nicht mehr zum Anzeigen von Metadaten Ihrer AWS-Ressourcen verwenden möchten, empfehlen wir auch, Resource Data Syncs zu löschen, die für die Bestandsdatenerfassung verwendet werden.

Löschen einer Inventory Resource Data Sync

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Inventory.
3. Wählen Sie Resource Data Syncs (Ressourcen-Datensynchronisierung).
4. Wählen Sie eine Synchronisierung aus der Liste aus.

   > **Important**
   >
   > Stellen Sie sicher, dass Sie die Synchronisierung für Inventory auswählen. Systems Manager unterstützt die Ressourcendaten-Synchronisierung für mehrere Tools. Wenn Sie die falsche Synchronisierung wählen, könnten Sie die Datenaggregation für Systems Manager Explorer oder Systems Manager Compliance unterbrechen.

5. Wählen Sie Delete (Löschen)
6. Wiederholen Sie diese Schritte für alle verbleibenden Resource Data Syncs, die Sie löschen möchten.
7. Löschen Sie den Amazon Simple Storage Service (Amazon S3)-Bucket, in dem die Daten gespeichert wurden. Weitere Informationen zum Löschen eines Amazon S3-Buckets finden Sie unter Deleting a bucket (Löschen eines Buckets).

Zuweisen benutzerdefinierter Bestands-Metadaten zu einem verwalteten Knoten

Das folgende Verfahren führt Sie durch den Prozess der Verwendung des AWS Systems Manager PutInventoryAPI-Vorgangs, um einem verwalteten Knoten benutzerdefinierte Inventarmetadaten zuzuweisen. In diesem Beispiel werden einem Knoten Informationen zum Rack-Standort zugewiesen. Weitere Informationen zum benutzerdefinierten Bestand finden Sie unter Arbeiten mit benutzerdefiniertem Bestand

So weisen Sie benutzerdefinierte Bestands-Metadaten zu einem verwalteten Knoten zu

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um einem Knoten Informationen zum Rack-Standort zuzuweisen.

   Linux

   ```
   aws ssm put-inventory --instance-id "ID" --items '[{"CaptureTime":
    "2016-08-22T10:01:01Z", "TypeName": "Custom:RackInfo", "Content":[{"RackLocation":
    "Bay B/Row C/Rack D/Shelf E"}], "SchemaVersion": "1.0"}]'
   ```

   Windows

   ```
   aws ssm put-inventory --instance-id "ID" --items
    "TypeName=Custom:RackInfo,SchemaVersion=1.0,CaptureTime=2021-05-22T10:01:01Z,Content=[{Rac
    B/Row C/Rack D/Shelf F'}]"
   ```

3. Führen Sie den folgenden Befehl aus, um die Einträge eines benutzerdefinierten Bestands für diesen Knoten anzuzeigen.

   ```
   aws ssm list-inventory-entries --instance-id ID --type-name "Custom:RackInfo"
   ```

   Das System gibt die folgenden Informationen zurück.

   ```
   {
       "InstanceId": "ID",
       "TypeName": "Custom:RackInfo",
       "Entries": [
           {
               "RackLocation": "Bay B/Row C/Rack D/Shelf E"
           }
       ],
       "SchemaVersion": "1.0",
       "CaptureTime": "2016-08-22T10:01:01Z"
   }
   ```

4. Führen Sie den folgenden Befehl aus, um das benutzerdefinierte Bestandsschema anzuzeigen.

   ```
   aws ssm get-inventory-schema --type-name Custom:RackInfo
   ```

   Das System gibt die folgenden Informationen zurück.

```
{
    "Schemas": [
        {
            "TypeName": "Custom:RackInfo",
            "Version": "1.0",
            "Attributes": [
                {
                    "DataType": "STRING",
                    "Name": "RackLocation"
                }
            ]
        }
    ]
}
```

Verwenden von AWS CLI , um die Inventardatenerfassung zu konfigurieren

Die folgenden Verfahren führen Sie durch die Schritte zur Konfiguration von AWS Systems Manager Inventory für das Erfassen von Metadaten aus Ihren verwalteten Knoten. Wenn Sie die Erfassung durch Inventory konfigurieren, erstellen Sie zuerst eine Systems Manager State Manager-Zuordnung. Systems Manager erfasst die Bestandsdaten, wenn der Zuordnungsstatus ausgeführt wird. Wenn Sie den Zuordnungsstatus nicht zuerst erstellen und versuchen, das `aws:softwareInventory`-Plugin z. B. mit Systems Manager Run Command aufzurufen, gibt das System den folgenden Fehler aus:

`The aws:softwareInventory plugin can only be invoked via ssm-associate.`

> **Note**
>
> Pro Knoten kann nur jeweils eine Bestandszuordnungs konfiguriert werden. Wenn Sie einen Knoten mit zwei oder mehr Bestandszuordnungen konfigurieren, wird die Zuordnung nicht ausgeführt und es werden keine Bestandsdaten erfasst.

Schnelle Konfiguration aller Ihrer verwalteten Knoten für Inventory (CLI)

Sie können schnell alle verwalteten Knoten in Ihrer AWS-Konto und in der aktuellen Region konfigurieren, um Inventardaten zu sammeln. Dieser Vorgang wird als „Erstellen einer globalen Bestandszuordnung“ bezeichnet. Um mithilfe von eine globale Inventarzuordnung zu erstellen AWS

CLI, verwenden Sie die Platzhalteroption für den `instanceIds` Wert, wie im folgenden Verfahren gezeigt.

So konfigurieren Sie das Inventar für alle verwalteten Knoten in Ihrer AWS-Konto und in der aktuellen Region (CLI)

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus.

   Linux & macOS

   ```
   aws ssm create-association \
   --name AWS-GatherSoftwareInventory \
   --targets Key=InstanceIds,Values=* \
   --schedule-expression "rate(1 day)" \
   --parameters
    applications=Enabled,awsComponents=Enabled,customInventory=Enabled,instanceDetailedInfo
   ```

   Windows

   ```
   aws ssm create-association ^
   --name AWS-GatherSoftwareInventory ^
   --targets Key=InstanceIds,Values=* ^
   --schedule-expression "rate(1 day)" ^
   --parameters
    applications=Enabled,awsComponents=Enabled,customInventory=Enabled,instanceDetailedInfo
   ```

> **Note**
>
> Mit diesem Befehl kann Inventory keine Metadaten für die Windows-Registrierung oder Dateien sammeln. Um diese Datentypen in den Bestand aufzunehmen, fahren Sie mit dem nächsten Schritt fort.

Manuelle Konfiguration von Inventory auf Ihren verwalteten Knoten (CLI)

Gehen Sie wie folgt vor, um AWS Systems Manager Inventar auf Ihren verwalteten Knoten mithilfe von Knoten IDs oder Tags manuell zu konfigurieren.

So konfigurieren Sie Ihre verwalteten Knoten manuell für Inventory (CLI)

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um eine State Manager-Zuordnung zu erstellen, die Systems Manager Inventory auf dem Knoten ausführt. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen. Mit diesem Befehl wird der Service so konfiguriert, dass er alle sechs Stunden ausgeführt wird und Metadaten über Netzwerkkonfigurationen, Windows Update und Anwendungen aus einem Knoten erfasst.

   Linux & macOS

   ```
   aws ssm create-association \
   --name "AWS-GatherSoftwareInventory" \
   --targets "Key=instanceids,Values=an_instance_ID" \
   --schedule-expression "rate(240 minutes)" \
   --output-location "{ \"S3Location\": { \"OutputS3Region\": \"region_ID,
    for example us-east-2\", \"OutputS3BucketName\": \"amzn-s3-demo-bucket\",
    \"OutputS3KeyPrefix\": \"Test\" } }" \
   --parameters "networkConfig=Enabled,windowsUpdates=Enabled,applications=Enabled"
   ```

   Windows

   ```
   aws ssm create-association ^
   --name "AWS-GatherSoftwareInventory" ^
   --targets "Key=instanceids,Values=an_instance_ID" ^
   --schedule-expression "rate(240 minutes)" ^
   --output-location "{ \"S3Location\": { \"OutputS3Region\": \"region_ID,
    for example us-east-2\", \"OutputS3BucketName\": \"amzn-s3-demo-bucket\",
    \"OutputS3KeyPrefix\": \"Test\" } }" ^
   --parameters "networkConfig=Enabled,windowsUpdates=Enabled,applications=Enabled"
   ```

Das System gibt die folgenden Informationen zurück.

```
{
    "AssociationDescription": {
        "ScheduleExpression": "rate(240 minutes)",
        "OutputLocation": {
            "S3Location": {
                "OutputS3KeyPrefix": "Test",
                "OutputS3BucketName": "Test bucket",
                "OutputS3Region": "us-east-2"
            }
        },
        "Name": "The name you specified",
        "Parameters": {
            "applications": [
                "Enabled"
            ],
            "networkConfig": [
                "Enabled"
            ],
            "windowsUpdates": [
                "Enabled"
            ]
        },
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "AssociationId": "1a2b3c4d5e6f7g-1a2b3c-1a2b3c-1a2b3c-1a2b3c4d5e6f7g",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1480544990.06,
        "Date": 1480544990.06,
        "Targets": [
            {
                "Values": [
                   "i-02573cafcfEXAMPLE"
                ],
                "Key": "InstanceIds"
            }
        ]
    }
```

```
}
```

Sie können große Gruppen von Knoten ansprechen, indem Sie den `Targets` Parameter mit EC2 Tags verwenden. Sehen Sie sich das folgende Beispiel an.

Linux & macOS

```
aws ssm create-association \
--name "AWS-GatherSoftwareInventory" \
--targets "Key=tag:Environment,Values=Production" \
--schedule-expression "rate(240 minutes)" \
--output-location "{ \"S3Location\": { \"OutputS3Region\": \"us-east-2\",
 \"OutputS3BucketName\": \"amzn-s3-demo-bucket\", \"OutputS3KeyPrefix\": \"Test
\" } }" \
--parameters "networkConfig=Enabled,windowsUpdates=Enabled,applications=Enabled"
```

Windows

```
aws ssm create-association ^
--name "AWS-GatherSoftwareInventory" ^
--targets "Key=tag:Environment,Values=Production" ^
--schedule-expression "rate(240 minutes)" ^
--output-location "{ \"S3Location\": { \"OutputS3Region\": \"us-east-2\",
 \"OutputS3BucketName\": \"amzn-s3-demo-bucket\", \"OutputS3KeyPrefix\": \"Test
\" } }" ^
--parameters "networkConfig=Enabled,windowsUpdates=Enabled,applications=Enabled"
```

Sie können auch Dateien und Windows-Registry-Schlüssel auf einem Windows Server-Knoten inventarisieren, indem Sie die Bestandstypen `files` und `windowsRegistry` mit Ausdrücken verwenden. Weitere Informationen zu diesen Bestandstypen finden Sie unter Arbeiten mit Datei- und Windows-Registrierungsbestand.

Linux & macOS

```
aws ssm create-association \
--name "AWS-GatherSoftwareInventory" \
--targets "Key=instanceids,Values=i-0704358e3a3da9eb1" \
--schedule-expression "rate(240 minutes)" \
```

```
--parameters '{"files":["[{\"Path\": \"C:\\Program Files\", \"Pattern\":
 [\"*.exe\"], \"Recursive\": true}]"], "windowsRegistry": ["[{\"Path\":
\"HKEY_LOCAL_MACHINE\\Software\\Amazon\", \"Recursive\":true}]"]}' \
--profile dev-pdx
```

Windows

```
aws ssm create-association ^
--name "AWS-GatherSoftwareInventory" ^
--targets "Key=instanceids,Values=i-0704358e3a3da9eb1" ^
--schedule-expression "rate(240 minutes)" ^
--parameters '{"files":["[{\"Path\": \"C:\\Program Files\", \"Pattern\":
 [\"*.exe\"], \"Recursive\": true}]"], "windowsRegistry": ["[{\"Path\":
\"HKEY_LOCAL_MACHINE\\Software\\Amazon\", \"Recursive\":true}]"]}' ^
--profile dev-pdx
```

3. Führen Sie den folgenden Befehl aus, um den Zuordnungsstatus anzuzeigen.

```
aws ssm describe-instance-associations-status --instance-id an_instance_ID
```

Das System gibt die folgenden Informationen zurück.

```
{
"InstanceAssociationStatusInfos": [
         {
            "Status": "Pending",
            "DetailedStatus": "Associated",
            "Name": "reInvent2016PolicyDocumentTest",
            "InstanceId": "i-1a2b3c4d5e6f7g",
            "AssociationId": "1a2b3c4d5e6f7g-1a2b3c-1a2b3c-1a2b3c-1a2b3c4d5e6f7g",
            "DocumentVersion": "1"
        }
]
}
```

Walkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten

In der folgenden exemplarischen Vorgehensweise wird beschrieben, wie Sie mithilfe von AWS Command Line Interface (AWS CLI) eine Konfiguration für die AWS Systems Manager

Ressourcendatensynchronisierung für Inventar erstellen. Eine Ressourcen-Datensynchronisierung portiert alle Bestandsdaten aus verwalteten Knoten automatisch in einen zentralen Amazon Simple Storage Service (Amazon S3)-Bucket. Die Synchronisierung aktualisiert die Daten in dem zentralen Amazon S3-Bucket automatisch, sobald neue Bestandsdaten erfasst werden.

In dieser exemplarischen Vorgehensweise wird auch beschrieben, wie Sie Amazon Athena und Amazon Quick Suite verwenden, um die aggregierten Daten abzufragen und zu analysieren. Informationen zum Erstellen einer Ressourcendatensynchronisierung mithilfe von Systems Manager finden Sie unterWalkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten. AWS-Managementkonsole Informationen zum Abfragen von Inventar von mehreren AWS-Regionen Konten mithilfe von Systems Manager finden Sie AWS-Managementkonsole unterAbfragen von Bestandsdaten aus mehreren Regionen und Konten.

> **Note**
>
> Diese Anleitung enthält Informationen dazu, wie die Synchronisierung mit AWS Key Management Service (AWS KMS) verschlüsselt werden kann. Inventory erfasst keine personenbezogenen, geschützten oder vertraulichen Daten, d. h. die Verschlüsselung ist optional. Weitere Informationen zu AWS KMS finden Sie im AWS Key Management Service Entwicklerhandbuch.

Bevor Sie beginnen

Überprüfen oder erledigen Sie die folgenden Aufgaben, bevor Sie mit dem Walkthrough in diesem Abschnitt beginnen:

- Sammeln Sie Bestandsdaten von Ihren verwalteten Knoten. Für die Zwecke der Abschnitte Amazon Athena und Amazon Quick Suite in dieser exemplarischen Vorgehensweise empfehlen wir Ihnen, Anwendungsdaten zu sammeln. Weitere Informationen zur Erfassung von Bestandsdaten finden Sie unter Konfigurieren der Bestandserfassung oder Verwenden von AWS CLI , um die Inventardatenerfassung zu konfigurieren.
- (Optional) Wenn die Inventardaten in einem Amazon Simple Storage Service (Amazon S3) - Bucket gespeichert werden, der die Verschlüsselung AWS Key Management Service (AWS KMS) verwendet, müssen Sie auch Ihr IAM-Konto und die `Amazon-GlueServiceRoleForSSM` Servicerolle für die AWS KMS Verschlüsselung konfigurieren. Wenn Sie Ihr IAM-Konto und diese Rolle nicht konfigurieren, wird Systems Manage `Cannot load Glue tables` anzeigen, wenn Sie die Registerkarte Detailed View (Detailansicht) in der Konsole wählen. Weitere

Informationen finden Sie unter (Optional) Konfigurieren Sie Berechtigungen für die Anzeige AWS KMS verschlüsselter Daten.

- (Optional) Wenn Sie die Ressourcendatensynchronisierung mithilfe von verschlüsseln möchten AWS KMS, müssen Sie entweder einen neuen Schlüssel erstellen, der die folgende Richtlinie enthält, oder Sie müssen einen vorhandenen Schlüssel aktualisieren und diese Richtlinie hinzufügen.

  JSON

```
{
    "Version":"2012-10-17",
    "Id": "ssm-access-policy",
    "Statement": [
        {
            "Sid": "ssm-access-policy-statement",
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Resource": "arn:aws:kms:us-east-1:123456789012:key/KMS_key_id",
            "Condition": {
                "StringLike": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:ssm:*:123456789012:resource-data-
sync/*"
                }
            }
        }
    ]
}
```

So erstellen Sie eine Resource Data Sync für Inventory

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Erstellen Sie einen Bucket zum Speichern der aggregierten Bestandsdaten. Weitere Informationen erhalten Sie unter Erstellen eines Buckets im Benutzerhandbuch für Amazon

Simple Storage Service. Notieren Sie sich den Namen des Buckets und den AWS-Region Ort, an dem Sie ihn erstellt haben.

3. Wenn Sie den Bucket erstellt haben, wählen Sie die Registerkarte Permissions aus und wählen Sie dann die Option Bucket Policy.

4. Kopieren Sie die folgende Bucket-Richtlinie in den Richtlinien-Editor. Ersetzen Sie amzn-s3-demo-bucket und *account-id* durch den Namen des Amazon S3-Buckets, den Sie erstellt haben, und eine gültige ID. AWS-Konto Wenn Sie mehrere Konten hinzufügen, fügen Sie für jedes Konto eine zusätzliche Bedingungszeichenfolge und einen ARN hinzu. Entfernen Sie die zusätzlichen Platzhalter aus dem Beispiel, wenn Sie einzelne Konten hinzufügen. Optional können Sie es *bucket-prefix* durch den Namen eines Amazon S3 S3-Präfix (Unterverzeichnis) ersetzen. Wenn Sie kein Präfix erstellt haben, entfernen Sie es *bucket-prefix/* aus dem ARN in der Richtlinie.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": " SSMBucketDelivery",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/bucket-prefix/*/
accountid=111122223333/*"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": [
                        "111122223333",
                        "444455556666",
                        "123456789012",
                        "777788889999"
                    ]
                },
                "ArnLike": {
```

```
                    "aws:SourceArn": [
                        "arn:aws:ssm:*:111122223333:resource-data-sync/*",
                        "arn:aws:ssm:*:444455556666:resource-data-sync/*",
                        "arn:aws:ssm:*:123456789012:resource-data-sync/*",
                        "arn:aws:ssm:*:777788889999:resource-data-sync/*"
                    ]
                }
            }
        }
    ]
}
```

5. (Optional) Wenn Sie die Synchronisierung verschlüsseln möchten, müssen Sie der im vorherigen Schritt aufgeführten Richtlinie die folgenden Bedingungen hinzufügen. Fügen Sie diese zum Abschnitt `StringEquals` hinzu.

```
"s3:x-amz-server-side-encryption":"aws:kms",
"s3:x-amz-server-side-encryption-aws-kms-key-
id":"arn:aws:kms:region:account_ID:key/KMS_key_ID"
```

   Ein Beispiel:

```
"StringEquals": {
          "s3:x-amz-acl": "bucket-owner-full-control",
          "aws:SourceAccount": "account-id",
          "s3:x-amz-server-side-encryption":"aws:kms",
          "s3:x-amz-server-side-encryption-aws-kms-key-
id":"arn:aws:kms:region:account_ID:key/KMS_key_ID"
        }
```

6. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

7. (Optional) Wenn Sie die Synchronisierung verschlüsseln möchten, führen Sie den folgenden Befehl aus, um zu überprüfen, ob die Bucket-Richtlinie die AWS KMS Schlüsselanforderung durchsetzt. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

Linux & macOS

```
aws s3 cp ./A_file_in_the_bucket s3://amzn-s3-demo-bucket/prefix/ \
--sse aws:kms \
--sse-kms-key-id "arn:aws:kms:region:account_ID:key/KMS_key_id" \
--region region, for example, us-east-2
```

Windows

```
aws s3 cp ./A_file_in_the_bucket s3://amzn-s3-demo-bucket/prefix/ ^
    --sse aws:kms ^
    --sse-kms-key-id "arn:aws:kms:region:account_ID:key/KMS_key_id" ^
    --region region, for example, us-east-2
```

8. Führen Sie den folgenden Befehl aus, um eine Resource Data Sync-Konfiguration mit dem zu Beginn dieses Verfahrens erstellten Amazon S3-Bucket zu erstellen. Dieser Befehl erstellt eine Synchronisation aus dem, bei dem AWS-Region Sie angemeldet sind.

> **Note**
>
> Wenn sich der Synchronisierungs- und der Amazon S3-Ziel-Bucket in verschiedenen Regionen befinden, können Kosten für Datenübertragung anfallen. Weitere Informationen finden Sie unter Amazon S3 – Preise.

Linux & macOS

```
aws ssm create-resource-data-sync \
--sync-name a_name \
--s3-destination "BucketName=amzn-s3-demo-bucket,Prefix=prefix_name,
 if_specified,SyncFormat=JsonSerDe,Region=bucket_region"
```

Windows

```
aws ssm create-resource-data-sync ^
--sync-name a_name ^
--s3-destination "BucketName=amzn-s3-demo-bucket,Prefix=prefix_name,
 if_specified,SyncFormat=JsonSerDe,Region=bucket_region"
```

Sie können über den `region`-Parameter angeben, wo die Synchronisierungskonfiguration erstellt werden soll. Im folgenden Beispiel werden Bestandsdaten aus der Region us-west-1 in dem Amazon S3-Bucket in der Region us-west-2 synchronisiert.

Linux & macOS

```
aws ssm create-resource-data-sync \
    --sync-name InventoryDataWest \
    --s3-destination "BucketName=amzn-s3-demo-
bucket,Prefix=HybridEnv,SyncFormat=JsonSerDe,Region=us-west-2"
    --region us-west-1
```

Windows

```
aws ssm create-resource-data-sync ^
--sync-name InventoryDataWest ^
--s3-destination "BucketName=amzn-s3-demo-
bucket,Prefix=HybridEnv,SyncFormat=JsonSerDe,Region=us-west-2" ^ --region us-
west-1
```

(Optional) Wenn Sie die Synchronisierung mit verschlüsseln möchten, führen Sie den folgenden Befehl aus AWS KMS, um die Synchronisierung zu erstellen. Wenn Sie die Synchronisierung verschlüsseln, müssen sich der AWS KMS Schlüssel und der Amazon S3 S3-Bucket in derselben Region befinden.

Linux & macOS

```
aws ssm create-resource-data-sync \
--sync-name sync_name \
--s3-destination "BucketName=amzn-s3-demo-bucket,Prefix=prefix_name,
 if_specified,SyncFormat=JsonSerDe,AWSKMSKeyARN=arn:aws:kms:region:account_ID:key/
KMS_key_ID,Region=bucket_region" \
--region region
```

Windows

```
aws ssm create-resource-data-sync ^
--sync-name sync_name ^
```

```
--s3-destination "BucketName=amzn-s3-demo-bucket,Prefix=prefix_name,
 if_specified,SyncFormat=JsonSerDe,AWSKMSKeyARN=arn:aws:kms:region:account_ID:key/
KMS_key_ID,Region=bucket_region" ^
--region region
```

9. Führen Sie den folgenden Befehl aus, um den Status der Synchronisierungskonfiguration anzuzeigen.

   ```
   aws ssm list-resource-data-sync
   ```

   Wenn Sie die Synchronisierungskonfiguration in einer anderen Region erstellt haben, müssen Sie den `region`-Parameter angeben, wie im folgenden Beispiel gezeigt.

   ```
   aws ssm list-resource-data-sync --region us-west-1
   ```

10. Wenn die Synchronisierungskonfiguration erfolgreich erstellt wurde, prüfen Sie den Ziel-Bucket in Amazon S3. Die Bestandsdaten sollten in der Regel nach nur wenige Minuten angezeigt werden.

Arbeiten mit den Daten in Amazon Athena

Im folgenden Abschnitt wird beschrieben, wie Sie die Daten in Amazon Athena anzeigen und abfragen können. Bevor Sie beginnen, empfehlen wir Ihnen, sich mit Athena vertraut zu machen. Weitere Informationen finden Sie unter Was ist Amazon Athena? und Arbeiten mit Daten im Benutzerhandbuch für Amazon Athena.

Anzeigen und Abfragen von Daten in Amazon Athena

1. Öffnen Sie die Athena-Konsole unter https://console.aws.amazon.com/athena/.
2. Kopieren Sie die folgende Anweisung, fügen Sie sie in den Abfrage-Editor ein und wählen Sie dann Run Query.

   ```
   CREATE DATABASE ssminventory
   ```

   Das System erstellt eine Datenbank mit dem Namen ssminventory.
3. Kopieren Sie die folgende Anweisung, fügen Sie sie in den Abfrage-Editor ein und wählen Sie dann Run Query. Ersetzen Sie amzn-s3-demo-bucket und durch den Namen und das Präfix des *bucket_prefix* Amazon S3 S3-Ziels.

   ```
   CREATE EXTERNAL TABLE IF NOT EXISTS ssminventory.AWS_Application (
   ```

```
Name string,
ResourceId string,
ApplicationType string,
Publisher string,
Version string,
InstalledTime string,
Architecture string,
URL string,
Summary string,
PackageId string
)
PARTITIONED BY (AccountId string, Region string, ResourceType string)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1'
) LOCATION 's3://amzn-s3-demo-bucket/bucket_prefix/AWS:Application/'
```

4. Kopieren Sie die folgende Anweisung, fügen Sie sie in den Abfrage-Editor ein und wählen Sie dann Run Query.

```
MSCK REPAIR TABLE ssminventory.AWS_Application
```

Das System partitioniert die Tabelle.

> **Note**
>
> Wenn Sie Ressourcendatensynchronisationen aus zusätzlichen AWS-Regionen oder erstellen, müssen Sie diesen Befehl erneut ausführen AWS-Konten, um die Partitionen zu aktualisieren. Sie müssen möglicherweise auch Ihre Amazon S3-Bucket-Richtlinie aktualisieren.

5. Sie können Ihre Daten in der Vorschau anzeigen, indem Sie das Ansichtssymbol neben der Tabelle `AWS_Application` wählen.

6. Kopieren Sie die folgende Anweisung, fügen Sie sie in den Abfrage-Editor ein und wählen Sie dann Run Query.

```
SELECT a.name, a.version, count( a.version) frequency
```

```
from aws_application a where
a.name = 'aws-cfn-bootstrap'
group by a.name, a.version
order  by frequency desc
```

Die Abfrage gibt die Anzahl der verschiedenen Versionen von zurück`aws-cfn-bootstrap`, wobei es sich um eine AWS Anwendung handelt, die auf Amazon Elastic Compute Cloud (Amazon EC2) -Instances für Linux vorhanden istmacOS, undWindows Server.

7. Kopieren Sie die folgenden Anweisungen einzeln und fügen Sie sie in den Abfrage-Editor ein, ersetzen Sie amzn-s3-demo-bucket und durch Informationen für Amazon S3 und ***bucket-prefix*** wählen Sie dann Abfrage ausführen. Diese Anweisungen richten zusätzliche Bestandstabellen in Athena ein.

```
CREATE EXTERNAL TABLE IF NOT EXISTS ssminventory.AWS_AWSComponent (
 `ResourceId` string,
  `Name` string,
  `ApplicationType` string,
  `Publisher` string,
  `Version` string,
  `InstalledTime` string,
  `Architecture` string,
  `URL` string
)
PARTITIONED BY (AccountId string, Region string, ResourceType string)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1'
) LOCATION 's3://amzn-s3-demo-bucket/bucket-prefix/AWS:AWSComponent/'
```

```
MSCK REPAIR TABLE ssminventory.AWS_AWSComponent
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS ssminventory.AWS_WindowsUpdate (
  `ResourceId` string,
  `HotFixId` string,
  `Description` string,
  `InstalledTime` string,
  `InstalledBy` string
)
PARTITIONED BY (AccountId string, Region string, ResourceType string)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
```

```
WITH SERDEPROPERTIES (
  'serialization.format' = '1'
) LOCATION 's3://amzn-s3-demo-bucket/bucket-prefix/AWS:WindowsUpdate/'
```

```
MSCK REPAIR TABLE ssminventory.AWS_WindowsUpdate
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS ssminventory.AWS_InstanceInformation (
  `AgentType` string,
  `AgentVersion` string,
  `ComputerName` string,
  `IamRole` string,
  `InstanceId` string,
  `IpAddress` string,
  `PlatformName` string,
  `PlatformType` string,
  `PlatformVersion` string
)
PARTITIONED BY (AccountId string, Region string, ResourceType string)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1'
) LOCATION 's3://amzn-s3-demo-bucket/bucket-prefix/AWS:InstanceInformation/'
```

```
MSCK REPAIR TABLE ssminventory.AWS_InstanceInformation
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS ssminventory.AWS_Network (
  `ResourceId` string,
  `Name` string,
  `SubnetMask` string,
  `Gateway` string,
  `DHCPServer` string,
  `DNSServer` string,
  `MacAddress` string,
  `IPV4` string,
  `IPV6` string
)
PARTITIONED BY (AccountId string, Region string, ResourceType string)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1'
```

```
) LOCATION 's3://amzn-s3-demo-bucket/bucket-prefix/AWS:Network/'
```

```
MSCK REPAIR TABLE ssminventory.AWS_Network
```

```
CREATE EXTERNAL TABLE IF NOT EXISTS ssminventory.AWS_PatchSummary (
  `ResourceId` string,
  `PatchGroup` string,
  `BaselineId` string,
  `SnapshotId` string,
  `OwnerInformation` string,
  `InstalledCount` int,
  `InstalledOtherCount` int,
  `NotApplicableCount` int,
  `MissingCount` int,
  `FailedCount` int,
  `OperationType` string,
  `OperationStartTime` string,
  `OperationEndTime` string
)
PARTITIONED BY (AccountId string, Region string, ResourceType string)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1'
) LOCATION 's3://amzn-s3-demo-bucket/bucket-prefix/AWS:PatchSummary/'
```

```
MSCK REPAIR TABLE ssminventory.AWS_PatchSummary
```

Arbeiten mit den Daten in Amazon Quick Suite

Der folgende Abschnitt bietet eine Übersicht mit Links zum Erstellen einer Visualisierung in Amazon Quick Suite.

Um eine Visualisierung in Amazon Quick Suite zu erstellen

1. Melden Sie sich bei Amazon Quick Suite an und melden Sie sich dann bei der Quick Suite-Konsole an.
2. Erstellen Sie einen Datensatz aus der Tabelle `AWS_Application` sowie aus allen anderen Tabellen, die Sie erstellt haben. Weitere Informationen finden Sie unter Erstellen eines Datensatzes mit Amazon Athena Athena-Daten im Amazon Quick Suite-Benutzerhandbuch.

3. Verknüpfen Sie Tabellen. Sie können z. B. die Spalte `instanceid` aus `AWS_InstanceInformation` verknüpfen, da sie der Spalte `resourceid` in anderen Bestandstabellen entspricht. Weitere Informationen zum Verbinden von Tabellen finden Sie unter Daten verknüpfen im Amazon Quick Suite-Benutzerhandbuch.
4. Erstellen Sie eine Visualisierung. Weitere Informationen finden Sie unter Analysen und Berichte: Visualisieren von Daten in Amazon Quick Sight im Amazon Quick Suite-Benutzerhandbuch.

Fehlerbehebung bei Problemen mit Systems Manager Inventory

Dieses Thema enthält Informationen zur Behebung häufiger Fehler oder Probleme mit AWS Systems Manager Inventar. Informationen zur Behebung von Problemen bei der Anzeige Ihrer Knoten in Systems Manager finden Sie unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten.

Themen

- Mehrere Anwenden aller Zuordnungen mit Dokument 'AWS-GatherSoftwareInventory' werden nicht unterstützt
- Der Inventory-Ausführungsstatus verlässt nie den Status „ausstehend“.
- Das AWS-ListWindowsInventory-Dokument kann nicht ausgeführt werden
- Konsole zeigt das Inventory-Dashboard nicht an | Detailed View (Detailansicht) | Registerkarte „Settings“ (Einstellungen)
- UnsupportedAgent
- Übersprungen
- Fehlgeschlagen
- Fehler bei der Einhaltung von Lagerbeständen für eine EC2 Amazon-Instance
- S3-Bucket-Objekt enthält alte Daten

Mehrere Anwenden aller Zuordnungen mit Dokument '**`AWS-GatherSoftwareInventory`**' werden nicht unterstützt

Ein Fehler `Multiple apply all associations with document 'AWS-GatherSoftwareInventory' are not supported`, der bedeutet, dass eine oder mehrere AWS-Regionen , in denen Sie versuchen, eine Bestandszuordnung für alle Knoten zu konfigurieren, sind bereits mit einer Bestandszuordnung für alle Knoten konfiguriert. Falls erforderlich, können Sie die vorhandene Bestandszuordnung für alle Knoten löschen und anschließend eine neue erstellen. Um vorhandene Bestandszuordnungen anzuzeigen, wählen

Sie State Manager in der Systems Manager Konsole und suchen Sie dann nach Zuordnungen, die das `AWS-GatherSoftwareInventory`-SSM-Dokument verwenden. Wenn die vorhandene Bestandszuordnung für alle Knoten in mehreren Regionen erstellt wurde und Sie eine neue erstellen möchten, müssen Sie die vorhandene Zuordnung aus jeder Region löschen, in der sie vorhanden ist.

Der Inventory-Ausführungsstatus verlässt nie den Status „ausstehend“.

Es gibt zwei Gründe, warum die Bestandsammlung niemals den `Pending` Status annimmt:

- Keine Knoten in der ausgewählten Liste AWS-Region:

  Wenn Sie eine globale Bestandszuordnung mithilfe von Systems Manager Quick Setup erstellen, zeigt der Status der Bestandszuordnung (`AWS-GatherSoftwareInventory`-Dokument) `Pending` an, wenn in der ausgewählten Region keine Knoten verfügbar sind.
- Unzureichende Berechtigungen:

  Eine Bestandszuordnung zeigt `Pending` an, wenn eine oder mehrere Knoten nicht über die Berechtigung zum Ausführen von Systems Manager Inventory verfügen. Stellen Sie sicher, dass das Instance-Profil AWS Identity and Access Management (IAM) die von Amazon SSMManaged InstanceCore verwaltete Richtlinie enthält. Weitere Informationen zum Hinzufügen dieser Richtlinie zu einem Instance-Profil finden Sie unter Alternative Konfiguration für EC2 Instance-Berechtigungen.

  Das Instance-Profil muss mindestens über die folgenden IAM-Berechtigungen verfügen.

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeAssociation",
                "ssm:ListAssociations",
                "ssm:ListInstanceAssociations",
                "ssm:PutInventory",
                "ssm:PutComplianceItems",
                "ssm:UpdateAssociationStatus",
                "ssm:UpdateInstanceAssociationStatus",
                "ssm:UpdateInstanceInformation",
                "ssm:GetDocument",
```

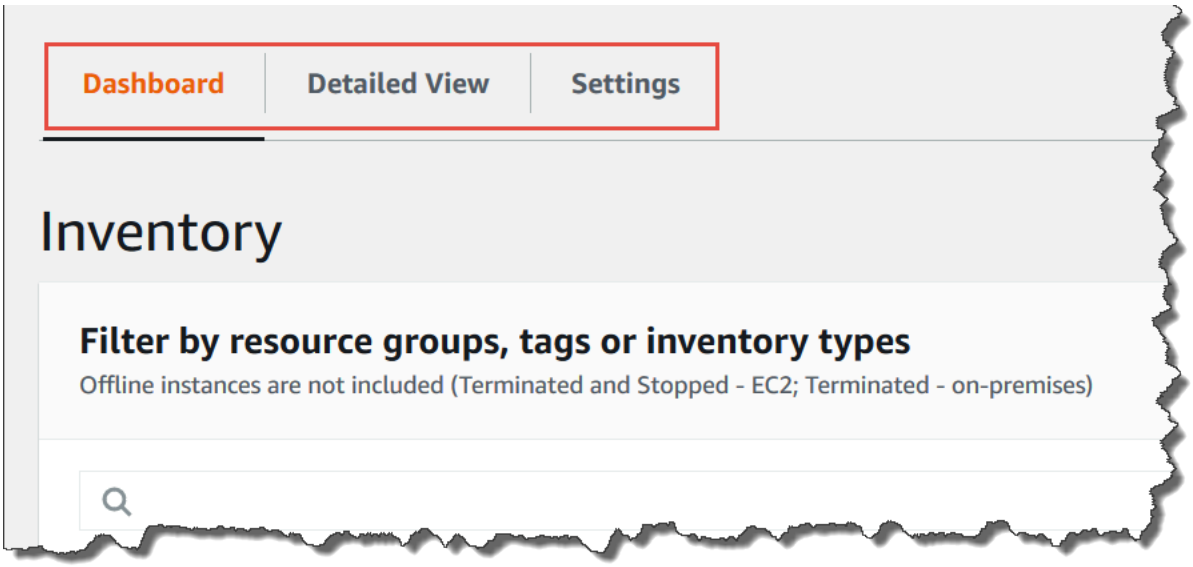
```
                "ssm:DescribeDocument"
            ],
            "Resource": "*"
        }
    ]
}
```

Das **`AWS-ListWindowsInventory`**-Dokument kann nicht ausgeführt werden

Das `AWS-ListWindowsInventory`-Dokument ist veraltet. Verwenden Sie dieses Dokument nicht zur Bestandserfassung. Verwenden Sie stattdessen einen der unter Konfigurieren der Bestandserfassung beschriebenen Prozesse.

Konsole zeigt das Inventory-Dashboard nicht an | Detailed View (Detailansicht) | Registerkarte „Settings“ (Einstellungen)

Die Seite Inventory Detailed View (Detailansicht) ist nur in AWS-Regionen verfügbar, die Amazon Athena anbieten. Wenn die folgenden Registerkarten nicht auf der Seite Systems Manager Inventory angezeigt werden, bedeutet dies, dass Athena nicht in der Region verfügbar ist und Sie die Detailansicht nicht verwenden können, um Daten abzufragen.



UnsupportedAgent

Wenn der detaillierte Status einer Inventarzuordnung und der Zuordnungsstatus Fehlgeschlagen angezeigt wird, ist die Version von AWS Systems Manager SSM Agent auf dem verwalteten Knoten nicht korrekt. UnsupportedAgent Um beispielsweise eine globale Inventarzuordnung zu erstellen (um

alle Knoten in Ihrem Verzeichnis zu inventarisieren AWS-Konto), müssen Sie SSM Agent Version 2.0.790.0 oder höher verwenden. Sie können die Ausführung der Agenten-Version auf jedem Ihrer Knoten auf der Seite Managed Instances (Verwaltete Instances) in der Spalte Agent version (Agent-Version) anzeigen. Weitere Informationen zur Aktualisierung von SSM Agent auf Ihren Knoten finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command.

Übersprungen

Wenn der Status der Bestandszuordnung für einen Knoten Übersprungen anzeigt, bedeutet dies, dass auf diesem Knoten bereits eine Bestandszuordnung mit höherer Priorität ausgeführt wird. Systems Manager folgt einer bestimmten Prioritätsreihenfolge, wenn mehrere Bestandszuordnungen für denselben verwalteten Knoten gelten können.

Prioritätsreihenfolge der Bestandszuordnung

Systems Manager wendet Bestandszuordnungen in der folgenden Prioritätsreihenfolge an:

1. Quick Setup-Bestandszuordnungen – Zuordnungen, die mit Quick Setup und der vereinheitlichten Konsole erstellt wurden. Diese Zuordnungen haben Namen, die mit `AWS-QuickSetup-SSM-CollectInventory-` beginnen und auf alle verwalteten Knoten abzielen.
2. Explizite Bestandszuordnungen – Zuordnungen, die auf bestimmte verwaltete Knoten abzielen und dabei Folgendes verwenden:
   - Instanz IDs
   - Tag-Schlüssel-Wert-Paare
   - AWS Ressourcengruppen
3. Globale Inventarzuordnungen – Zuordnungen, die auf alle verwalteten Knoten abzielen (mithilfe von `--targets "Key=InstanceIds,Values=*"`), aber nicht über Quick Setup erstellt wurden.

Gängige Szenarien

Szenario 1: Die Quick Setup-Zuordnung hat Vorrang vor der expliziten Zuordnung

- Sie haben eine Quick Setup-Inventarzuordnung, die auf alle Instances abzielt
- Sie erstellen eine manuelle Zuordnung, die nach Tag auf bestimmte verwaltete Knoten abzielt
- Ergebnis: Die manuelle Zuordnung zeigt `Skipped` mit detailliertem Status `OverriddenByExplicitInventoryAssociation` an

- Die Quick Setup-Zuordnung erfasst weiterhin Inventar von allen Instances

Szenario 2: Eine explizite Zuordnung hat Vorrang vor der globalen Zuordnung

- Sie haben eine globale Inventarzuordnung, die auf alle Instances abzielt (nicht erstellt von Quick Setup)
- Sie erstellen eine Zuordnung, die eine spezifische Instance anvisiert
- Ergebnis: Die globale Zuordnung zeigt `Skipped` für die speziell anvisierten Instances an
- Die explizite Zuordnung wird auf den anvisierten Instances ausgeführt

Lösungsschritte

Wenn Sie anstelle von Quick Setup Ihre eigene Inventarzuordnung verwenden möchten:

1. Quick Setup-Zuordnungen identifizieren: Gehen Sie in der Systems-Manager-Konsole zu State Manager und suchen Sie nach Zuordnungen, deren Namen mit `AWS-QuickSetup-SSM-CollectInventory-` beginnen.
2. Entfernen der Quick Setup-Konfiguration:
   - Gehen Sie zu Quick Setup in der Systems-Manager-Konsole.
   - Finden Sie Ihre Konfiguration für die Inventarerfassung.
   - Löschen Sie die Quick Setup-Konfiguration (dadurch wird die zugehörige Inventarzuordnung entfernt).

   > **Note**
   >
   > Sie müssen die Zuordnung, die von Quick Setup erstellt wurde, nicht manuell löschen.

3. Stellen Sie sicher, dass Ihre Zuordnung ausgeführt wird: Nachdem Sie die Quick Setup-Konfiguration entfernt haben, sollte Ihre explizite Inventarzuordnung erfolgreich ausgeführt werden.

Wenn Sie das bestehende Verhalten ändern möchten:

- Um alle vorhandenen Inventarzuordnungen anzuzeigen, wählen Sie State Manager in der Systems-Manager-Konsole und suchen Sie dann nach Zuordnungen, die das `AWS-GatherSoftwareInventory`-SSM-Dokument verwenden.

- Denken Sie daran, dass jeder verwaltete Knoten jeweils nur über eine aktive Inventarzuordnung verfügen kann.

> **Important**
>
> - Inventardaten werden weiterhin von übersprungenen Knoten erfasst, wenn die ihnen zugewiesene Inventarzuordnung (mit höherer Priorität) ausgeführt wird.
> - Quick Setup-Inventarzuordnungen haben Vorrang vor allen anderen Typen, auch solchen mit expliziter Ausrichtung.
> - Die detaillierte Statusmeldung `OverriddenByExplicitInventoryAssociation` wird unabhängig vom Zuordnungstyp angezeigt, wenn eine Zuordnung durch eine Zuordnung mit höherer Priorität außer Kraft gesetzt wird.

Fehlgeschlagen

Wenn der Status der Bestandszuordnung für einen Knoten Failed (Fehlgeschlagen) anzeigt, könnte dies bedeuten, dass der Knoten über mehrere ihm zugewiesene Bestandszuordnungen verfügt. Ein Knoten kann jeweils nur über eine zugewiesene Bestandszuordnung verfügen. Eine Inventarvereinigung verwendet das `AWS-GatherSoftwareInventory` AWS Systems Manager Dokument (SSM-Dokument). Sie können den folgenden Befehl ausführen, indem Sie die AWS Command Line Interface (AWS CLI) verwenden, um eine Liste der Verknüpfungen für einen Knoten anzuzeigen.

```
aws ssm describe-instance-associations-status --instance-id instance-ID
```

Fehler bei der Einhaltung von Lagerbeständen für eine EC2 Amazon-Instance

Die Inventarkonformität für eine Amazon Elastic Compute Cloud (Amazon EC2) -Instance kann fehlschlagen, wenn Sie der Instance mehrere Inventarzuordnungen zuweisen.

Um dieses Problem zu beheben, löschen Sie eine oder mehrere Inventarzuordnungen, die der Instance zugewiesen sind. Weitere Informationen finden Sie unter Löschen einer Zuordnung.

> **Note**
>
> Beachten Sie das folgende Verhalten, wenn Sie mehrere Bestandszuordnungen für einen verwalteten Knoten erstellen:

- Jedem Knoten kann eine Inventarzuordnung zugewiesen werden, die auf alle Knoten abzielt (--targets „Key=InstanceIds, Values=*“).
- Jedem Knoten kann auch eine bestimmte Assoziation zugewiesen werden, die entweder Tag-Schlüssel-Wert-Paare oder eine Ressourcengruppe verwendet. AWS
- Wenn einem Knoten mehrere Bestandszuordnungen zugewiesen sind, zeigt der Status Skipped (Übersprungen) für die Zuordnung an, die nicht ausgeführt wurde. Die zuletzt durchgeführte Zuordnung zeigt den aktuellen Status der Bestandszuordnung an.
- Wenn einem Knoten mehrere Bestandszuordnungen zugewiesen sind und jede ein Tag-Schlüssel-Wert-Paar verwendet, können diese Bestandszuordnungen aufgrund des Tag-Konflikts nicht auf dem Knoten ausgeführt werden. Die Zuordnung wird weiterhin auf Knoten ausgeführt, bei denen der Tag-Schlüssel-Wert-Konflikt nicht besteht.

S3-Bucket-Objekt enthält alte Daten

Die Daten im Amazon-S3-Bucket-Objekt werden aktualisiert, wenn die Zuordnung zum Bestand erfolgreich ist und neue Daten entdeckt werden. Das Amazon-S3-Bucket-Objekt wird für jeden Knoten aktualisiert, wenn die Zuordnung läuft und fehlschlägt, aber die Daten innerhalb des Objekts werden in diesem Fall nicht aktualisiert. Die Daten im Amazon-S3-Bucket-Objekt werden nur dann aktualisiert, wenn die Zuordnung erfolgreich verläuft. Wenn die Bestandszuordnung fehlschlägt, sehen Sie alte Daten in dem Amazon-S3-Bucket-Objekt.

AWS Systems Manager Patch Manager

Patch Manager, ein Tool in AWS Systems Manager, automatisiert das Patchen von verwalteten Knoten sowohl mit sicherheitsrelevanten Updates als auch mit anderen Arten von Updates.

> **Note**
>
> Systems Manager bietet Unterstützung für Patch-Richtlinien in Quick Setup, einem Tool in AWS Systems Manager. Die Verwendung von Patch-Richtlinien ist die empfohlene Methode zur Konfiguration Ihrer Patching-Vorgänge. Mit einer einzelnen Patch-Richtlinienkonfiguration können Sie Patches für alle Konten in allen Regionen in Ihrer Organisation, nur für die von Ihnen ausgewählten Konten und Regionen oder für ein einzelnes Konto-Region-Paar definieren. Weitere Informationen finden Sie unter Patch-Richtlinienkonfigurationen in Quick Setup.

Sie können Patch Manager verwenden, um Patches sowohl für Betriebssysteme als auch für Anwendungen durchzuführen. (Unter Windows Server ist der Anwendungssupport auf Updates für Microsoft-Anwendungen beschränkt.) Sie können mit Patch Manager Service Packs auf Windows-Instances installieren und Nebenversionsupgrades auf Linux-Knoten ausführen. Sie können Flotten von Amazon Elastic Compute Cloud (Amazon EC2) -Instances, Edge-Geräten, lokalen Servern und virtuellen Maschinen (VMs) nach Betriebssystemtyp patchen. Dazu gehören unterstützte Versionen mehrerer Betriebssysteme, wie unter Patch Manager-Voraussetzungen aufgeführt. Sie können Instances nur auf Patches hin durchsuchen und dann einen Bericht zu fehlenden Patches anzeigen oder automatisch alle fehlenden Patches installieren. Um mit Patch Manager zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Patch Manager aus.

AWS testet Patches nicht, bevor sie in verfügbar gemacht werden. Patch Manager Unterstützt auch Patch Manager nicht die Aktualisierung von Hauptversionen von Betriebssystemen, wie Windows Server 2016 bis Windows Server 2019 oder Red Hat Enterprise Linux (RHEL) 7.0 auf RHEL 8.0.

Für Linux-basierte Betriebssysteme, die einen Schweregrad für Patches melden, verwendet Patch Manager den vom Software-Publisher gemeldeten Schweregrad für den Update-Hinweis oder den einzelnen Patch. Patch Manager leitet keinen Schweregrad aus Drittquellen wie dem Common Vulnerability Scoring System (CVSS) oder aus Metriken ab, die von der National Vulnerability Database (NVD) veröffentlicht werden.

Welche Vorteile bietet Patch Manager meiner Organisation?

Patch Manager automatisiert den Prozess des Patchens verwalteter Knoten sowohl mit sicherheitsrelevanten Updates als auch mit anderen Arten von Updates. Er bietet mehrere große Vorteile:

- Zentralisierte Patching-Kontrolle – Mit Patch-Richtlinien können Sie wiederkehrende Patches für alle Konten in allen Regionen in Ihrer Organisation, für bestimmte Konten und Regionen oder für ein einzelnes Konto-Region-Paar einrichten.
- Flexible Patching-Vorgänge – Sie können Instances nur auf Patches hin durchsuchen und dann einen Bericht zu fehlenden Patches anzeigen oder automatisch alle fehlenden Patches installieren.
- Umfassende Compliance-Berichterstattung – Nach dem Scannen können Sie detaillierte Informationen darüber abrufen, für welche verwalteten Knoten die Patch-Konformität nicht eingehalten wurde und welche Patches fehlen.
- Plattformübergreifende Unterstützung – Patch Manager unterstützt mehrere Betriebssysteme, einschließlich verschiedener Linux-Distributionen, macOS, und Windows Server.

- Benutzerdefinierte Patch-Baselines – Mithilfe von benutzerdefinierten Patch-Baselines, die angeben, welche Patches für die Installation zugelassen sind, können Sie definieren, was Patch-Compliance für Ihr Unternehmen bedeutet.
- Integration mit anderen AWS Diensten — Patch Manager lässt sich in AWS Organizations, AWS Security Hub CSPM, und integrieren AWS CloudTrail, AWS Config um Verwaltung und Sicherheit zu verbessern.
- Deterministische Upgrades – Support für deterministische Upgrades über versionierte Repositorys für Betriebssysteme wie Amazon Linux 2023.

An wen richtet sich Patch Manager?

Patch Manager ist auf die folgenden Anwendungsfälle ausgelegt:

- IT-Administratoren, die die Patch-Compliance für ihre gesamte Flotte verwalteter Knoten sicherstellen müssen
- Betriebsmanager, die Einblick in den Status der Patch-Compliance in ihrer gesamten Infrastruktur benötigen
- Cloud-Architekten, die automatisierte Patching-Lösungen in großem Umfang implementieren möchten
- DevOps Techniker, die Patching in ihre betrieblichen Arbeitsabläufe integrieren müssen
- Unternehmen mit Bereitstellungen mit mehreren Konten oder mehreren Regionen, die ein zentrales Patch-Management benötigen
- Jeder, der für die Aufrechterhaltung der Sicherheitslage und des Betriebszustands von AWS verwalteten Knoten, Edge-Geräten, lokalen Servern und virtuellen Maschinen verantwortlich ist

Was sind die Hauptfeatures von Patch Manager?

Patch Manager bietet mehrere wichtige Features:

- Patch-Richtlinien — Konfigurieren Sie Patch-Operationen über mehrere AWS-Konten Regionen hinweg mithilfe einer einzigen Richtlinie durch Integration mit. AWS Organizations
- Benutzerdefinierte Patch-Baselines – Definieren Sie Regeln für die automatische Genehmigung von Patches innerhalb weniger Tage nach ihrer Veröffentlichung, zusammen mit genehmigten und abgelehnten Patches.

- Verschiedene Patching-Methoden – Wählen Sie je nach Ihren spezifischen Anforderungen zwischen Patch-Richtlinien, Wartungsfenstern oder On-Demand-Vorgängen „Jetzt patchen“.
- Compliance-Berichte – Generieren Sie detaillierte Berichte zum Patch-Compliance-Status, die im CSV-Format an einen Amazon-S3-Bucket gesendet werden können.
- Plattformübergreifende Unterstützung – Patchen Sie sowohl Betriebssysteme als auch Anwendungen auf Windows Server, verschiedenen Linux-Distributionen und macOS.
- Flexibilität bei der Terminplanung – Legen Sie mithilfe benutzerdefinierter CRON- oder Rate-Ausdrücke unterschiedliche Zeitpläne für das Scannen und Installieren von Patches fest.
- Lebenszyklus-Hooks – Führen Sie benutzerdefinierte Skripts vor und nach Patch-Vorgängen mithilfe von Systems-Manager-Dokumenten aus.
- Sicherheitsfokus – Der Schwerpunkt von Patch Manager liegt standardmäßig auf sicherheitsrelevanten Updates und nicht auf der Installation aller verfügbaren Patches.
- Ratenkontrolle – Konfigurieren Sie Schwellenwerte für Parallelität und Fehler bei Patch-Vorgängen, um die Auswirkungen auf den Betrieb zu minimieren.

Worin besteht Compliance in Patch Manager?

Der Maßstab dafür, was die Patch-Compliance für die verwalteten Knoten in Ihren Systems Manager Manager-Flotten ausmacht AWS, wird nicht von Betriebssystemanbietern (OS) oder von Dritten wie Sicherheitsberatungsfirmen definiert.

Stattdessen definieren Sie in einer Patch-Baseline, was Patch-Compliance für verwaltete Knoten in Ihrer Organisation oder Ihrem Konto bedeutet. Eine Patch-Baseline ist eine Konfiguration, die Regeln festlegt, nach denen Patches auf einem verwalteten Knoten installiert werden müssen. Ein verwalteter Knoten ist patch-konform, wenn er über alle Patches auf dem neuesten Stand ist, die die in der Patch-Baseline angegebenen Genehmigungskriterien erfüllen.

Beachten Sie, dass die Einhaltung einer Patch-Baseline nicht bedeutet, dass ein verwalteter Knoten unbedingt sicher ist. Konformität bedeutet, dass die in der Patch-Baseline definierten Patches, die sowohl verfügbar als auch genehmigt sind, auf dem Knoten installiert wurden. Die Gesamtsicherheit eines verwalteten Knotens wird von vielen Faktoren bestimmt, die nicht in den Anwendungsbereich von Patch Manager fallen. Weitere Informationen finden Sie unter Sicherheit.

Jede Patch-Baseline ist eine Konfiguration für einen bestimmten unterstützten Betriebssystemtyp (OS), z. B. Red Hat Enterprise Linux (RHEL), macOS oder Windows Server. Eine Patch-Baseline kann Patchregeln für alle unterstützten Versionen eines Betriebssystems definieren oder nur auf die von Ihnen angegebenen beschränkt sein, z. B. RHEL 7.8 und RHEL 9.3.

In einer Patch-Baseline könnten Sie angeben, dass alle Patches mit bestimmten Klassifizierungen und Schweregraden für die Installation zugelassen werden. Sie könnten beispielsweise alle als `Security` klassifizierten Patches einbeziehen, andere Klassifizierungen wie `Bugfix` oder `Enhancement` jedoch ausschließen. Und Sie könnten alle Patches mit einem Schweregrad von `Critical` einbeziehen und andere ausschließen, z. B. `Important` und `Moderate`.

Sie können Patches auch explizit in einer Patch-Baseline definieren, indem Sie sie IDs zu Listen bestimmter Patches hinzufügen, die genehmigt oder abgelehnt werden sollen, z. B. `KB2736693` für Windows Server oder `dbus.x86_64:1:1.12.28-1.amzn2023.0.1` für Amazon Linux 2023 (AL2023). Sie können optional eine bestimmte Anzahl von Tagen angeben, die auf das Patching gewartet wird, nachdem ein Patch verfügbar ist. Für Linux und macOS haben Sie die Möglichkeit, anstelle der in den Patch-Baseline-Regeln definierten Patches eine externe Liste von Patches aus Compliance-Gründen (eine Install Override-Liste) anzugeben.

Wenn ein Patching-Vorgang ausgeführt wird, vergleicht Patch Manager die Patches, die derzeit auf einen verwalteten Knoten angewendet werden, mit denen, die gemäß den in der Patch-Baseline oder in einer Install Override-Liste festgelegten Regeln angewendet werden sollten. Sie können auswählen, dass Patch Manager Ihnen nur einen Bericht über fehlende Patches anzeigt (ein `Scan`-Vorgang), oder Sie können auswählen, dass Patch Manager automatisch alle Patches installiert, die es auf einem verwalteten Knoten findet (ein `Scan and install`-Vorgang).

> **Note**
>
> Die Daten zur Patch-Konformität stellen eine point-in-time Momentaufnahme des letzten erfolgreichen Patch-Vorgangs dar. Jeder Konformitätsbericht enthält eine Erfassungszeit, aus der hervorgeht, wann der Konformitätsstatus berechnet wurde. Berücksichtigen Sie bei der Überprüfung der Compliance-Daten die Erfassungszeit, um festzustellen, ob der Vorgang wie erwartet ausgeführt wurde.

Patch Manager stellt vordefinierte Patch-Baselines bereit, die Sie für Ihre Patching-Vorgänge verwenden können. Diese vordefinierten Konfigurationen dienen jedoch nur als Beispiele und nicht als empfohlene bewährte Methoden. Wir empfehlen, dass Sie selbst benutzerdefinierte Patch-Baselines erstellen, um besser kontrollieren zu können, was unter Patch-Compliance für Ihre Flotte zu verstehen ist.

Weitere Informationen zu Patch-Baselines finden Sie in den folgenden Themen:

- Vordefinierte und benutzerdefinierte Patch-Baselines

- Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen
- So zeigen Sie von AWS vordefinierte Patch-Baselines an
- Arbeiten mit benutzerdefinierten Patch-Baselines
- Arbeiten mit Patch-Compliance-Berichten

Primäre Komponenten

Bevor Sie mit der Arbeit mit dem Patch Manager-Tool beginnen, sollten Sie sich mit einigen wichtigen Komponenten und Features der Patching-Vorgänge des Tools vertraut machen.

Patch-Baselines

Patch Manager verwendet Patch-Baselines, die Regeln für die automatische Genehmigung von Patches innerhalb weniger Tage nach ihrer Veröffentlichung enthalten, zusätzlich zu den optionalen Listen der genehmigten und abgelehnten Patches. Wenn ein Patching-Vorgang ausgeführt wird, vergleicht Patch Manager die Patches, die derzeit auf einen verwalteten Knoten angewendet werden, mit denen, die gemäß den in der Patch-Baseline festgelegten Regeln angewendet werden sollten. Sie können auswählen, dass Patch Manager Ihnen nur einen Bericht über fehlende Patches anzeigt (ein `Scan`-Vorgang), oder Sie können auswählen, dass Patch Manager automatisch alle Patches installiert, die es auf einem verwalteten Knoten findet (ein `Scan and install`-Vorgang).

Methoden für das Patchen von Vorgängen

Patch Manager bietet derzeit vier Methoden zum Ausführen von `Scan`- und `Scan and install`-Operationen:

- (Empfohlen) Eine in konfigurierte Patch-Richtlinie Quick Setup — Basierend auf der Integration mit können mit AWS Organizations einer einzigen Patch-Richtlinie Patch-Zeitpläne und Patch-Baselines für eine gesamte Organisation definiert werden, einschließlich mehrerer AWS-Konten und all AWS-Regionen dieser Konten, in denen alle diese Konten betrieben werden. Eine Patch-Richtlinie kann sich auch nur auf einige Organisationseinheiten (OUs) in einer Organisation beziehen. Sie können eine einzige Patch-Richtlinie verwenden, um nach verschiedenen Zeitplänen zu scannen und zu installieren. Weitere Informationen erhalten Sie unter Das Patchen für Instances in einer Organisation mithilfe einer Patch-Richtlinie für Quick Setup konfigurieren und Patch-Richtlinienkonfigurationen in Quick Setup.
- Eine Host-Management-Option, konfiguriert in Quick Setup — Host-Management-Konfigurationen werden auch durch die Integration mit unterstützt AWS Organizations, sodass ein Patch-Vorgang

für bis zu eine gesamte Organisation ausgeführt werden kann. Diese Option ist jedoch darauf beschränkt, anhand der aktuellen Standard-Patch-Baseline nach fehlenden Patches zu suchen und Ergebnisse in Compliance-Berichten bereitzustellen. Mit dieser Vorgangsmethode können keine Patches installiert werden. Weitere Informationen finden Sie unter Amazon-EC2-Host-Verwaltung mit Quick Setup einrichten.

- Ein Wartungsfenster zum Ausführen eines Patch-**`Scan`** oder einer **`Install`**-Aufgabe – Ein Wartungsfenster, das Sie im Systems-Manager-Tool mit dem Namen Maintenance Windows einrichten, kann so konfiguriert werden, dass es verschiedene Arten von Aufgaben nach einem von Ihnen definierten Zeitplan ausführt. Eine Aufgabe vom Run Command-Typ kann verwendet werden, um `Scan` oder `Scan and install` Aufgaben auf einem Satz verwalteter Knoten Ihrer Wahl auszuführen. Jede Aufgabe im Wartungsfenster kann nur auf verwaltete Knoten in einem einzigen AWS-Region Paar AWS-Konto abzielen. Weitere Informationen finden Sie unter Tutorial: Erstellen Sie ein Wartungsfenster zum Patchen über die Konsole.
- Ein „Jetzt patchen“-On-Demand-Vorgang in Patch Manager – Mit der Option Jetzt patchen können Sie Zeitplan-Einrichtungen umgehen, wenn Sie verwaltete Knoten so schnell wie möglich patchen müssen. Mit Patch now (Jetzt patchen) geben Sie an, ob der `Scan`- oder `Scan and install`-Vorgang ausgeführt werden soll und auf welchen verwalteten Knoten der Vorgang ausgeführt werden soll. Sie können sich auch dafür entscheiden, Systems-Manager-Dokumente (SSM-Dokumente) als Lebenszyklus-Hooks während des Patch-Vorgangs auszuführen. Jeder Patch-Now-Vorgang kann auf verwaltete Knoten in nur einem einzigen AWS-KontoAWS-Region Paar abzielen. Weitere Informationen finden Sie unter On-Demand-Patchen von verwalteten Knoten.

Compliance-Meldung

Nach einer `Scan`-Operation können Sie die Systems-Manager-Konsole verwenden, um Informationen darüber anzuzeigen, welche Ihrer verwalteten Knoten die Patch-Compliance nicht erfüllen und welche Patches auf jedem dieser Knoten fehlen. Sie können auch Patch-Compliance-Berichte im CSV-Format generieren, die an einen Amazon Simple Storage Service (Amazon S3)-Bucket Ihrer Wahl gesendet werden. Sie können einmalige Berichte erstellen oder Berichte nach einem regelmäßigen Zeitplan erstellen. Für einen einzelnen verwalteten Knoten enthalten Berichte Details aller Patches für den Knoten. Für einen Bericht über alle verwaltete Knoten wird nur eine Zusammenfassung der fehlenden Patches bereitgestellt. Nachdem ein Bericht generiert wurde, können Sie ein Tool wie Amazon Quick Suite verwenden, um die Daten zu importieren und zu analysieren. Weitere Informationen finden Sie unter Arbeiten mit Patch-Compliance-Berichten.

> **Note**
>
> Ein durch die Verwendung einer Patch-Richtlinie generiertes Compliance-Element hat den Ausführungstyp `PatchPolicy`. Ein Compliance-Element, das nicht in einem Patch-Richtlinienvorgang generiert wurde, hat den Ausführungstyp `Command`.

Integrationen

Patch Managerlässt sich in die folgenden anderen AWS-Services integrieren:

- AWS Identity and Access Management (IAM) — Verwenden Sie IAM, um zu steuern, welche Benutzer, Gruppen und Rollen Zugriff Patch Manager auf Operationen haben. Weitere Informationen erhalten Sie unter Funktionsweise von AWS Systems Manager mit IAM und Konfiguration von erforderlichen Instance-Berechtigungen für Systems Manager.
- AWS CloudTrail— Wird verwendet CloudTrail , um einen überprüfbaren Verlauf von Patch-Vorgängen aufzuzeichnen, die von Benutzern, Rollen oder Gruppen ausgelöst wurden. Weitere Informationen finden Sie unter Protokollierung von AWS Systems Manager-API-Aufrufen mit AWS CloudTrail.
- AWS Security Hub CSPM— Daten zur Patch-Konformität von Patch Manager können an gesendet werden. AWS Security Hub CSPM Security Hub CSPM bietet Ihnen einen umfassenden Überblick über Ihre Sicherheitswarnungen mit hoher Priorität und den Compliance-Status. Er überwacht auch den Patching-Status Ihrer Flotte. Weitere Informationen finden Sie unter Integration von Patch Manager in AWS Security Hub CSPM.
- AWS Config— Richten Sie die Aufzeichnung ein AWS Config , um die EC2 Amazon-Instanzverwaltungsdaten im Patch Manager Dashboard anzuzeigen. Weitere Informationen finden Sie unter Patch-Dashboard-Zusammenfassungen anzeigen.

Themen

- Patch-Richtlinienkonfigurationen in Quick Setup
- Patch Manager-Voraussetzungen
- So funktionieren Patch Manager-Operationen
- SSM-Befehlsdokumente zum Patchen verwalteter Knoten
- Patch-Baselines
- Verwenden von Kernel Live Patching auf von Amazon Linux 2 verwalteten Knoten

- Arbeiten mit Patch Manager-Ressourcen und Compliance mithilfe der Konsole
- Arbeiten mit Patch Manager-Ressourcen unter Verwendung der AWS CLI
- AWS Systems Manager Patch Manager Tutorials
- Fehlerbehebung für Patch Manager

Patch-Richtlinienkonfigurationen in Quick Setup

AWS empfiehlt die Verwendung von Patch-Richtlinien zur Konfiguration von Patches für Ihr Unternehmen und AWS-Konten. Patch-Richtlinien wurden in Patch Manager im Dezember 2022 eingeführt.

Eine Patch-Richtlinie ist eine Konfiguration, die Sie mit Quick Setup, einem Tool in AWS Systems Manager, einrichten. Patch-Richtlinien bieten eine umfassendere und zentralisiertere Kontrolle über Ihre Patching-Vorgänge, als dies mit früheren Methoden zum Konfigurieren von Patches möglich war. Patch-Richtlinien können mit allen von Patch Manager unterstützten Betriebssystemen verwendet werden, einschließlich unterstützter Versionen von Linux, macOS und Windows Server. Anweisungen zum Erstellen einer Patch-Richtlinie finden Sie unter Das Patchen für Instances in einer Organisation mithilfe einer Patch-Richtlinie für Quick Setup konfigurieren.

Hauptfeatures von Patch-Richtlinien

Anstatt andere Methoden zum Patchen Ihrer Knoten zu verwenden, verwenden Sie eine Patch-Richtlinie, um die Vorteile dieser Hauptfeatures zu nutzen:

- Einzelkonfiguration – Das Einrichten von Patching-Vorgängen mithilfe eines Wartungsfensters oder einer State Manager-Zuordnung kann mehrere Aufgaben in verschiedenen Teilen der Systems-Manager-Konsole erfordern. Mithilfe einer Patch-Richtlinie können alle Ihre Patching-Vorgänge in einem einzigen Assistenten eingerichtet werden.
- Unterstützung für mehrere Konten/mehrere Regionen – Mithilfe eines Wartungsfensters, einer State Manager-Zuordnung oder dem Feature Patch now (Jetzt patchen) in Patch Manager können Sie nur verwaltete Knoten in einem einzigen AWS-Konto-AWS-Region-Paar anvisieren. Wenn Sie mehrere Konten und mehrere Regionen verwenden, können Ihre Einrichtungs- und Wartungsaufgaben viel Zeit in Anspruch nehmen, da Sie Einrichtungsaufgaben in jedem Konto-Region-Paar durchführen müssen. Wenn Sie jedoch AWS Organizations verwenden, können Sie eine Patch-Richtlinie einrichten, die für alle Ihre verwalteten Knoten in allen AWS-Regionen in allen Ihren AWS-Konten gilt. Wenn Sie möchten, kann eine Patch-Richtlinie auch nur für bestimmte

Organisationseinheiten (OEs) in den von Ihnen gewählten Konten und Regionen gelten. Eine Patch-Richtlinie kann auf Wunsch auch für ein einzelnes lokales Konto gelten.

- Installationsunterstützung auf Organisationsebene – Die vorhandene Host-Management-Konfigurationsoption in Quick Setup bietetUnterstützung für einen täglichen Scan Ihrer verwalteten Knoten auf Patch-Compliance. Dieser Scan wird jedoch zu einem vorher festgelegten Zeitpunkt durchgeführt und liefert nur Patch-Compliance-Informationen. Es werden keine Patch-Installationen durchgeführt. Mithilfe einer Patch-Richtlinie können Sie unterschiedliche Zeitpläne für das Scannen und Installieren festlegen. Sie können auch die Häufigkeit und Zeit dieser Vorgänge auswählen, indem Sie benutzerdefinierte CRON- oder Rate-Ausdrücke verwenden. Sie könnten beispielsweise jeden Tag nach fehlenden Patches suchen, um regelmäßig aktualisierte Compliance-Informationen bereitzustellen. Ihr Installationsplan könnte jedoch nur einmal pro Woche sein, um unerwünschte Ausfallzeiten zu vermeiden.
- Vereinfachte Auswahl von Patch-Baselines – Patch-Richtlinien enthalten weiterhin Patch-Baselines, und es gibt keine Änderungen an der Art und Weise, wie Patch-Baselines konfiguriert werden. Wenn Sie jedoch eine Patch-Richtlinie erstellen oder aktualisieren, können Sie die AWS-verwaltete oder benutzerdefinierte Baseline, die Sie für jeden Betriebssystemtyp (OS) verwenden möchten, in einer einzigen Liste auswählen. Es ist nicht erforderlich, die Standard-Baseline für jeden Betriebssystemtyp in separaten Aufgaben anzugeben.

> **Note**
>
> Wenn Patching-Vorgänge, die auf einer Patch-Richtlinie basieren, ausgeführt werden, verwenden diese das `AWS-RunPatchBaseline`-SSM-Dokument. Weitere Informationen finden Sie unter SSM-Befehlsdokument zum Patchen: `AWS-RunPatchBaseline`.

Ähnliche Informationen

Zentrales Bereitstellen von Patching-Vorgängen über Ihre AWS-Organisation hinweg mithilfe von Systems Manager Quick Setup (Blog zu AWS-Cloud-Operationen und -Migrationen)

Weitere Unterschiede bei Patch-Richtlinien

Hier sind einige weitere Unterschiede, die bei der Verwendung von Patch-Richtlinien anstelle der vorherigen Methoden zum Konfigurieren von Patches zu beachten sind:

- Keine Patchgruppen erforderlich – Bei früheren Patching-Vorgängen konnten Sie mehrere Knoten so kennzeichnen, dass sie zu einer Patch-Gruppe gehören, und dann die Patch-Baseline angeben,

die für diese Patch-Gruppe verwendet werden soll. Wenn keine Patch-Gruppe definiert wurde, patcht Patch Manager die Instances mit der aktuellen Standard-Patch-Baseline für den OS-Typ. Durch die Verwendung von Patch-Richtlinien ist es nicht mehr erforderlich, Patch-Gruppen einzurichten und zu verwalten.

> **Note**
>
> Die Patchgruppenfunktion wird in der Konsole für Konto-Regionen-Paare nicht unterstützt, die vor der Veröffentlichung der Patch-Richtlinienunterstützung am 22. Dezember 2022 noch keine Patchgruppen verwendet haben. Die Patchgruppenfunktion ist weiterhin für Konto-Regionen-Paare verfügbar, die vor diesem Datum mit der Verwendung von Patchgruppen begonnen haben.

- Seite „Patching konfigurieren“ entfernt – Vor der Veröffentlichung von Patch-Richtlinien konnten Sie auf der Seite Configure patching (Patching konfigurieren) Standardwerte für die zu patchenden Knoten, einen Patch-Zeitplan und einen Patching-Vorgang angeben. Diese Seite wurde von Patch Manager entfernt. Diese Optionen werden jetzt in Patch-Richtlinien festgelegt.
- Keine „Patch now“ (Jetzt patchen)-Unterstützung – Die Möglichkeit, Knoten bei Bedarf zu patchen, ist immer noch auf ein einzelnes AWS-Konto-AWS-Region-Paar gleichzeitig beschränkt. Weitere Informationen finden Sie unter On-Demand-Patchen von verwalteten Knoten.
- Patch-Richtlinien und Compliance-Informationen – Wenn Ihre verwalteten Knoten gemäß einer Patching-Richtlinienkonfiguration auf Compliance gescannt werden, werden Ihnen Compliance-Daten zur Verfügung gestellt. Sie können die Daten auf die gleiche Weise wie bei anderen Methoden des Compliance-Scannens anzeigen und bearbeiten. Obwohl Sie eine Patch-Richtlinie für eine gesamte Organisation oder mehrere Organisationseinheiten einrichten können, werden Compliance-Informationen für jedes AWS-Konto-AWS-Region-Paar einzeln gemeldet. Weitere Informationen finden Sie unter Arbeiten mit Patch-Compliance-Berichten.
- Konformitätsstatus der Zuordnungen und Patch-Richtlinien – Der Patching-Status für einen verwalteten Knoten, für den eine Quick Setup-Patch-Richtlinie gilt, entspricht dem Status der State Manager-Zuordnungsausführung für diesen Knoten. Wenn der Status der Zuordnungsausführung `Compliant` lautet, wird der Patching-Status für den verwalteten Knoten ebenfalls als `Compliant` markiert. Wenn der Status der Zuordnungsausführung `Non-Compliant` lautet, wird der Patching-Status für den verwalteten Knoten ebenfalls als `Non-Compliant` markiert.

AWS-Regionen wird für Patch-Richtlinien unterstützt

Patch-Richtlinien-Konfigurationen in Quick Setup werden derzeit in den folgenden Regionen unterstützt:

- USA Ost (Ohio): (us-east-2)
- USA Ost (Nord-Virginia): (us-east-1)
- USA West (Nordkalifornien) (us-west-1)
- USA West (Oregon): (us-west-2)
- Asien-Pazifik (Mumbai): (ap-south-1)
- Asien-Pazifik (Seoul): (ap-northeast-2)
- Asien-Pazifik (Singapur): (ap-southeast-1)
- Asien-Pazifik (Sydney): (ap-southeast-2)
- Asien-Pazifik (Tokyo) (ap-northeast-1)
- Kanada (Zentral): (ca-central-1)
- Europa (Frankfurt) (eu-central-1)
- Europa (Irland) (eu-west-1)
- Europa (London) (eu-west-2)
- Europa (Paris) (eu-west-3)
- Europa (Stockholm) (eu-north-1)
- Südamerika (São Paulo) (sa-east-1)

Patch Manager-Voraussetzungen

Stellen Sie sicher, dass Sie die erforderlichen Voraussetzungen erfüllt habenPatch Manager, bevor Sie ein Tool in verwenden AWS Systems Manager.

Themen

- SSM Agent-Version
- Python-Version
- Zusätzliche Paketanforderungen
- Konnektivität zur Patch-Quelle

- S3-Endpunkt-Zugriff
- Berechtigungen zur lokalen Installation von Patches
- Unterstützte Betriebssysteme für Patch Manager

SSM Agent-Version

Version 2.0.834.0 oder höher von SSM Agent muss auf dem verwalteten Knoten ausgeführt werden, den Sie mit Patch Manager verwalten möchten.

> **Note**
>
> Wenn Systems Manager neue Tools hinzugefügt oder Aktualisierungen an den vorhandenen Tools vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Tools und Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent in Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die SSM Agent-Versionshinweise-Seite in GitHub, um Benachrichtigungen über SSM Agent-Updates zu erhalten.

Python-Version

Unterstützt Patch Manager derzeit die Python-Versionen 2.6 bis 3.12 für macOS und die meisten Linux-Betriebssysteme (OSs). Die AlmaLinuxDebian Server, und Ubuntu Server OSs erfordern eine unterstützte Version von Python 3 (3.0-3.12).

Zusätzliche Paketanforderungen

Bei DNF-basierten Betriebssystemen können die `unzip` Dienstprogramme `zstdxz`, und zum Dekomprimieren von Repository-Informationen und Patch-Dateien erforderlich sein. Zu den DNF-basierten Betriebssystemen gehören Amazon Linux 2023, Red Hat Enterprise Linux 8 und spätere Versionen, Oracle Linux 8 und höhere Versionen, Rocky Linux AlmaLinux, und CentOS 8 und spätere Versionen. Wenn Sie einen ähnlichen Fehler wie `No such file or directory: b'zstd'No such file or directory: b'unxz'`, sehen oder wenn das Patchen aufgrund fehlender Informationen fehlschlägt`unzip`, müssen Sie diese Dienstprogramme installieren. `zstdxz`, und `unzip` kann installiert werden, indem Sie den folgenden Befehl ausführen:

```
dnf install zstd xz unzip
```

Konnektivität zur Patch-Quelle

Wenn Ihre verwalteten Knoten keine direkte Verbindung zum Internet haben und Sie eine Amazon Virtual Private Cloud (Amazon VPC) mit einem VPC-Endpunkt verwenden, müssen Sie sicherstellen, dass die Knoten Zugriff auf die Quell-Patch-Verzeichnisse (Repos) haben. Auf Linux-Knoten werden Patch-Updates normalerweise von den auf dem Knoten konfigurierten Remote-Repos heruntergeladen. Daher muss der Knoten eine Verbindung zu den Repos herstellen können, damit die Patches ausgeführt werden können. Weitere Informationen finden Sie unter Wie Sicherheitspatches ausgewählt werden.

Wenn Sie einen Knoten patchen, der in einer IPv6 reinen Umgebung ausgeführt wird, stellen Sie sicher, dass der Knoten mit der Patch-Quelle verbunden ist. Sie können die Run Command Ausgabe der Patching-Ausführung überprüfen, um nach Warnungen zu unzugänglichen Repositorys zu suchen. Für DNF-basierte Betriebssysteme ist es möglich, nicht verfügbare Repositorys so zu konfigurieren, dass sie beim Patchen übersprungen werden, wenn die Option auf unter gesetzt ist. `skip_if_unavailable` `True` `/etc/dnf/dnf.conf` Zu den DNF-basierten Betriebssystemen gehören Amazon Linux 2023, Red Hat Enterprise Linux 8 und spätere Versionen, Oracle Linux 8 und höhere Versionen, Rocky Linux AlmaLinux, und CentOS 8 und spätere Versionen. Unter Amazon Linux 2023 ist die `skip_if_unavailable` Option `True` standardmäßig auf eingestellt.

CentOS Stream: Das **`EnableNonSecurity`**-Flag aktivieren

CentOS Stream-Knoten verwenden DNF als Paketmanager, das das Konzept eines Update-Hinweises verwendet. Ein Update-Hinweis ist einfach eine Sammlung von Paketen, die bestimmte Probleme beheben.

Standard-Repos von CentOS Stream werden jedoch nicht mit einem Update-Hinweis konfiguriert. Das bedeutet, dass Patch Manager Pakete auf einem Standard-CentOS Stream-Repos nicht erkennt. Um Patch Manager für die Verarbeitung von Paketen, die nicht in einem Update-Hinweis enthalten sind, zu aktivieren, müssen Sie die `EnableNonSecurity`-Markierung in den Patch-Baseline-Regeln aktivieren.

Windows Server: Stellen Sie die Konnektivität zum Windows Update Catalog oder Windows Server Update Services (WSUS) sicher

Von Windows Server verwaltete Knoten müssen eine Verbindung zum Windows Update Catalog oder Windows Server Update Services (WSUS) herstellen können. Vergewissern Sie sich, dass

Ihre Knoten über ein Internet-Gateway, ein NAT-Gateway oder eine NAT-Instance eine Verbindung zum Microsoft Update Catalog hergestellt haben. Wenn Sie WSUS verwenden, stellen Sie sicher, dass der Knoten eine Verbindung zum WSUS-Server in Ihrer Umgebung hat. Weitere Informationen finden Sie unter Problem: Der verwaltete Knoten hat keinen Zugriff auf Windows Update Catalog oder WSUS.

S3-Endpunkt-Zugriff

Unabhängig davon, ob Ihre verwalteten Knoten in einem privaten oder öffentlichen Netzwerk betrieben werden, ohne Zugriff auf die erforderlichen AWS verwalteten Amazon Simple Storage Service (Amazon S3) -Buckets schlagen Patch-Vorgänge fehl. Informationen zu den S3-Buckets, auf die Ihre verwalteten Knoten zugreifen können müssen, finden Sie unter SSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets und Verbessern Sie die Sicherheit von EC2 Instanzen mithilfe von VPC-Endpunkten für Systems Manager.

Berechtigungen zur lokalen Installation von Patches

Unter den Betriebssystemen Windows Server und Linux setzt Patch Manager zum Installieren von Patches die Benutzerkonten Administrator bzw. Root voraus.

Unter macOS unterstützt Homebrew für Brew und Brew Cask jedoch nicht die Ausführung seiner Befehle unter dem Root-Benutzerkonto. Darum fragt Patch Manager entweder als Eigentümer des Homebrew-Verzeichnisses oder als gültiger Benutzer,der zur Eigentümergruppe des Homebrew-Verzeichnisses gehört, Homebrew–Befehle ab und führt sie aus. Um Patches installieren zu können, benötigt der Besitzer des `homebrew`-Verzeichnisses daher auch rekursive Eigentümerberechtigungen für das `/usr/local`-Verzeichnis.

> **Tip**
>
> Der folgende Befehl stellt diese Berechtigung für den angegebenen Benutzer bereit:
>
> ```
> sudo chown -R $USER:admin /usr/local
> ```

Unterstützte Betriebssysteme für Patch Manager

Das Patch Manager-Tool unterstützt nicht dieselben Betriebssystemversionen, die von anderen Systems-Manager-Tools unterstützt werden. (Eine vollständige Liste der von Systems Manager unterstützten Betriebssysteme finden Sie unter Unterstützte Betriebssysteme für Systems Manager.)

Stellen Sie daher sicher, dass die verwalteten Knoten, die Sie mit Patch Manager verwenden möchten, eines der Betriebssysteme ausführen, die in der folgenden Tabelle aufgeführt sind.

> **Note**
>
> Patch Manager stützt sich auf die Patch-Repositorys, die auf einem verwalteten Knoten konfiguriert sind, z. B. Windows Update Catalog und Windows Server Update Services für Windows, um verfügbare Patches für die Installation abzurufen. Daher können Betriebssystemversionen, die am Ende ihres Lebenszyklus (EOL) sind, ist Patch Manager möglicherweise nicht in der Lage, über die neuen Updates zu berichten, wenn keine neuen Updates verfügbar sind. Dies kann daran liegen, dass vom Linux-Distributionsbetreuer, Microsoft oder Apple keine neuen Updates veröffentlicht wurden oder dass der verwaltete Knoten nicht über die richtige Lizenz für den Zugriff auf die neuen Updates verfügt.
> Wir empfehlen dringend, die Verwendung von Betriebssystemversionen zu vermeiden, die End-of-Life (EOL) erreicht haben. Betriebssystemanbieter, darunter auch, bieten AWS in der Regel keine Sicherheitspatches oder andere Updates für Versionen an, die EOL erreicht haben. Die weitere Verwendung eines EOL-Systems erhöht das Risiko, dass Upgrades, einschließlich Sicherheitsupdates, und andere Betriebsprobleme nicht durchgeführt werden können, erheblich. AWS testet die Systems Manager Manager-Funktionalität nicht auf Betriebssystemversionen, die EOL erreicht haben.
> Patch Manager meldet den Konformitätsstatus anhand der verfügbaren Patches auf dem verwalteten Knoten. Wenn auf einer Instance ein EOL-Betriebssystem ausgeführt wird und keine Updates verfügbar sind, wird Patch Manager den Knoten daher möglicherweise als konform melden, je nachdem, welche Patch-Baselines für den Patchvorgang konfiguriert wurden.

Betriebssystem	Details
Linux	• AlmaLinux 8 x, 9 x • Amazon Linux 2 Version 2.0 und alle späteren Versionen • Amazon Linux 2023 • CentOS Stream 9 • Debian Server 11.x und 12.x • Oracle Linux 7.5–8.x, 9.x

Betriebssystem	Details
	• Red Hat Enterprise Linux (RHEL) 7.x–8.x, 9.x, 10x • Rocky Linux 8.x, 9.x • SUSE Linux Enterprise Server 15.3 und höher • Ubuntu Server16,04 LTS, 18,04 LTS, 20,04 LTS, 22,04 LTS, 24,04 LTS und 25,04

Betriebssystem	Details
macOS	macOSwird nur für EC2 Amazon-Instances unterstützt. 13.0–13.7 (Ventura) 14.x (Sonoma) 15.x (Sequoia) macOS Updates Patch Manager unterstützt keine Updates oder Upgrades für macOS, z. B. von 13.1 auf 13.2. Für die Aktualisierung der Betriebss ystemversion von macOS empfehlen wir, die in Apple integrierten Mechanismen zu verwenden . Weitere Informationen finden Sie auf der Website mit Entwicklerdokumentation von Apple unter Device Management. Unterstützung für Homebrew Patch Manager erfordert, dass Homebrew, das Open-Source-Softwarepaketverwaltungs system, an einem der folgenden Standardi nstallationsverzeichnisse installiert ist: • `/usr/local/bin` • `/opt/homebrew/bin` Patch-Vorgänge, die Patch Manager verwenden, funktionieren nicht richtig, wenn Homebrew nicht installiert ist. Regionsunterstützung

Betriebssystem	Details
	macOSwird nicht in allen unterstützt AWS-Regionen. Weitere Informationen zum EC2 Amazon-Support für macOS finden Sie unter Amazon EC2 Mac-Instances im EC2 Amazon-Benutzerhandbuch. macOS-Edge-Geräte SSM Agentfür AWS IoT Greengrass Core-Geräte wird auf nicht unterstütztmacOS. Sie können Patch Manager nicht verwenden, um macOS-Edge-Geräte zu patchen.
Windows	Windows Server 2012 bis Windows Server 2025, einschließlich R2-Versionen. **Note** SSM Agentfür AWS IoT Greengrass Kerngeräte wird unter Windows 10 nicht unterstützt. Sie können Patch Manager nicht verwenden, um Windows-10-Edge-Geräte zu patchen. Windows Server 2012 und 2012 R2 Windows Server 2012 und 2012 R2 haben am 10. Oktober 2023 das Ende der Unterstützung erreicht. Für die Verwendung Patch Manager mit diesen Versionen empfehlen wir, auch Extended Security Updates (ESUs) von Microsoft zu verwenden. Weitere Informationen finden Sie auf der Microsoft-Website unter Windows Server 2012 und 2012 R2 erreichen das Ende des Supports.

So funktionieren Patch Manager-Operationen

Dieser Abschnitt enthält technische Details, die erklären, wie Patch Manager, ein Tool in AWS Systems Manager, bestimmt, welche Patches installiert werden und wie er sie auf dem jeweiligen unterstützten Betriebssystem installiert. Für Linux-Betriebssysteme enthält er auch Informationen zur Angabe einer Quell-Repository, in einer benutzerdefinierten Patch-Baseline, für andere Patches als diejenigen, die standardmäßig auf einem verwalteten Knoten konfiguriert sind. Dieser Abschnitt bietet außerdem Informationen darüber, wie Patch-Baseline-Regeln auf verschiedenen Verteilungen des Linux-Betriebssystems funktionieren.

> **Note**
>
> Die Informationen in den folgenden Themen gelten unabhängig davon, welche Methode oder Art der Konfiguration Sie für Ihre Patching-Vorgänge verwenden:
>
> - Eine in Quick Setup konfigurierte Patch-Richtlinie
> - Eine in Quick Setup konfigurierte Host-Management-Option
> - Ein Wartungsfenster zum Ausführen eines Patch-`Scan` oder einer `Install`-Aufgabe
> - Ein On-Demand-Jetzt patchen-Vorgang

Themen

- So werden Veröffentlichungs- und Aktualisierungsdaten von Paketen berechnet
- Wie Sicherheitspatches ausgewählt werden
- So geben Sie ein alternatives Patch-Quell-Repository an (Linux)
- Wie Patches installiert werden
- Funktionsweise von Patch-Baseline-Regeln auf Linux-basierten Systemen
- Unterschiede beim Patchvorgang zwischen Linux und Windows Server

So werden Veröffentlichungs- und Aktualisierungsdaten von Paketen berechnet

> **Important**
>
> Die Informationen auf dieser Seite beziehen sich auf die Betriebssysteme Amazon Linux 2 und Amazon Linux 2023 (OSs) für Amazon Elastic Compute Cloud (Amazon EC2) -Instances. Die Pakete für diese Betriebssystemtypen werden von Amazon Web Services erstellt und

verwaltet. Wie die Hersteller anderer Betriebssysteme ihre Pakete und Repositorys verwalten, wirkt sich darauf aus, wie ihre Veröffentlichungs- und Aktualisierungsdaten berechnet werden. OSs Neben Amazon Linux 2 und Amazon Linux 2023 finden Sie in der Dokumentation des Herstellers Informationen darüberRed Hat Enterprise Linux, wie ihre Pakete aktualisiert und gewartet werden.

In den Einstellungen für benutzerdefinierte Patch-Baselines, die Sie für die meisten Betriebssystemtypen erstellen, können Sie angeben, dass Patches nach einer bestimmten Anzahl von Tagen automatisch für die Installation genehmigt werden. AWS bietet mehrere vordefinierte Patch-Baselines, die automatische Genehmigungsdaten von 7 Tagen enthalten.

Eine Verzögerung der automatischen Genehmigung ist die Anzahl an Tagen, die gewartet werde soll, nachdem die Patch veröffentlicht wurde, bevor der Patch automatisch genehmigt wird. Beispielsweise erstellen Sie eine Regel mit der `CriticalUpdates`-Klassifizierung und konfigurieren sie für eine Verzögerung der automatischen Genehmigung um sieben Tage. Infolgedessen wird ein neuer kritischer Patch mit einem Veröffentlichungsdatum oder dem letzten Aktualisierungsdatum vom 7. Juli automatisch am 14. Juli genehmigt.

Um unerwartete Ergebnisse mit Verzögerungen bei der automatischen Genehmigung auf Amazon Linux 2 und Amazon Linux 2023 zu vermeiden, ist es wichtig zu wissen, wie die Veröffentlichungs- und Aktualisierungsdaten berechnet werden.

> **Note**
>
> Wenn ein Amazon Linux 2- oder Amazon Linux 2023-Repository keine Informationen zum Veröffentlichungsdatum von Paketen bereitstellt, verwendet Patch Manager die Erstellungszeit des Pakets als Datum zur automatischen Genehmigung der Datumsangaben. Wenn die Erstellungszeit des Pakets nicht bestimmt werden kann, verwendet Patch Manager als Standarddatum den 1. Januar 1970. Dies führt dazu, dass alle Angaben zum Datum der automatischen Genehmigung in Patch-Baselines von Patch Manager umgangen werden, die so konfiguriert sind, dass Patches für jedes Datum nach dem 1. Januar 1970 genehmigt werden.

In den meisten Fällen wird die Wartezeit für die automatische Genehmigung vor der Installation von Patches aus einem `Updated Date`-Wert in `updateinfo.xml` und nicht aus einem `Release Date`-Wert berechnet. Im Folgenden finden Sie wichtige Details zu diesen Datumsberechnungen:

- Das `Release Date` ist das Datum, an dem ein Hinweis veröffentlicht wird. Dies bedeutet nicht, dass das Paket bereits in den zugehörigen Repositorys verfügbar ist.
- Das `Update Date` ist das letzte Datum, an dem der Hinweis aktualisiert wurde. Eine Aktualisierung eines Hinweises kann etwas so Kleines wie eine Text- oder Beschreibungsaktualisierung darstellen. Dies bedeutet nicht, dass das Paket ab diesem Datum veröffentlicht wurde oder notwendigerweise in den zugehörigen Repositorys verfügbar ist.

  Dies bedeutet, dass ein Paket einen `Update Date`-Wert vom 7. Juli haben kann, aber erst (zum Beispiel) am 13. Juli für die Installation verfügbar sein kann. Angenommen, in diesem Fall wird am 14. Juli in einem `Install`-Vorgang eine Patch-Baseline ausgeführt, die eine 7-tägige automatische Genehmigungsverzögerung angibt. Da der `Update Date`-Wert sieben Tage vor dem Ausführungsdatum liegt, werden die Patches und Updates im Paket am 14. Juli installiert. Die Installation erfolgt, obwohl erst ein Tag vergangen ist, seit das Paket für die eigentliche Installation verfügbar ist.
- Ein Paket, das Betriebssystem- oder Anwendungs-Patches enthält, kann nach der ersten Veröffentlichung mehrmals aktualisiert werden.
- Ein Paket kann in den AWS verwalteten Repositorys veröffentlicht, dann aber zurückgesetzt werden, wenn später Probleme damit entdeckt werden.

Bei einigen Patch-Vorgängen sind diese Faktoren möglicherweise nicht wichtig. Wenn beispielsweise eine Patch-Baseline so konfiguriert ist, dass ein Patch mit den Schweregraden `Low` und `Medium` und der Klassifizierung `Recommended` installiert wird, kann jede Verzögerung der automatischen Genehmigung nur geringe Auswirkungen auf Ihren Betrieb haben.

In Fällen, in denen das Timing kritischer Patches oder Patches mit hohem Schweregrad wichtiger ist, sollten Sie möglicherweise mehr Kontrolle darüber haben, wann Patches installiert werden. Die empfohlene Methode hierfür ist die Verwendung alternativer Patch-Quell-Repositorys anstelle der Standard-Repositorys für Patch-Vorgänge auf einem verwalteten Knoten.

Sie können beim Erstellen einer benutzerdefinierten Patch-Baseline alternative Patch-Quell-Repositorys angeben. Für jede benutzerdefinierte Patch-Baseline können Sie Patch-Quellkonfigurationen für bis zu 20 Versionen eines unterstützten Linux-Betriebssystems angeben. Weitere Informationen finden Sie unter So geben Sie ein alternatives Patch-Quell-Repository an (Linux).

Wie Sicherheitspatches ausgewählt werden

Das Hauptaugenmerk vonPatch Manager, einem Tool in AWS Systems Manager, liegt auf der Installation sicherheitsrelevanter Betriebssystemupdates auf verwalteten Knoten. Standardmäßig installiert Patch Manager daher nicht alle verfügbaren Patches, sondern eher eine kleinere Reihe von Patches mit Schwerpunkt auf der Sicherheit.

Standardmäßig ersetzt Patch Manager kein Paket, das in einem Paket-Repository mit Ersatzpaketen als veraltet markiert wurde mit Ersatzpaketen mit anderen Namen, es sei denn, dieser Ersatz ist für die Installation eines anderen Paket-Updates erforderlich. Stattdessen meldet und installiert Patch Manager bei Befehlen, die ein Paket aktualisieren, nur fehlende Updates für das Paket, das installiert wird, aber veraltet ist. Dies liegt daran, dass das Ersetzen eines veralteten Pakets normalerweise die Deinstallation eines vorhandenen Pakets und die Installation des Ersatzpakets erfordert. Durch das Ersetzen des veralteten Pakets könnten grundlegende Änderungen oder zusätzliche Funktionen eingeführt werden, die Sie nicht beabsichtigt hatten.

Dieses Verhalten entspricht dem `update-minimal`-Befehl von YUM und DNF, der sich eher auf Sicherheitsupdates als auf Feature-Updates konzentriert. Weitere Informationen finden Sie unter the section called “Wie Patches installiert werden”.

> **Note**
>
> Wenn Sie den `ApproveUntilDate` Parameter oder `ApproveAfterDays` Parameter in einer Patch-Baseline-Regel verwenden, werden die Veröffentlichungsdaten von Patches anhand der koordinierten Weltzeit (UTC) Patch Manager ausgewertet.
> Wenn Sie beispielsweise ein Datum angeben`ApproveUntilDate`, werden Patches`2025-11-16`, die zwischen `2025-11-16T00:00:00Z` und veröffentlicht wurden`2025-11-16T23:59:59Z`, genehmigt.
> Beachten Sie, dass die Veröffentlichungsdaten von Patches, die von systemeigenen Paketmanagern auf Ihren verwalteten Knoten angezeigt werden, je nach der lokalen Zeitzone Ihres Systems unterschiedliche Zeiten anzeigen können. Für Genehmigungsberechnungen wird jedoch Patch Manager immer die UTC-Datum/Uhrzeit verwendet. Dadurch wird die Konsistenz mit den auf offiziellen Websites mit Sicherheitshinweisen veröffentlichten Patch-Veröffentlichungsdaten gewährleistet.

Für Linux-basierte Betriebssysteme, die einen Schweregrad für Patches melden, verwendet Patch Manager den vom Softwareherausgeber gemeldeten Schweregrad für den Update-Hinweis oder

den einzelnen Patch. Patch Manager leitet keinen Schweregrad aus Drittquellen wie dem Common Vulnerability Scoring System (CVSS) oder aus Metriken ab, die von der National Vulnerability Database (NVD) veröffentlicht werden.

> **Note**
>
> Auf allen Linux-basierten Systemen, die von Patch Manager unterstützt werden, können Sie ein anderes für den verwalteten Knoten konfiguriertes Quell-Repository auswählen, typischerweise zur Installation von nicht-sicherheitsrelevanten Updates. Weitere Informationen finden Sie unter So geben Sie ein alternatives Patch-Quell-Repository an (Linux).

Der Rest dieses Abschnitts erläutert, wie Patch Manager-Sicherheits-Patches für die verschiedenen unterstützten Betriebssysteme auswählt.

Amazon Linux 2 and Amazon Linux 2023

Vorkonfigurierte Repositorys werden auf Amazon Linux 2 anders behandelt als auf Amazon Linux 2023.

Auf Amazon Linux 2 verwendet der Patch-Baseline-Server des Systems Managers vorkonfigurierte Repositorys auf dem verwalteten Knoten. Es gibt in der Regel zwei vorkonfigurierte Repositorys (Repos) auf einem Knoten:

Auf Amazon Linux 2

- Repo-ID: `amzn2-core/2/`*`architecture`*

  Repo-Name: `Amazon Linux 2 core repository`
- Repo-ID: `amzn2extra-docker/2/`*`architecture`*

  Repo-Name: `Amazon Extras repo for docker`

> **Note**
>
> *`architecture`*kann x86_64 oder (für Graviton-Prozessoren) aarch64 sein.

Wenn Sie eine Amazon Linux 2023 (AL2023) -Instance erstellen, enthält sie die Updates, die in der Version AL2 023 verfügbar waren, und die von AMI Ihnen ausgewählte Version. Ihre AL2 023-Instance erhält beim Start nicht automatisch zusätzliche kritische und wichtige Sicherheitsupdates. Stattdessen können Sie mit der für AL2 023 unterstützten Funktion für deterministische Upgrades über versionierte Repositorys, die standardmäßig aktiviert ist, Updates nach einem Zeitplan anwenden, der Ihren spezifischen Anforderungen entspricht. Weitere Informationen finden Sie unter Deterministische Upgrades durch versionierte Repositories im Benutzerhandbuch von Amazon Linux 2023.

In Version AL2 023 sieht das vorkonfigurierte Repository wie folgt aus:

- Repo-ID: `amazonlinux`

  Repository-Name: Amazon-Linux-2023-Repository

Bei Amazon Linux 2023 (Vorschauversion) sind die vorkonfigurierten Repositories an gesperrte Versionen von Paket-Updates gebunden. Wenn new Amazon Machine Images (AMIs) für AL2 023 veröffentlicht wird, sind sie für eine bestimmte Version gesperrt. Bei Patch-Aktualisierungen ruft Patch Manager die letzte gesperrte Version des Patch-Aktualisierungs-Repositorys ab und aktualisiert dann die Pakete auf dem verwalteten Knoten basierend auf dem Inhalt dieser gesperrten Version.

Paket-Manager

Amazon Linux 2 verwaltete Knoten verwenden YUM als Paket-Manager. Amazon Linux 2023 verwendet DNF als Paket-Manager.

Beide Paket-Manager verwenden das Konzept einer Aktualisierungsbenachrichtigung in Form einer Datei mit dem Namen `updateinfo.xml`. Ein Update-Hinweis ist einfach eine Sammlung von Paketen, die bestimmte Probleme beheben. Alle Pakete in einem Update-Hinweis werden von Patch Manager als sicherheitsrelevant betrachtet. Einzelnen Paketen werden keine Klassifizierungen oder Schweregrade zugewiesen. Daher weist Patch Manager die Attribute eines Update-Hinweises den jeweiligen Paketen zu.

> **Note**
>
> Wenn Sie das Kontrollkästchen Funktionsupdates einschließen auf der Seite Patch-Baseline erstellen aktivieren, können Pakete, die nicht in einer `updateinfo.xml`-Datei klassifiziert sind (oder Pakete, die eine Datei ohne korrekt formatierte Klassifizierung,

Schweregrad und Datenwerte), in die vorgefilterte Patchliste aufgenommen werden. Damit ein Patch jedoch angewendet werden kann, muss er dennoch die benutzerspezifischen Patch-Baseline-Regeln erfüllen.
Weitere Informationen zur Option Nicht sicherheitsrelevante Updates einbeziehen finden Sie unter Wie Patches installiert werden und Funktionsweise von Patch-Baseline-Regeln auf Linux-basierten Systemen.

CentOS Stream

Der Patch-Baseline-Service des Systems Managers auf CentOS Stream verwendet vorkonfigurierte Repositorys (Repos) auf dem verwalteten Knoten. Die folgende Liste enthält Beispiele für ein fiktives CentOS 9.2 Amazon Machine Image (AMI):

- Repo-ID: `example-centos-9.2-base`

  Repo-Name: `Example CentOS-9.2 - Base`
- Repo-ID: `example-centos-9.2-extras`

  Repo-Name: `Example CentOS-9.2 - Extras`
- Repo-ID: `example-centos-9.2-updates`

  Repo-Name: `Example CentOS-9.2 - Updates`
- Repo-ID: `example-centos-9.x-examplerepo`

  Repo-Name: `Example CentOS-9.x – Example Repo Packages`

> **Note**
>
> Alle Updates werden von den auf dem verwalteten Knoten konfigurierten Remote-Repos heruntergeladen. Daher muss der Knoten über einen ausgehenden Zugang zum Internet verfügen, um eine Verbindung zu den Repos herzustellen, damit das Patching durchgeführt werden kann.

CentOS Stream-Knoten verwenden DNF als Paket-Manager. Der Paket-Manager verwenden das Konzept eines Update-Hinweises. Ein Update-Hinweis ist einfach eine Sammlung von Paketen, die bestimmte Probleme beheben.

Standard-Repos von CentOS Stream werden jedoch nicht mit einem Update-Hinweis konfiguriert. Das bedeutet, dass Patch Manager Pakete auf einem Standard-CentOS Stream-Repos nicht erkennt. Um Patch Manager für die Verarbeitung von Paketen, die nicht in einem Update-Hinweis enthalten sind, zu aktivieren, müssen Sie die `EnableNonSecurity`-Markierung in den Patch-Baseline-Regeln aktivieren.

> **Note**
>
> CentOS Stream-Update-Hinweise werden unterstützt. Repos mit Update-Hinweisen können nach dem Start heruntergeladen werden.

Debian Server

Der Systems Manager-Patch-Baseline-Service auf Debian Server verwendet vorkonfigurierte Repositorys (Repos) auf der Instance. Diese vorkonfigurierten Repos werden verwendet, um eine aktualisierte Liste der verfügbaren Paket-Upgrades abzurufen. Dazu führt Systems Manager das Äquivalent eines `sudo apt-get update`-Befehls durch.

Pakete werden dann aus `debian-security` *`codename`*-Repos gefiltert. Das bedeutet, dass für jede Version von Debian Server, Patch Manager nur Upgrades identifiziert, die Teil des zugehörigen Repositorys für diese Version sind, und zwar wie folgt:

- Debian Server11: `debian-security bullseye`
- Debian Server12: `debian-security bookworm`

Oracle Linux

Der Patch-Baseline-Service des Systems Managers auf Oracle Linux verwendet vorkonfigurierte Repositorys (Repos) auf dem verwalteten Knoten. Es gibt in der Regel zwei vorkonfigurierte Repositorys (Repos) auf einem Knoten.

Oracle Linux 7:

- Repo-ID: `ol7_UEKR5/x86_64`

  Repo-Name: `Latest Unbreakable Enterprise Kernel Release 5 for Oracle Linux 7Server (x86_64)`
- Repo-ID: `ol7_latest/x86_64`

Repo-Name: `Oracle Linux 7Server Latest (x86_64)`

Oracle Linux 8:

- Repo-ID: `ol8_baseos_latest`

  Repo-Name: `Oracle Linux 8 BaseOS Latest (x86_64)`
- Repo-ID: `ol8_appstream`

  Repo-Name: `Oracle Linux 8 Application Stream (x86_64)`
- Repo-ID: `ol8_UEKR6`

  Repo-Name: `Latest Unbreakable Enterprise Kernel Release 6 for Oracle Linux 8 (x86_64)`

Oracle Linux 9:

- Repo-ID: `ol9_baseos_latest`

  Repo-Name: `Oracle Linux 9 BaseOS Latest (x86_64)`
- Repo-ID: `ol9_appstream`

  Repo-Name: `Oracle Linux 9 Application Stream Packages(x86_64)`
- Repo-ID: `ol9_UEKR7`

  Repo-Name: `Oracle Linux UEK Release 7 (x86_64)`

> **Note**
>
> Alle Updates werden von den auf dem verwalteten Knoten konfigurierten Remote-Repos heruntergeladen. Daher muss der Knoten über einen ausgehenden Zugang zum Internet verfügen, um eine Verbindung zu den Repos herzustellen, damit das Patching durchgeführt werden kann.

Von Oracle Linux verwaltete Knoten verwenden Yum als Paketmanager und Yum verwendet das Konzept eines Update-Hinweises in Form einer Datei namens `updateinfo.xml`. Ein Update-

Hinweis ist einfach eine Sammlung von Paketen, die bestimmte Probleme beheben. Einzelnen Paketen werden keine Klassifizierungen oder Schweregrade zugewiesen. Aus diesem Grund weist Patch Manager die Attribute eines Update-Hinweises den jeweiligen Paketen zu und installiert Pakete basierend auf den Klassifizierungsfiltern, die in der Patch-Baseline angegeben sind.

> **Note**
>
> Wenn Sie das Kontrollkästchen Funktionsupdates einschließen auf der Seite Patch-Baseline erstellen aktivieren, können Pakete, die nicht in einer `updateinfo.xml`-Datei klassifiziert sind (oder Pakete, die eine Datei ohne korrekt formatierte Klassifizierung, Schweregrad und Datenwerte), in die vorgefilterte Patchliste aufgenommen werden. Damit ein Patch jedoch angewendet werden kann, muss er dennoch die benutzerspezifischen Patch-Baseline-Regeln erfüllen.

AlmaLinux, RHEL, and Rocky Linux

Ein AlmaLinuxRed Hat Enterprise Linux, und Rocky Linux der Systems Manager Patch Baseline Service verwendet vorkonfigurierte Repositorys (Repos) auf dem verwalteten Knoten. Es gibt in der Regel drei vorkonfigurierte Repositorys (Repos) auf einem Knoten.

Alle Updates werden von den auf dem verwalteten Knoten konfigurierten Remote-Repos heruntergeladen. Daher muss der Knoten über einen ausgehenden Zugang zum Internet verfügen, um eine Verbindung zu den Repos herzustellen, damit das Patching durchgeführt werden kann.

> **Note**
>
> Wenn Sie das Kontrollkästchen Funktionsupdates einschließen auf der Seite Patch-Baseline erstellen aktivieren, können Pakete, die nicht in einer `updateinfo.xml`-Datei klassifiziert sind (oder Pakete, die eine Datei ohne korrekt formatierte Klassifizierung, Schweregrad und Datenwerte), in die vorgefilterte Patchliste aufgenommen werden. Damit ein Patch jedoch angewendet werden kann, muss er dennoch die benutzerspezifischen Patch-Baseline-Regeln erfüllen.

Red Hat Enterprise Linux7 verwaltete Knoten verwenden Yum als Paketmanager. AlmaLinux, Red Hat Enterprise Linux 8, und Rocky Linux verwaltete Knoten verwenden DNF als Paketmanager.

Beide Paketmanager verwenden das Konzept eines Update-Hinweises als Datei namens `updateinfo.xml`. Ein Update-Hinweis ist einfach eine Sammlung von Paketen, die bestimmte Probleme beheben. Einzelnen Paketen werden keine Klassifizierungen oder Schweregrade zugewiesen. Aus diesem Grund weist Patch Manager die Attribute eines Update-Hinweises den jeweiligen Paketen zu und installiert Pakete basierend auf den Klassifizierungsfiltern, die in der Patch-Baseline angegeben sind.

RHEL 7

> **Note**
>
> Das folgende Repo ist mit IDs RHUI 2 verknüpft. RHUI 3 wurde im Dezember 2019 veröffentlicht und führte ein anderes Benennungsschema für das Yum-Repository ein. IDs Abhängig von dem RHEL-7-AMI, aus dem Sie Ihre verwalteten Knoten erstellen, müssen Sie möglicherweise Ihre Befehle aktualisieren. Weitere Informationen finden Sie unter Repository IDs for RHEL 7 in Have AWS Changed im Red Hat Customer Portal.

- Repo-ID: `rhui-REGION-client-config-server-7/x86_64`

  Repo-Name: `Red Hat Update Infrastructure 2.0 Client Configuration Server 7`
- Repo-ID: `rhui-REGION-rhel-server-releases/7Server/x86_64`

  Repo-Name: `Red Hat Enterprise Linux Server 7 (RPMs)`
- Repo-ID: `rhui-REGION-rhel-server-rh-common/7Server/x86_64`

  Repo-Name: `Red Hat Enterprise Linux Server 7 RH Common (RPMs)`

AlmaLinux, 8, RHEL 8 und Rocky Linux 8

- Repo-ID: `rhel-8-appstream-rhui-rpms`

  Repo-Name: `Red Hat Enterprise Linux 8 for x86_64 - AppStream from RHUI (RPMs)`
- Repo-ID: `rhel-8-baseos-rhui-rpms`

  Repo-Name: `Red Hat Enterprise Linux 8 for x86_64 - BaseOS from RHUI (RPMs)`

- Repo-ID: `rhui-client-config-server-8`

  Repo-Name: `Red Hat Update Infrastructure 3 Client Configuration Server 8`

AlmaLinux 9, RHEL 9 und Rocky Linux 9

- Repo-ID: `rhel-9-appstream-rhui-rpms`

  Repo-Name: `Red Hat Enterprise Linux 9 for x86_64 - AppStream from RHUI (RPMs)`

- Repo-ID: `rhel-9-baseos-rhui-rpms`

  Repo-Name: `Red Hat Enterprise Linux 9 for x86_64 - BaseOS from RHUI (RPMs)`

- Repo-ID: `rhui-client-config-server-9`

  Repo-Name: `Red Hat Enterprise Linux 9 Client Configuration`

SLES

Auf von SUSE Linux Enterprise Server (SLES) verwalteten Knoten erhält die ZYPP-Bibliothek die Liste der verfügbaren Patches (eine Sammlung von Paketen) von den folgenden Standorten:

- Liste der Repositorys: `etc/zypp/repos.d/*`
- Paketinformationen: `/var/cache/zypp/raw/*`

Von SLES verwaltete Knoten verwenden Zypper als Paketmanager und Zypper verwendet das Konzept eines Patches. Ein Patch ist einfach eine Sammlung von Paketen, die ein bestimmtes Problem beheben. Patch Manager behandelt alle Pakete, auf die in einem Patch verwiesen wird, als sicherheitsbezogen. Da einzelnen Paketen weder Klassifizierungen noch Schweregrade zugewiesen werden, weist Patch Manager den Paketen die Attribute des Patches zu, dem sie angehören.

Ubuntu Server

Der Patch-Baseline-Service des Systems Managers auf Ubuntu Server verwendet vorkonfigurierte Repositorys (Repos) auf dem verwalteten Knoten. Diese vorkonfigurierten Repos werden verwendet, um eine aktualisierte Liste der verfügbaren Paket-Upgrades abzurufen. Dazu führt Systems Manager das Äquivalent eines `sudo apt-get update`-Befehls durch.

Pakete werden dann aus *codename*`-security`-Repos gefiltert, wobei der Codename für die Release-Version eindeutig ist, z. B. `trusty` für Ubuntu Server 14. Patch Manager identifiziert nur Upgrades, die Teil dieser Repos sind:

-
- Ubuntu Server 16.04 LTS: `xenial-security`
- Ubuntu Server 18.04 LTS: `bionic-security`
- Ubuntu Server 20.04 LTS: `focal-security`
-
- Ubuntu Server 22.04 LTS (`jammy-security`)
-
-
- Ubuntu Server 24.04 LTS (`noble-security`)
-
- Ubuntu Server 25.04 (`plucky-security`)

Windows Server

Unter Microsoft Windows-Betriebssystemen ruft Patch Manager eine Liste der verfügbaren, von Microsoft über Microsoft Update veröffentlichten und automatisch auf Windows Server Update Services (WSUS) verfügbaren Updates ab.

> **Note**
>
> Patch Manager stellt nur dann Patches für Windows Server-Betriebssystemversionen bereit, wenn diese für Patch Manager unterstützt werden. Beispiel: Patch Manager kann nicht zum Patchen von Windows RT verwendet werden.

Patch Manager überwacht kontinuierlich neue Updates in allen AWS-Region. Die Liste der verfügbaren Updates wird in jeder Region mindestens einmal pro Tag aktualisiert. Wenn die Patch-Informationen von Microsoft verarbeitet werden, entfernt Patch Manager Updates, die durch spätere Updates ersetzt wurden, aus der Patch-Liste. Daher werden nur die neuesten Updates angezeigt und zur Installation zur Verfügung gestellt. Beispiel: Wenn `KB4012214` `KB3135456` ersetzt, steht nur `KB4012214` als Update in Patch Manager zur Verfügung.

Ebenso kann Patch Manager nur Patches installieren, die zum Zeitpunkt des Patchvorgangs auf dem verwalteten Knoten verfügbar sind. Standardmäßig entfernen Windows Server 2019 und Windows Server 2022 Updates, die durch spätere Updates ersetzt werden. Wenn Sie den `ApproveUntilDate`-Parameter in einer Windows Server-Patch-Baseline verwenden, das im `ApproveUntilDate`-Parameter ausgewählte Datum jedoch vor dem Datum des letzten Patches liegt, tritt das folgende Szenario ein:

- Der ersetzte Patch wurde vom Knoten entfernt und kann daher nicht mit Patch Manager installiert werden.
- Der neueste Ersatz-Patch ist auf dem Knoten vorhanden, wurde aber zum angegebenen Datum noch nicht für die `ApproveUntilDate`-Installation freigegeben.

Das bedeutet, dass der verwaltete Knoten die Anforderungen des Systems-Manager-Betriebs erfüllt, auch wenn ein kritischer Patch aus dem Vormonat möglicherweise nicht installiert wurde. Das gleiche Szenario kann bei Verwendung des `ApproveAfterDays`-Parameters auftreten. Aufgrund des Verhaltens von Microsoft bei abgelösten Patches ist es möglich, eine Zahl (im Allgemeinen größer als 30 Tage) festzulegen, sodass Patches für Windows Server nie installiert werden, wenn der neueste verfügbare Patch von Microsoft veröffentlicht wird, bevor die Anzahl der Tage in `ApproveAfterDays` verstrichen ist. Beachten Sie, dass dieses Systemverhalten nicht zutrifft, wenn Sie Ihre Einstellungen für das Windows-Gruppenrichtlinienobjekt (GPO) so geändert haben, dass der abgelöste Patch auf Ihren verwalteten Knoten verfügbar ist.

> **Note**
>
> In einigen Fällen veröffentlicht Microsoft Patches für Anwendungen, die kein aktualisiertes Datum und keine aktualisierte Uhrzeit angeben. In diesen Fällen wird ein aktualisiertes Datum und eine Uhrzeit von `01/01/1970` standardmäßig angegeben.

So geben Sie ein alternatives Patch-Quell-Repository an (Linux)

Wenn Sie die auf einem verwalteten Knoten konfigurierten Standardrepositorys für Patch-Operationen verwenden Patch Manager AWS Systems Manager, sucht ein Tool in nach sicherheitsrelevanten Patches oder installiert diese. Dies ist das Standardverhalten für Patch Manager. Ausführliche Informationen darüber, wie Patch Manager Sicherheits-Patches auswählt und installiert, finden Sie unter Wie Sicherheitspatches ausgewählt werden.

Sie können auf Linux-Systemen jedoch auch mithilfe von Patch Manager Patches installieren, die nicht auf Sicherheit bezogen sind oder die sich in einem anderen als dem Standard-Quell-Repository befinden, das in dem verwalteten Knoten konfiguriert ist. Sie können beim Erstellen einer benutzerdefinierten Patch-Baseline alternative Patch-Quell-Repositorys angeben. Für jede benutzerdefinierte Patch-Baseline können Sie Patch-Quellkonfigurationen für bis zu 20 Versionen eines unterstützten Linux-Betriebssystems angeben.

Angenommen, Ihre Ubuntu Server-Flotte beinhaltet Ubuntu Server 25.04-verwaltete Knoten. In diesem Fall können Sie alternative Repositorys für jede Version in derselben benutzerdefinierten Patch-Baseline angeben. Geben Sie für jede Version einen Namen, die Version und den Typ des Betriebssystems (Produkt) und eine Repository-Konfiguration an. Sie können auch ein einziges alternatives Quell-Repository angeben, das für alle Versionen eines unterstützten Betriebssystems gilt.

> **Note**
>
> Wenn Sie eine benutzerdefinierte Patch-Baseline ausführen, die alternative Patch-Repositorys für einen verwalteten Knoten angeben, werden diese Repositorys dadurch nicht zum neuen Standard-Repository auf dem Betriebssystem. Nach Abschluss der Patching-Operation bleiben die zuvor definierten Standard-Repositorys für das Betriebssystem des Knoten als Standard erhalten.

Eine Liste mit Beispielszenarien zur Verwendung dieser Option finden Sie weiter Anwendungsbeispiele für alternative Patch-Quell-Repositorys unten in diesem Thema.

Weitere Informationen zu Standard- und benutzerdefinierten Patch-Baselines finden Sie unter Vordefinierte und benutzerdefinierte Patch-Baselines.

Beispiel: Verwenden der Konsole

Um alternative Patch-Quell-Repositorys anzugeben, wenn Sie in der Systems Manager-Konsole arbeiten, verwenden Sie den Abschnitt Patch sources auf der Seite Create patch baseline. Weitere Informationen zur Verwendung der Optionen in Patch sources (Patch-Quellen) finden Sie unter So erstellen Sie eine benutzerdefinierte Patch-Baseline für Linux.

Beispiel: Verwendung des AWS CLI

Ein Beispiel für die Verwendung der Option `--sources` mit der AWS Command Line Interface (AWS CLI) finden Sie in Erstellen einer Patch-Baseline mit benutzerdefinierten Repositorys für verschiedene Betriebssystemversionen.

Themen

- Wichtige Überlegungen für alternative Repositorys
- Anwendungsbeispiele für alternative Patch-Quell-Repositorys

Wichtige Überlegungen für alternative Repositorys

Beachten Sie die folgenden Punkte, wenn Sie planen, in Ihrer Patching-Strategie alternative Patch-Repositorys zu verwenden.

Erzwingen Sie Überprüfungen von Repo-Updates (YUM und DNF)

Eine Standardkonfiguration für einen Paketmanager auf einer Linux-Distribution kann so eingestellt sein, dass ein nicht erreichbares Paket-Repository übersprungen wird, wenn keine Verbindung zum Repository hergestellt werden kann. Um die Überprüfung des Repository-Updates `skip_if_unavailable=False` zu erzwingen, fügen Sie es der Repository-Konfiguration hinzu.

Weitere Informationen zur Option `skip_if_available` finden Sie unter the section called “Konnektivität zur Patch-Quelle”.

Nur angegebene Repositorys werden für das Einspielen von Patches verwendet.

Angeben von alternativen Repositorys bedeutet nicht das Angeben zusätzlicher Repositorys. Sie können wählen, andere Repositorys als die auf einem verwalteten Knoten als Standardwerte konfigurierten festzulegen. Sie müssen jedoch auch die Standard-Repositorys als Teil der alternativen Patch-Quell-Konfiguration angeben, wenn Sie möchten, dass deren Updates übernommen werden.

Beispielsweise sind die Standard-Repositorys auf von Amazon Linux 2 verwalteten Knoten `amzn2-core` und `amzn2extra-docker`. Wenn Sie das Repository "Extra Packages for Enterprise Linux (EPEL)" in Ihre Patching-Operationen einschließen möchten, müssen Sie alle drei Repositorys als alternative Repositorys angeben.

> **Note**
>
> Wenn Sie eine benutzerdefinierte Patch-Baseline ausführen, die alternative Patch-Repositorys für einen verwalteten Knoten angeben, werden diese Repositorys dadurch nicht

zum neuen Standard-Repository auf dem Betriebssystem. Nach Abschluss der Patching-Operation bleiben die zuvor definierten Standard-Repositorys für das Betriebssystem des Knoten als Standard erhalten.

Das Patching-Verhalten für YUM-basierte Distributionen hängt vom updateinfo.xml-Manifest ab

Wenn Sie alternative Patch-Repositorys für YUM-basierte Distributionen festlegen, z. B. Amazon Linux 2 oder Red Hat Enterprise Linux, hängt das Patching-Verhalten davon ab, ob das Repository ein Update-Manifest in Form einer vollständig und korrekt formatierten `updateinfo.xml`-Datei enthält. Diese Datei gibt das Release-Datum, Klassifizierungen und Schweregrade der verschiedenen Pakete an. Jede der folgenden Optionen wirkt sich auf das Patching-Verhalten aus:

- Wenn Sie nach Klassifizierung und Schweregrad filtern, diese aber nicht in `updateinfo.xml` angegeben sind, wird das Paket nicht in Filter aufgenommen. Dies bedeutet auch, dass Pakete ohne eine `updateinfo.xml`-Datei werden nicht in das Patching eingeschlossen werden.
- Wenn Sie nach filtern ApprovalAfterDays, aber das Veröffentlichungsdatum des Pakets nicht im Format Unix Epoch ist (oder kein Veröffentlichungsdatum angegeben ist), wird das Paket nicht in den Filter aufgenommen.
- Eine Ausnahme besteht, wenn Sie das Kontrollkästchen Genehmigte Patches umfassen nicht sicherheitsrelevante Updates auf der Seite Patch-Baseline erstellen aktivieren. In diesem Fall werden Pakete ohne eine `updateinfo.xml`-Datei (oder die diese Datei ohne ordnungsgemäß formatierte Werte für Klassifizierung, Schweregrad und Datum enthalten) in die vorgefilterte Liste der Patches aufgenommen. (Diese müssen nach wie vor die übrigen Anforderungen Patch-Baseline Regel erfüllen, damit sie installiert werden.)

Anwendungsbeispiele für alternative Patch-Quell-Repositorys

Beispiel 1: Nicht sicherheitsrelevante Updates für Ubuntu Server

Sie verwenden bereitsPatch Manager, um Sicherheitspatches auf einer Flotte von Ubuntu Server verwalteten Knoten mithilfe der von AWS Ihnen bereitgestellten vordefinierten Patch-Baseline zu installieren. `AWS-UbuntuDefaultPatchBaseline` Sie können eine neue Patch-Baseline erstellen, die auf diesem Standard basiert, aber in den Genehmigungsregeln angeben, dass nicht auf die Sicherheit bezogene Updates, die Teil der Standardverteilung sind, ebenfalls installiert werden sollen. Wenn diese Patch-Baseline für Ihre Knoten ausgeführt wird, werden sowohl sicherheitsbezogene als auch nicht-sicherheitsbezogene Patches angewendet. Sie können auch auswählen, dass

nicht sicherheitsbezogene Patches in den Patch-Ausnahmen, die Sie für eine Baseline angeben, genehmigt werden.

Beispiel 2: Personal Package Archives (PPA) für Ubuntu Server

Auf Ihren von Ubuntu Server verwalteten Knoten wird Software ausgeführt, die über ein Personal Package Archives (PPA) für Ubuntu verteilt wird. In diesem Fall erstellen Sie eine Patch-Baseline, die ein PPA-Repository angibt, das Sie auf dem verwalteten Knoten als das Quell-Repository für die Patch-Operation konfiguriert haben. Führen Sie anschließend mithilfe von Run Command das Patch-Baseline-Dokument auf den Knoten aus.

Beispiel 3: Interne Firmenanwendungen in unterstützten Amazon-Linux-Versionen

Sie müssen auf Ihren von Amazon Linux verwalteten Knoten einige Anwendungen ausführen, die für die Compliance gesetzlicher Vorschriften und Branchenstandards erforderlich sind. Sie können ein Repository für diese Anwendungen auf den Knoten konfigurieren, mit YUM die Anwendungen erstmals installieren und eine neue Patch-Baseline aktualisieren oder erstellen, um dieses neue Unternehmens-Repository hinzuzufügen. Anschließend können Sie mit Run Command das Dokument `AWS-RunPatchBaseline` mit der Option `Scan` ausführen, um zu prüfen, ob das Unternehmenspaket unter den installierten Paketen aufgelistet und auf dem verwalteten Knoten aktuell ist. Wenn es nicht aktuell ist, können Sie das Dokument mithilfe der Option `Install` erneut ausführen, um die Anwendungen zu aktualisieren.

Wie Patches installiert werden

Patch Manager, ein Tool in AWS Systems Manager, verwendet den im Betriebssystem integrierten Paketmanager, um Updates auf verwalteten Knoten zu installieren. Beispielsweise verwendet es die Windows Update API `DNF` auf Windows Server und unter Amazon Linux 2023. Patch Manager berücksichtigt bestehende Paketmanager- und Repository-Konfigurationen auf den Knoten, einschließlich Einstellungen wie Repository-Status URLs, Spiegelung, GPG-Überprüfung und Optionen wie`skip_if_unavailable`.

Patch Manager installiert kein neues Paket, das ein veraltetes Paket ersetzt, das derzeit installiert ist. (Ausnahmen: Das neue Paket hängt davon ab, ob ein anderes Paket-Update installiert wird, oder das neue Paket hat denselben Namen wie das veraltete Paket.) Stattdessen meldet und installiert Patch Manager verfügbare Updates für installierte Pakete. Dieser Ansatz trägt dazu bei, unerwartete Änderungen an der Systemfunktionalität zu verhindern, die auftreten können, wenn ein Paket ein anderes ersetzt.

Wenn Sie ein veraltetes Paket deinstallieren und dessen Ersatz installieren müssen, müssen Sie möglicherweise ein benutzerdefiniertes Skript verwenden oder Paket-Manager-Befehle außerhalb der Standardvorgänge von Patch Manager verwenden.

Der Rest dieses Abschnitts erläutert, wie Patch Manager Patches auf einem Betriebssystem installiert.

Amazon Linux 2 and Amazon Linux 2023

Auf von Amazon Linux 2 und Amazon Linux 2023 verwalteten Knoten sieht der Workflow der Patch-Installation wie folgt aus:

1. Wenn eine Liste mit Patches mithilfe einer HTTPS-URL oder einer Amazon Simple Storage Service (Amazon S3)-URL im PathStyle über den `InstallOverrideList`-Parameter für die `AWS-RunPatchBaseline`- oder `AWS-RunPatchBaselineAssociation`-Dokumente angegeben wird, werden die aufgelisteten Patches installiert, und die Schritte 2-7 werden übersprungen.
2. Wenden Sie GlobalFilters wie in der Patch-Baseline an und behalten Sie nur die qualifizierten Pakete zur weiteren Verarbeitung.
3. Wenden Sie ApprovalRules wie in der Patch-Baseline angegeben an. Jede Genehmigungsregel kann ein Paket als genehmigt definieren.

   Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

   Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein.

   Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.
4. Wenden Sie ApprovedPatches wie in der Patch-Baseline angegeben an. Die genehmigten Patches sind für Updates genehmigt, auch wenn sie von GlobalFilters verworfen wurden oder wenn keine in ApprovalRules festgelegte Genehmigungsregel ihnen diese Genehmigung erteilt.

5. Wenden Sie RejectedPatches wie in der Patch-Baseline angegeben an. Die abgelehnten Patches werden aus der Liste der genehmigten Patches entfernt und werden nicht angewendet.
6. Wenn mehrere Versionen von Patches genehmigt wurden, wird die neueste Version angewendet.
7. Die YUM-Aktualisierungs-API (Amazon Linux 2) oder die DNF-Aktualisierungs-API (Amazon Linux 2023) wird wie folgt auf genehmigte Patches angewendet:
   - Für vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, werden nur die in `updateinfo.xml` angegebenen Patches angewendet (nur Sicherheitsupdates). Dies liegt daran, dass das Kontrollkästchen Funktionsupdates einschließen nicht aktiviert ist. Die vordefinierten Baselines entsprechen einer benutzerdefinierten Baseline mit folgenden Eigenschaften:
     - Das Kontrollkästchen Funktionsupdates einschließen ist nicht aktiviert
     - Eine Liste des SCHWEREGRADE von `[Critical, Important]`
     - Eine Liste der KLASSIFIZIERUNGEN von `[Security, Bugfix]`

     Für Amazon Linux 2 lautet der entsprechende YUM-Befehl für diesen Workflow wie folgt:

     ```
     sudo yum update-minimal --sec-severity=Critical,Important --bugfix -y
     ```

     Für Amazon Linux 2023 lautet der entsprechende DNF-Befehl für diesen Workflow wie folgt:

     ```
     sudo dnf upgrade-minimal --sec-severity=Critical --sec-severity=Important --
     bugfix -y
     ```

     Wenn das Kontrollkästchen Funktionsupdates einschließen aktiviert ist, werden alle Patches angewendet (Sicherheits- und Funktionsupdates), die in `updateinfo.xml` und nicht in `updateinfo.xml` sind.

     Für Amazon Linux 2, wenn eine Baseline mit Funktionsupdates einschließen ausgewählt ist, und die eine SCHWEREGRAD-Liste von `[Critical, Important]` und eine KLASSIFIKATION-Liste von `[Security, Bugfix]` hat, lautet der entsprechende YUM-Befehl:

     ```
     sudo yum update --security --sec-severity=Critical,Important --bugfix -y
     ```

Für Amazon Linux 2023 lautet der entsprechende DNF-Befehl wie folgt:

```
sudo dnf upgrade --security --sec-severity=Critical --sec-severity=Important --
bugfix -y
```

> **Note**
>
> Neue Pakete, die veraltete Pakete mit anderen Namen ersetzen, werden installiert, wenn Sie diese `yum`- oder `dnf`-Befehle außerhalb von Patch Manager ausführen. Sie werden jedoch nicht mit den entsprechenden Patch Manager-Vorgängen installiert.

Zusätzliche Patchdetails für Amazon Linux 2023

Support für den Schweregrad „Keine“

Amazon Linux 2023 unterstützt auch den Patch-Schweregrad `None`, der vom DNF-Paket-Manager erkannt wird.

Support für den Schweregrad „Mittel“

Für Amazon Linux 2023 entspricht ein Patch-Schweregrad von `Medium` einem Schweregrad von `Moderate`, der möglicherweise in einigen externen Repositorys definiert ist. Wenn Sie Patches mit dem `Medium`-Schweregrad in die Patch-Baseline aufnehmen, werden auch Patches mit dem `Moderate`-Schweregrad von externen Patches auf den Instances installiert.

Wenn Sie Konformitätsdaten mit der API-Aktion DescribeInstancePatches abfragen, werden beim Filtern nach dem Schweregrad `Medium` Patches mit den Schweregraden `Medium` und `Moderate` gemeldet.

Transitive Abhängigkeitsbehandlung für Amazon Linux 2023

Für Amazon Linux 2023 installiert Patch Manager möglicherweise andere Versionen von transitiven Abhängigkeiten als die entsprechenden `dnf`-Befehle install. Transitive Abhängigkeiten sind Pakete, die automatisch installiert werden, um die Anforderungen anderer Pakete zu erfüllen (Abhängigkeiten von Abhängigkeiten).

`dnf upgrade-minimal --security` installiert beispielsweise die Minimalversionen transitiver Abhängigkeiten, die zur Lösung bekannter Sicherheitsprobleme erforderlich

sind, und Patch Manager installiert gleichzeitig die neuesten verfügbaren Versionen derselben transitiven Abhängigkeiten.

8. Der verwaltete Knoten wird neu gestartet, wenn Updates installiert wurden. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

> **Note**
>
> Eine Standardkonfiguration für einen Paketmanager auf einer Linux-Distribution könnte so eingestellt sein, dass ein Paket-Repository, das nicht erreichbar ist, ohne Fehler übersprungen wird. In solchen Fällen wird der entsprechende Patchvorgang ohne Installation von Updates aus dem Projektarchiv fortgesetzt und erfolgreich abgeschlossen. Um Repository-Updates zu erzwingen, fügen Sie `skip_if_unavailable=False` es der Repository-Konfiguration hinzu.
> Weitere Informationen zur Option `skip_if_available` finden Sie unter the section called “Konnektivität zur Patch-Quelle”.

CentOS Stream

Auf von CentOS Stream verwalteten Knoten sieht der Patch-Installations-Workflow wie folgt aus:

1. Wenn eine Liste mit Patches mithilfe einer HTTPS-URL oder einer Amazon Simple Storage Service (Amazon S3)-URL im PathStyle über den `InstallOverrideList`-Parameter für die `AWS-RunPatchBaseline`- oder `AWS-RunPatchBaselineAssociation`-Dokumente angegeben wird, werden die aufgelisteten Patches installiert, und die Schritte 2-7 werden übersprungen.

   Wenden Sie GlobalFilters wie in der Patch-Baseline an und behalten Sie nur die qualifizierten Pakete zur weiteren Verarbeitung.

2. Wenden Sie ApprovalRules wie in der Patch-Baseline angegeben an. Jede Genehmigungsregel kann ein Paket als genehmigt definieren.

   Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein.

Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.

3. Wenden Sie ApprovedPatches wie in der Patch-Baseline angegeben an. Die genehmigten Patches sind für Updates genehmigt, auch wenn sie von GlobalFilters verworfen wurden oder wenn keine in ApprovalRules festgelegte Genehmigungsregel ihnen diese Genehmigung erteilt.
4. Wenden Sie RejectedPatches wie in der Patch-Baseline angegeben an. Die abgelehnten Patches werden aus der Liste der genehmigten Patches entfernt und werden nicht angewendet.
5. Wenn mehrere Versionen von Patches genehmigt wurden, wird die neueste Version angewendet.
6. Das DNF-Update auf CentOS Stream wird auf genehmigte Patches angewendet.

> **Note**
>
> Für CentOS Stream installiert Patch Manager möglicherweise andere Versionen transitiver Abhängigkeiten als die entsprechenden `dnf`-Befehle install. Transitive Abhängigkeiten sind Pakete, die automatisch installiert werden, um die Anforderungen anderer Pakete zu erfüllen (Abhängigkeiten von Abhängigkeiten). `dnf upgrade-minimal --security` installiert beispielsweise die Minimalversionen transitiver Abhängigkeiten, die zur Lösung bekannter Sicherheitsprobleme erforderlich sind, und Patch Manager installiert gleichzeitig die neuesten verfügbaren Versionen derselben transitiven Abhängigkeiten.

7. Der verwaltete Knoten wird neu gestartet, wenn Updates installiert wurden. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

Debian Server

Auf Debian Server-Instances sieht der Patch-Installations-Workflow wie folgt aus:

1. Wenn eine Liste mit Patches mithilfe einer HTTPS-URL oder einer Amazon Simple Storage Service (Amazon S3)-URL im PathStyle über den `InstallOverrideList`-Parameter für die `AWS-RunPatchBaseline`- oder `AWS-RunPatchBaselineAssociation`-Dokumente angegeben wird, werden die aufgelisteten Patches installiert, und die Schritte 2-7 werden übersprungen.
2. Wenn eine Aktualisierung für `python3-apt` (eine Python-Bibliotheks-Schnittstelle zu `libapt`) verfügbar ist, wird es auf die neueste Version aktualisiert. (Dieses nicht sicherheitsrelevante Paket wird aktualisiert, auch wenn Sie die Option Mit nicht sicherheitsrelevanten Updates nicht ausgewählt haben.)
3. Wenden Sie GlobalFilters wie in der Patch-Baseline an und behalten Sie nur die qualifizierten Pakete zur weiteren Verarbeitung.
4. Wenden Sie ApprovalRules wie in der Patch-Baseline angegeben an. Jede Genehmigungsregel kann ein Paket als genehmigt definieren.

   > **Note**
   >
   > Da es nicht möglich ist, die Veröffentlichungstermine von Update-Paketen für Debian Server zuverlässig zu bestimmen, werden die Optionen für die automatische Genehmigung für dieses Betriebssystem nicht unterstützt.

   Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

   Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein.

   Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.

   > **Note**
   >
   > Für Debian Server sind Patch-Kandidaten-Versionen auf Patches beschränkt, die in `debian-security` enthalten sind.

5. Wenden Sie ApprovedPatches wie in der Patch-Baseline angegeben an. Die genehmigten Patches sind für Updates genehmigt, auch wenn sie von GlobalFilters verworfen wurden oder wenn keine in ApprovalRules festgelegte Genehmigungsregel ihnen diese Genehmigung erteilt.
6. Wenden Sie RejectedPatches wie in der Patch-Baseline angegeben an. Die abgelehnten Patches werden aus der Liste der genehmigten Patches entfernt und werden nicht angewendet.
7. Die APT-Bibliothek wird verwendet, um Upgrades für Pakete durchzuführen.

   > **Note**
   >
   > Patch Manager unterstützt nicht die Verwendung der `Pin-Priority`-APT-Option, um Paketen Prioritäten zuzuweisen. Patch Manager aggregiert verfügbare Updates aus allen aktivierten Repositorys und wählt das neueste Update aus, das der Baseline für jedes installierte Paket entspricht.

8. Der verwaltete Knoten wird neu gestartet, wenn Updates installiert wurden. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

macOS

Auf von macOS verwalteten Knoten sieht der Patch-Installations-Workflow wie folgt aus:

1. Die `/Library/Receipts/InstallHistory.plist`-Eigenschaft ist ein Datensatz der Software, die mit den `softwareupdate`- und `installer`-Paketmanagern installiert und aktualisiert wurde. Unter Verwendung der`pkgutil`-Befehlszeilen-Tool (für `installer`) und des `softwareupdate`-Paketmanagers werden CLI-Befehle ausgeführt, um diese Liste zu analysieren.

   Für `installer` umfasst die Antwort auf die CLI-Befehle Details zu `package name`, `version`, `volume`, `location` und `install-time`, aber nur der `package name` und die `version` werden von Patch Manager verwendet.

   Für `softwareupdate` umfasst die Antwort auf die CLI-Befehle den Paketnamen (`display name`),`version` und`date`, aber nur der Paketname und die Version werden vom Patch Manager verwendet.

Für Brew and Brew Cask unterstützt Homebrew seine Befehle, die unter dem Root-Benutzer ausgeführt werden, nicht. Darum fragt Patch Manager entweder als Eigentümer des Homebrew-Verzeichnisses oder als gültiger Benutzer,der zur Eigentümergruppe des Homebrew-Verzeichnisses gehört, Homebrew–Befehle ab und führt sie aus. Die Befehle sind ähnlich wie `softwareupdate` und `installer` und werden durch einen Python-Subprozess ausgeführt, um Paketdaten zu sammeln. Die Ausgabe wird dann analysiert, um Paketnamen und -versionen zu identifizieren.

2. Wenden Sie GlobalFilters wie in der Patch-Baseline an und behalten Sie nur die qualifizierten Pakete zur weiteren Verarbeitung.
3. Wenden Sie ApprovalRules wie in der Patch-Baseline angegeben an. Jede Genehmigungsregel kann ein Paket als genehmigt definieren.
4. Wenden Sie ApprovedPatches wie in der Patch-Baseline angegeben an. Die genehmigten Patches sind für Updates genehmigt, auch wenn sie von GlobalFilters verworfen wurden oder wenn keine in ApprovalRules festgelegte Genehmigungsregel ihnen diese Genehmigung erteilt.
5. Wenden Sie RejectedPatches wie in der Patch-Baseline angegeben an. Die abgelehnten Patches werden aus der Liste der genehmigten Patches entfernt und werden nicht angewendet.
6. Wenn mehrere Versionen von Patches genehmigt wurden, wird die neueste Version angewendet.
7. Ruft die entsprechende Paket-CLI auf dem verwalteten Knoten auf, um genehmigte Patches wie folgt zu verarbeiten:

> **Note**
>
> `installer` fehlt die Funktion, um nach Updates zu suchen und sie zu installieren. Daher meldet Patch Manager für `installer` nur, welche Pakete installiert sind. Das Ergebnis: `installer`-Pakete werden nie als `Missing` gemeldet.

- Für vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, und für benutzerdefinierte Patch-Baselines, bei denen das Kästchen Funktionsupdates einschließen nicht ausgewählt wurde, werden nur Sicherheitsupdates angewendet.
- Bei benutzerdefinierten Patch-Baselines, bei denen das Kästchen Funktionsupdates einschließen aktiviert ist, werden sowohl Sicherheits- als auch Funktionsupdates angewendet.

8. Der verwaltete Knoten wird neu gestartet, wenn Updates installiert wurden. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

Oracle Linux

Auf von Oracle Linux verwalteten Knoten sieht der Patch-Installations-Workflow wie folgt aus:

1. Wenn eine Liste mit Patches mithilfe einer HTTPS-URL oder einer Amazon Simple Storage Service (Amazon S3)-URL im PathStyle über den `InstallOverrideList`-Parameter für die `AWS-RunPatchBaseline`- oder `AWS-RunPatchBaselineAssociation`-Dokumente angegeben wird, werden die aufgelisteten Patches installiert, und die Schritte 2-7 werden übersprungen.
2. Wenden Sie GlobalFilters wie in der Patch-Baseline an und behalten Sie nur die qualifizierten Pakete zur weiteren Verarbeitung.
3. Wenden Sie ApprovalRules wie in der Patch-Baseline angegeben an. Jede Genehmigungsregel kann ein Paket als genehmigt definieren.

   Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

   Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein.

   Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.
4. Wenden Sie ApprovedPatches wie in der Patch-Baseline angegeben an. Die genehmigten Patches sind für Updates genehmigt, auch wenn sie von GlobalFilters verworfen wurden oder wenn keine in ApprovalRules festgelegte Genehmigungsregel ihnen diese Genehmigung erteilt.
5. Wenden Sie RejectedPatches wie in der Patch-Baseline angegeben an. Die abgelehnten Patches werden aus der Liste der genehmigten Patches entfernt und werden nicht angewendet.

6. Wenn mehrere Versionen von Patches genehmigt wurden, wird die neueste Version angewendet.
7. Auf von Version 7 verwalteten Knoten wird die YUM-Update-API wie folgt auf genehmigte Patches angewendet:
   - Für vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, und für benutzerdefinierte Patch-Baselines, bei denen das Kästchen Funktionsupdates einschließen nicht ausgewählt wurde, werden nur Patches, die in `updateinfo.xml` angegeben sind (nur Sicherheitsupdates), angewendet.

     Der entsprechende Yum-Befehl für diesen Workflow lautet:

     ```
     sudo yum update-minimal --sec-severity=Important,Moderate --bugfix -y
     ```

   - Bei benutzerdefinierten Patch-Baselines, bei denen das Kästchen Funktionsupdates einschließen aktiviert ist, werden sowohl Patches aus der Datei `updateinfo.xml` als auch solche, die nicht in `updateinfo.xml` enthalten sind, angewendet (sowohl Sicherheits- als auch Funktionsupdates).

     Der entsprechende Yum-Befehl für diesen Workflow lautet:

     ```
     sudo yum update --security --bugfix -y
     ```

   Auf von Version 8 und 9 verwalteten Knoten wird die DNF-Update-API wie folgt auf genehmigte Patches angewendet:

   - Für vordefinierte Standard-Patch-Baselines, die von bereitgestellt werden AWS, und für benutzerdefinierte Patch-Baselines, bei denen das Kontrollkästchen Nicht sicherheitsrelevante Updates einbeziehen nicht aktiviert ist, `updateinfo.xml` werden nur die in angegebenen Patches angewendet (nur Sicherheitsupdates).

     Der entsprechende Yum-Befehl für diesen Workflow lautet:

     ```
     sudo dnf upgrade-minimal --security --sec-severity=Moderate --sec-
     severity=Important
     ```

     > **Note**
     >
     > Für Oracle Linux installiert Patch Manager möglicherweise andere Versionen von transitiven Abhängigkeiten als die entsprechenden `dnf`-Befehle install.

Transitive Abhängigkeiten sind Pakete, die automatisch installiert werden, um die Anforderungen anderer Pakete zu erfüllen (Abhängigkeiten von Abhängigkeiten). `dnf upgrade-minimal --security` installiert beispielsweise die Minimalversionen transitiver Abhängigkeiten, die zur Lösung bekannter Sicherheitsprobleme erforderlich sind, und Patch Manager installiert gleichzeitig die neuesten verfügbaren Versionen derselben transitiven Abhängigkeiten.

- Bei benutzerdefinierten Patch-Baselines, bei denen das Kästchen Funktionsupdates einschließen aktiviert ist, werden sowohl Patches aus der Datei `updateinfo.xml` als auch solche, die nicht in `updateinfo.xml` enthalten sind, angewendet (sowohl Sicherheits- als auch Funktionsupdates).

  Der entsprechende Yum-Befehl für diesen Workflow lautet:

  ```
  sudo dnf upgrade --security --bugfix
  ```

> **Note**
>
> Neue Pakete, die veraltete Pakete mit anderen Namen ersetzen, werden installiert, wenn Sie diese `yum`- oder `dnf`-Befehle außerhalb von Patch Manager ausführen. Sie werden jedoch nicht mit den entsprechenden Patch Manager-Vorgängen installiert.

8. Der verwaltete Knoten wird neu gestartet, wenn Updates installiert wurden. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

> **Note**
>
> Eine Standardkonfiguration für einen Paketmanager auf einer Linux-Distribution kann so eingestellt sein, dass ein Paket-Repository, das nicht erreichbar ist, ohne Fehler übersprungen wird. In solchen Fällen wird der entsprechende Patchvorgang ohne Installation von Updates aus dem Projektarchiv fortgesetzt und erfolgreich abgeschlossen. Um Repository-Updates zu erzwingen, fügen Sie `skip_if_unavailable=False` es der Repository-Konfiguration hinzu.

Weitere Informationen zur Option `skip_if_available` finden Sie unter the section called “Konnektivität zur Patch-Quelle”.

AlmaLinux, RHEL, and Rocky Linux

Auf AlmaLinuxRed Hat Enterprise Linux, und Rocky Linux verwalteten Knoten sieht der Arbeitsablauf für die Patch-Installation wie folgt aus:

1. Wenn eine Liste mit Patches mithilfe einer HTTPS-URL oder einer Amazon Simple Storage Service (Amazon S3)-URL im PathStyle über den `InstallOverrideList`-Parameter für die `AWS-RunPatchBaseline`- oder `AWS-RunPatchBaselineAssociation`-Dokumente angegeben wird, werden die aufgelisteten Patches installiert, und die Schritte 2-7 werden übersprungen.
2. Wenden Sie GlobalFilters wie in der Patch-Baseline an und behalten Sie nur die qualifizierten Pakete zur weiteren Verarbeitung.
3. Wenden Sie ApprovalRules wie in der Patch-Baseline angegeben an. Jede Genehmigungsregel kann ein Paket als genehmigt definieren.

   Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

   Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein.

   Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.
4. Wenden Sie ApprovedPatches wie in der Patch-Baseline angegeben an. Die genehmigten Patches sind für Updates genehmigt, auch wenn sie von GlobalFilters verworfen wurden oder wenn keine in ApprovalRules festgelegte Genehmigungsregel ihnen diese Genehmigung erteilt.
5. Wenden Sie RejectedPatches wie in der Patch-Baseline angegeben an. Die abgelehnten Patches werden aus der Liste der genehmigten Patches entfernt und werden nicht angewendet.

6. Wenn mehrere Versionen von Patches genehmigt wurden, wird die neueste Version angewendet.
7. Die YUM-Update-API (auf RHEL 7) oder die DNF-Update-API (auf AlmaLinux 8 und 9, RHEL 8, 9 und 10 sowie Rocky Linux 8 und 9) wird gemäß den folgenden Regeln auf genehmigte Patches angewendet:

   Szenario 1: Nicht sicherheitsrelevante Updates ausgeschlossen

   - Gilt für: Vordefinierte Standard-Patch-Baselines, die von bereitgestellt werden, AWS und benutzerdefinierte Patch-Baselines.
   - Das Kontrollkästchen Funktionsupdates einschließen ist nicht aktiviert
   - Angewendete Patches: Die unter `updateinfo.xml` (nur Sicherheitsupdates) angegebenen Patches werden nur angewendet, wenn sie beide der Patch-Baseline-Konfiguration entsprechen und sich in den konfigurierten Repositorys befinden.

     In einigen Fällen ist ein in `updateinfo.xml` spezifizierter Patch möglicherweise nicht mehr in einem konfigurierten Repository verfügbar. Konfigurierte Repositorys verfügen normalerweise nur über die neueste Version eines Patches, bei dem es sich um einen kumulativen Rollup aller vorherigen Updates handelt. Die neueste Version entspricht jedoch möglicherweise nicht den Patch-Baseline-Regeln und wird beim Patchen nicht berücksichtigt.
   - Befehle: Für RHEL 7 lautet der entsprechende YUM-Befehl für diesen Workflow wie folgt:

     ```
     sudo yum update-minimal --sec-severity=Critical,Important --bugfix -y
     ```

     Für AlmaLinux, RHEL 8 undRocky Linux, lauten die entsprechenden dnf-Befehle für diesen Workflow:

     ```
     sudo dnf update-minimal --sec-severity=Critical --bugfix -y ; \
     sudo dnf update-minimal --sec-severity=Important --bugfix -y
     ```

     > **Note**
     >
     > For AlmaLinuxRHEL, und Rocky Linux Rocky Linux installiert Patch Manager möglicherweise andere Versionen von transitiven Abhängigkeiten als die entsprechenden `dnf` Befehle install. Transitive Abhängigkeiten sind Pakete, die

automatisch installiert werden, um die Anforderungen anderer Pakete zu erfüllen (Abhängigkeiten von Abhängigkeiten).
`dnf upgrade-minimal --security` installiert beispielsweise die Minimalversionen transitiver Abhängigkeiten, die zur Lösung bekannter Sicherheitsprobleme erforderlich sind, und Patch Manager installiert gleichzeitig die neuesten verfügbaren Versionen derselben transitiven Abhängigkeiten.

Szenario 2: Nicht sicherheitsrelevante Updates enthalten

- Gilt für: Benutzerdefinierte Patch-Baselines.
- Das Kontrollkästchen Funktionsupdates einschließen ist aktiviert.
- Angewendete Patches: Patches in `updateinfo.xml` und nicht in `updateinfo.xml` werden angewendet (Sicherheits- und andere Updates).
- Befehle: Für RHEL 7 lautet der entsprechende YUM-Befehl für diesen Workflow wie folgt:

```
sudo yum update --security --bugfix -y
```

Für AlmaLinux 8 und 9, RHEL 8 und 9 sowie Rocky Linux 8 und 9 lautet der entsprechende dnf-Befehl für diesen Workflow:

```
sudo dnf update --security --bugfix -y
```

> **Note**
>
> Neue Pakete, die veraltete Pakete mit anderen Namen ersetzen, werden installiert, wenn Sie diese `yum`- oder `dnf`-Befehle außerhalb von Patch Manager ausführen. Sie werden jedoch nicht mit den entsprechenden Patch Manager-Vorgängen installiert.

8. Der verwaltete Knoten wird neu gestartet, wenn Updates installiert wurden. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

> **Note**
>
> Eine Standardkonfiguration für einen Paketmanager auf einer Linux-Distribution könnte so eingestellt sein, dass ein Paket-Repository, das nicht erreichbar ist, ohne Fehler übersprungen wird. In solchen Fällen wird der entsprechende Patchvorgang ohne Installation von Updates aus dem Projektarchiv fortgesetzt und erfolgreich abgeschlossen. Um Repository-Updates zu erzwingen, fügen Sie `skip_if_unavailable=False` es der Repository-Konfiguration hinzu.
> Weitere Informationen zur Option `skip_if_available` finden Sie unter the section called “Konnektivität zur Patch-Quelle”.

SLES

Auf von SUSE Linux Enterprise Server (SLES) verwalteten Knoten sieht der Patch-Installations-Workflow wie folgt aus:

1. Wenn eine Liste mit Patches mithilfe einer HTTPS-URL oder einer Amazon Simple Storage Service (Amazon S3)-URL im PathStyle über den `InstallOverrideList`-Parameter für die `AWS-RunPatchBaseline`- oder `AWS-RunPatchBaselineAssociation`-Dokumente angegeben wird, werden die aufgelisteten Patches installiert, und die Schritte 2-7 werden übersprungen.
2. Wenden Sie GlobalFilters wie in der Patch-Baseline an und behalten Sie nur die qualifizierten Pakete zur weiteren Verarbeitung.
3. Wenden Sie ApprovalRules wie in der Patch-Baseline angegeben an. Jede Genehmigungsregel kann ein Paket als genehmigt definieren.

   Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

   Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein.

   Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.

4. Wenden Sie ApprovedPatches wie in der Patch-Baseline angegeben an. Die genehmigten Patches sind für Updates genehmigt, auch wenn sie von GlobalFilters verworfen wurden oder wenn keine in ApprovalRules festgelegte Genehmigungsregel ihnen diese Genehmigung erteilt.
5. Wenden Sie RejectedPatches wie in der Patch-Baseline angegeben an. Die abgelehnten Patches werden aus der Liste der genehmigten Patches entfernt und werden nicht angewendet.
6. Wenn mehrere Versionen von Patches genehmigt wurden, wird die neueste Version angewendet.
7. Die Zypper-Update-API wird auf genehmigte Patches angewendet.
8. Der verwaltete Knoten wird neu gestartet, wenn Updates installiert wurden. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

Ubuntu Server

Auf von Ubuntu Server verwalteten Knoten sieht der Patch-Installations-Workflow wie folgt aus:

1. Wenn eine Liste mit Patches mithilfe einer HTTPS-URL oder einer Amazon Simple Storage Service (Amazon S3)-URL im PathStyle über den `InstallOverrideList`-Parameter für die `AWS-RunPatchBaseline`- oder `AWS-RunPatchBaselineAssociation`-Dokumente angegeben wird, werden die aufgelisteten Patches installiert, und die Schritte 2-7 werden übersprungen.
2. Wenn eine Aktualisierung für `python3-apt` (eine Python-Bibliotheks-Schnittstelle zu `libapt`) verfügbar ist, wird es auf die neueste Version aktualisiert. (Dieses nicht sicherheitsrelevante Paket wird aktualisiert, auch wenn Sie die Option Mit nicht sicherheitsrelevanten Updates nicht ausgewählt haben.)
3. Wenden Sie GlobalFilters wie in der Patch-Baseline an und behalten Sie nur die qualifizierten Pakete zur weiteren Verarbeitung.
4. Wenden Sie ApprovalRules wie in der Patch-Baseline angegeben an. Jede Genehmigungsregel kann ein Paket als genehmigt definieren.

> **Note**
>
> Da es nicht möglich ist, die Veröffentlichungsdaten von Updatepaketen für Ubuntu Server zuverlässig zu bestimmen, werden die Optionen für die automatische Genehmigung für dieses Betriebssystem nicht unterstützt.

Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein.

Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.

Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

> **Note**
>
> Für jede Version von Ubuntu Server sind die Patchkandidatenversionen wie folgt auf Patches beschränkt, die Teil des zugehörigen Repositorys für diese Version sind:
>
> - Ubuntu Server 16.04 LTS: `xenial-security`
> - Ubuntu Server 18.04 LTS: `bionic-security`
> - Ubuntu Server 20.04 LTS): `focal-security`
> - Ubuntu Server 22.04 LTS: `jammy-security`
> - Ubuntu Server 24.04 LTS (`noble-security`)
> - Ubuntu Server 25.04 (`plucky-security`)

5. Wenden Sie ApprovedPatches wie in der Patch-Baseline angegeben an. Die genehmigten Patches sind für Updates genehmigt, auch wenn sie von GlobalFilters verworfen wurden oder wenn keine in ApprovalRules festgelegte Genehmigungsregel ihnen diese Genehmigung erteilt.
6. Wenden Sie RejectedPatches wie in der Patch-Baseline angegeben an. Die abgelehnten Patches werden aus der Liste der genehmigten Patches entfernt und werden nicht angewendet.
7. Die APT-Bibliothek wird verwendet, um Upgrades für Pakete durchzuführen.

   > **Note**
   >
   > Patch Manager unterstützt nicht die Verwendung der `Pin-Priority`-APT-Option, um Paketen Prioritäten zuzuweisen. Patch Manager aggregiert verfügbare Updates aus allen aktivierten Repositorys und wählt das neueste Update aus, das der Baseline für jedes installierte Paket entspricht.

8. Der verwaltete Knoten wird neu gestartet, wenn Updates installiert wurden. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

Windows Server

Wenn eine Patch-Operation auf einem von Windows Server verwalteten Knoten durchgeführt wird, fordert die Instance einen Snapshot der entsprechenden Patch-Baseline von Systems Manager an. Dieser Snapshot enthält die Liste aller in der Patch-Baseline verfügbaren Updates, die für die Bereitstellung genehmigt wurden. Diese Liste der Updates wird an die Windows-Update-API gesendet, die festlegt, welche Updates für den verwalteten Knoten zutreffen, und diese bei Bedarf installiert. Windows erlaubt nur die Installation der neuesten verfügbaren Version einer KB. Patch Manager installiert die neueste Version einer KB, wenn sie oder eine frühere Version der KB mit der angewendeten Patch-Baseline übereinstimmt. Wenn Updates installiert sind, wird der verwaltete Knoten danach so oft wie nötig neu gestartet, bis alle erforderlichen Patches abgeschlossen sind. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.) Die Zusammenfassung des Patch-Vorgangs finden Sie in der Ausgabe der Run Command-Anfrage. Zusätzliche Protokolle finden Sie auf dem verwalteten Knoten im `%PROGRAMDATA%\Amazon\PatchBaselineOperations\Logs`-Ordner.

Da die Windows Update-API zum Herunterladen und Installieren verwendet wird KBs, werden alle Gruppenrichtlinieneinstellungen für Windows Update berücksichtigt. Für die Verwendung von Patch Manager sind keine Gruppenrichtlinien-Einstellungen erforderlich. Allerdings werden alle definierten Einstellungen angewendet, wie etwa die Weiterleitung von verwalteten Knoten an einen Windows Server Update Services (WSUS)-Server.

> **Note**
>
> Standardmäßig lädt Windows alles KBs von der Windows Update-Website von Microsoft herunter, da die Windows Update-API Patch Manager verwendet wird, um den Download und die Installation von voranzutreiben KBs. Der verwaltete Knoten muss daher die Website von Microsoft Windows Update erreichen können. Andernfalls tritt ein Fehler bei der Patch-Operation auf. Alternativ können Sie einen WSUS-Server als KB-Repository konfigurieren und Ihre verwalteten Knoten so konfigurieren, dass sie den WSUS-Server anvisieren, anstatt Gruppenrichtlinien zu verwenden.
> Patch Managerverweist möglicherweise auf KB, IDs wenn benutzerdefinierte Patch-Baselines für erstellt werdenWindows Server, z. B. wenn eine Liste zulässiger Patches oder abgelehnter Patches in der Basiskonfiguration enthalten ist. Nur Updates, denen in Microsoft Windows Update oder einem WSUS-Server eine KB-ID zugewiesen wurde, werden von Patch Manager installiert. Updates, denen eine KB-ID fehlt, sind nicht in den Patchvorgängen enthalten.
> Informationen zum Erstellen benutzerdefinierter Patch-Baselines finden Sie in den folgenden Themen:
>
> - Erstellen einer benutzerdefinierten Patch-Baseline für Windows Server
> - Erstellen einer Patch-Baseline (CLI)
> - Paketnamen-Formate für Windows-Betriebssysteme

Funktionsweise von Patch-Baseline-Regeln auf Linux-basierten Systemen

Die Regeln in einer Patch-Baseline für Linux-Verteilungen funktionieren je nach Verteilungstyp unterschiedlich. Im Gegensatz zu Patch-Updates auf Windows Server verwalteten Knoten werden Regeln auf jedem Knoten ausgewertet, um die konfigurierten Repos auf der Instanz zu berücksichtigen. Patch Manager, ein Tool in AWS Systems Manager, verwendet den systemeigenen Paketmanager, um die Installation von Patches voranzutreiben, die von der Patch-Baseline genehmigt wurden.

Für Linux-basierte Betriebssysteme, die einen Schweregrad für Patches melden, verwendet Patch Manager den vom Software-Publisher gemeldeten Schweregrad für den Update-Hinweis oder den einzelnen Patch. Patch Manager leitet keinen Schweregrad aus Drittquellen wie dem Common Vulnerability Scoring System (CVSS) oder aus Metriken ab, die von der National Vulnerability Database (NVD) veröffentlicht werden.

Themen

- Funktionsweise von Patch-Baseline-Regeln unter Amazon Linux 2 und Amazon Linux 2023
- Funktionsweise von Patch-Baseline-Regeln auf CentOS Stream
- Funktionsweise von Patch-Baseline-Regeln auf Debian Server
- Funktionsweise von Patch-Baseline-Regeln auf macOS
- Funktionsweise von Patch-Baseline-Regeln auf Oracle Linux
- So funktionieren Patch-Basisregeln für AlmaLinuxRHEL, und Rocky Linux
- Funktionsweise von Patch-Baseline-Regeln auf SUSE Linux Enterprise Server
- Funktionsweise von Patch-Baseline-Regeln auf Ubuntu Server

Funktionsweise von Patch-Baseline-Regeln unter Amazon Linux 2 und Amazon Linux 2023

> **Note**
>
> Amazon Linux 2023 (AL2023) verwendet versionierte Repositorys, die über eine oder mehrere Systemeinstellungen für eine bestimmte Version gesperrt werden können. Patch ManagerVerwendet für alle Patch-Operationen auf AL2 EC2 023-Instances unabhängig von der Systemkonfiguration die neuesten Repository-Versionen. Weitere Informationen finden Sie unter Deterministische Upgrades durch versionierte Repositories im Benutzerhandbuch von Amazon Linux 2023.

Unter Amazon Linux 2 und Amazon Linux 2023 erfolgt die Patch-Auswahl folgendermaßen:

1. Auf dem verwalteten Knoten greift die YUM-Bibliothek (Amazon Linux 2) oder die DNF-Bibliothek (Amazon Linux 2023) auf die `updateinfo.xml`-Datei für jedes konfigurierte Repository zu.

   Wenn keine `updateinfo.xml`-Datei gefunden wird, hängt es von den Einstellungen für Funktionsupdates einschließen und Automatische Genehmigung ab, ob Patches installiert werden.

Wenn beispielsweise nicht sicherheitsrelevante Updates zulässig sind, werden sie installiert, wenn die automatische Genehmigung eintrifft.

2. Jeder Update-Hinweis in `updateinfo.xml` enthält mehrere Attribute, die die Eigenschaften der Pakete im Hinweis kennzeichnen, wie in der folgenden Tabelle beschrieben.

   Update-Hinweis-Attribute

Attribut	Description
Typ	Entspricht dem Wert des Klassifizierungsschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Typ des im Update-Hinweis enthaltenen Pakets. Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation anzeigen. DescribePatchProperties Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.
severity	Entspricht dem Wert des Schweregradschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Schweregrad der im Update-Hinweis enthaltenen Pakete. Gilt in der Regel nur für Update-Hinweise im Hinblick auf die Sicherheit. Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation anzeigenDescribePatchProperties. Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.

Attribut	Description
update_id	Kennzeichnet die Advisory ID, wie etwa ALAS-2017-867. Die Advisory ID kann in der Patch-Baseline im ApprovedPatches oder RejectedPatches-Attribut verwendet werden.
Referenzen	Enthält weitere Informationen über den Update-Hinweis, wie etwa eine CVE ID (Format: CVE-2017-1234567). Die CVE ID kann in der Patch-Baseline im ApprovedPatches- oder im RejectedPatches-Attribut verwendet werden.
Aktualisiert	Entspricht ApproveAfterDays in der Patch-Baseline. Kennzeichnet das Veröffentlichungsdatum (Aktualisierungsdatum) der im Update-Hinweis enthaltenen Pakete. Ein Vergleich zwischen dem aktuellen Zeitstempel und dem Wert dieses Attributs plus `ApproveAfterDays` wird verwendet, um zu bestimmen, ob der Patch für die Bereitstellung genehmigt wurde.

Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen.

3. Das Produkt des verwalteten Knotens wird durch SSM Agent bestimmt. Dieses Attribut entspricht dem Wert des Produktschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline.
4. Pakete für das Update werden gemäß den folgenden Richtlinien ausgewählt.

Sicherheitsoption	Patch-Auswahl
Vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, und benutzerdefinierte Patch-Baselines, bei denen das	Für jeden Update-Hinweis in `updateinfo.xml` wird die Patch-Baseline als Filter verwendet, der nur den qualifizierten Paketen die Aufnahme in das Update erlaubt. Wenn

Sicherheitsoption	Patch-Auswahl

Kontrollkästchen Funktionsupdates einschließen nicht ausgewählt wurde	mehrere Pakete zutreffen, wird die aktuelle Version nach Anwenden der Patch-Baseline-Definition verwendet.
	
	Für Amazon Linux 2 lautet der entsprechende YUM-Befehl für diesen Workflow wie folgt:
	
	`sudo yum update-minimal --sec-severity=Critical,Important --bugfix -y`
	
	Für Amazon Linux 2023 lautet der entsprechende DNF-Befehl für diesen Workflow wie folgt:
	
	`sudo dnf upgrade-minimal --sec-severity=Critical --sec-severity=Important --bugfix -y`

Sicherheitsoption	Patch-Auswahl

Benutzerdefinierte Patch-Baselines, bei denen das Kontrollkästchen Funktionsupdates einschließen aktiviert ist, mit einer SCHWEREGRAD-Liste von `[Critical, Important]` und einer KLASSIFIZIERUNG-Liste von `[Security, Bugfix]`	Neben den aus `updateinfo.xml` ausgewählten Sicherheits-Updates wendet Patch Manager auch nicht sicherheitsrelevante Updates an, die ansonsten den Patch-Filterregeln entsprechen.
	
	Für Amazon Linux 2 lautet der entsprechende YUM-Befehl für diesen Workflow wie folgt:
	
	`sudo yum update --security --sec-severity=Critical,Important --bugfix -y`
	
	Für Amazon Linux 2023 lautet der entsprechende DNF-Befehl für diesen Workflow wie folgt:
	
	`sudo dnf upgrade --security --sec-severity=Critical --sec-severity=Important --bugfix -y`
	
	Note
	Neue Pakete, die veraltete Pakete mit anderen Namen ersetzen, werden installiert, wenn Sie diese `yum`- oder `dnf`-Befehle außerhalb von Patch Manager ausführen. Sie werden jedoch nicht mit den entsprechenden Patch Manager-Vorgängen installiert.

Weitere Informationen über Patch-Compliance-Statuswerte finden Sie unter Statuswerte der Patch-Compliance.

Funktionsweise von Patch-Baseline-Regeln auf CentOS Stream

Die CentOS Stream-Standard-Repositorys enthalten keine `updateinfo.xml`-Datei. Benutzerdefinierte Repositorys, die Sie erstellen oder verwenden, können diese Datei jedoch enthalten. In diesem Thema beziehen sich Verweise nur `updateinfo.xml` auf diese benutzerdefinierten Repositorys.

In CentOS Stream erfolgt die Patch-Auswahl folgendermaßen:

1. Auf dem verwalteten Knoten ruft die DNF-Bibliothek die `updateinfo.xml`-Datei für jedes konfigurierte Repository auf, sofern diese in einem benutzerdefinierten Repository vorhanden ist.

   Wenn kein `updateinfo.xml` gefunden wird (was immer die Standard-Repos einschließt), hängt die Installation von Patches von den Einstellungen für Nicht sicherheitsrelevante Updates einschließen und Automatische Genehmigung ab. Wenn beispielsweise nicht sicherheitsrelevante Updates zulässig sind, werden sie installiert, wenn die automatische Genehmigung eintrifft.
2. Wenn `updateinfo.xml` vorhanden ist, enthält jede Aktualisierungsmitteilung in der Datei mehrere Attribute, die die Eigenschaften der Pakete in der Mitteilung bezeichnen, wie in der folgenden Tabelle beschrieben.

   Update-Hinweis-Attribute

Attribut	Description
Typ	Entspricht dem Wert des Klassifizierungsschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Typ des im Update-Hinweis enthaltenen Pakets. Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation anzeigenDescribePatchProperties. Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.

Attribut	Description
severity	Entspricht dem Wert des Schweregradschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Schweregrad der im Update-Hinweis enthaltenen Pakete. Gilt in der Regel nur für Update-Hinweise im Hinblick auf die Sicherheit. Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation anzeigenDescribePatchProperties. Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.
update_id	Kennzeichnet die Advisory ID, wie beispielsweise CVE-2019-17055. Die Advisory ID kann in der Patch-Baseline im ApprovedPatches oder RejectedPatches-Attribut verwendet werden.
Referenzen	Enthält weitere Informationen über den Update-Hinweis, wie beispielsweise eine CVE-ID (Format: CVE-2019-17055) oder eine Bugzilla-ID (Format: 1463241). Die CVE ID und die Bugzilla ID können in der Patch-Baseline im ApprovedPatches- oder im RejectedPatches-Attribut verwendet werden.

Attribut	Description
Aktualisiert	Entspricht ApproveAfterDays in der Patch-Baseline. Kennzeichnet das Veröffentlichungsdatum (Aktualisierungsdatum) der im Update-Hinweis enthaltenen Pakete. Ein Vergleich zwischen dem aktuellen Zeitstempel und dem Wert dieses Attributs plus `ApproveAfterDays` wird verwendet, um zu bestimmen, ob der Patch für die Bereitstellung genehmigt wurde.

Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen.

3. Das Produkt des verwalteten Knotens wird in allen Fällen durch SSM Agent bestimmt. Dieses Attribut entspricht dem Wert des Produktschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline.
4. Pakete für das Update werden gemäß den folgenden Richtlinien ausgewählt.

Sicherheitsoption	Patch-Auswahl
Vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, und benutzerdefinierte Patch-Baselines, bei denen das Kontrollkästchen Funktionsupdates einschließen nicht ausgewählt wurde	Für jeden Aktualisierungshinweis in `updateinfo.xml` (sofern dieser in einem benutzerdefinierten Repository vorhanden ist) wird die Patch-Baseline als Filter verwendet, sodass nur die qualifizierten Pakete in die Aktualisierung aufgenommen werden. Wenn mehrere Pakete zutreffen, wird die aktuelle Version nach Anwenden der Patch-Baseline-Definition verwendet. Für CentOS Stream, wo `updateinfo.xml` vorhanden ist, lautet der entsprechende DNF-Befehl für diesen Workflow wie folgt:

Sicherheitsoption	Patch-Auswahl
	`sudo dnf upgrade-minimal --sec-severity=Critical --sec-severity=Important --bugfix -y`

Sicherheitsoption	Patch-Auswahl
Benutzerdefinierte Patch-Baselines, bei denen das Kontrollkästchen Funktions updates einschließen aktiviert ist, mit einer SCHWEREGRAD-Liste von `[Critical, Important]` und einer KLASSIFIZIERUNG-Liste von `[Security, Bugfix]`	Neben den aus `updateinfo.xml` ausgewählten Sicherheits-Updates, falls im benutzerdefinierten Repository vorhanden, wendet Patch Manager auch nicht sicherhei tsrelevante Updates an, die ansonsten den Patch-Filterregeln entsprechen. Für CentOS Stream, wo `updateinfo.xml` vorhanden ist, lautet der entsprechende DNF-Befehl für diesen Workflow wie folgt: `sudo dnf upgrade --security --sec-sev erity=Critical --sec-severity=Imp ortant --bugfix -y` Für Standard-Repositorys und benutzerd efinierte Repositorys ohne `updateinf o.xml` müssen Sie das Kontrollkästchen Nicht sicherheitsrelevante Updates einschlie ßen aktivieren, um Betriebssystempakete (OS) zu aktualisieren. **Note** Neue Pakete, die veraltete Pakete mit anderen Namen ersetzen, werden installiert, wenn Sie diese `yum`- oder `dnf`-Befehle außerhalb von Patch Manager ausführen. Sie werden jedoch nicht mit den entsprechenden Patch Manager-Vorgängen installiert.

Weitere Informationen über Patch-Compliance-Statuswerte finden Sie unter Statuswerte der Patch-Compliance.

Funktionsweise von Patch-Baseline-Regeln auf Debian Server

Auf Debian Server bietet der Patch-Baseline-Service Filter in den Feldern Priorität und Abschnitt. Diese Felder sind normalerweise für alle Debian Server-Pakete vorhanden. Um zu bestimmen, ob ein Patch von der Patch-Baseline ausgewählt wird, geht Patch Manager folgendermaßen vor:

1. Auf Debian Server-Systemen wird das Äquivalent von `sudo apt-get update` ausgeführt, um die Liste der verfügbaren Pakete zu aktualisieren. Repos sind nicht konfiguriert und die Daten werden aus Repos abgerufen, die in einer `sources`-Liste konfiguriert sind.
2. Wenn eine Aktualisierung für `python3-apt` (eine Python-Bibliotheks-Schnittstelle zu `libapt`) verfügbar ist, wird es auf die neueste Version aktualisiert. (Dieses nicht sicherheitsrelevante Paket wird aktualisiert, auch wenn Sie die Option Mit nicht sicherheitsrelevanten Updates nicht ausgewählt haben.)
3. Als Nächstes werden die Listen GlobalFilters, ApprovalRules, ApprovedPatches und RejectedPatches angewendet.

   > **Note**
   >
   > Da es nicht möglich ist, die Veröffentlichungstermine von Update-Paketen für Debian Server zuverlässig zu bestimmen, werden die Optionen für die automatische Genehmigung für dieses Betriebssystem nicht unterstützt.

   Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

   Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein. Für Debian Server sind Patch-Kandidaten-Versionen in diesem Fall auf Patches beschränkt, die in den folgenden Repos enthalten sind:

   Diese Repos werden wie folgt benannt:

   - Debian Server11: `debian-security bullseye`
   - Debian Server12: `debian-security bookworm`

Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.

Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen.

Zum Anzeigen der Inhalte der Felder Priorität und Abschnitt führen Sie den folgenden `aptitude`-Befehl aus:

> **Note**
>
> Möglicherweise müssen Sie zuerst Aptitude auf Debian Server-Systemen installieren.

```
aptitude search -F '%p %P %s %t %V#' '~U'
```

In der Antwort auf diesen Befehl werden alle Pakete, für die ein Upgrade durchgeführt werden kann, in diesem Format gemeldet:

```
name, priority, section, archive, candidate version
```

Weitere Informationen über Patch-Compliance-Statuswerte finden Sie unter Statuswerte der Patch-Compliance.

Funktionsweise von Patch-Baseline-Regeln auf macOS

In macOS erfolgt die Patch-Auswahl folgendermaßen:

1. Auf dem verwalteten Knoten greift Patch Manager auf den geparsten Inhalt der `InstallHistory.plist`-Datei zu und identifiziert Paketnamen und -versionen.

   Details zum Parsing-Prozess finden Sie im Abschnitt macOS in Wie Patches installiert werden.
2. Das Produkt des verwalteten Knotens wird durch SSM Agent bestimmt. Dieses Attribut entspricht dem Wert des Produktschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline.
3. Pakete für das Update werden gemäß den folgenden Richtlinien ausgewählt.

Sicherheitsoption	Patch-Auswahl
Vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, und benutzerdefinierte Patch-Baselines, bei denen das Kontrollkästchen Funktionsupdates einschließen nicht ausgewählt wurde	Für jedes verfügbare Paket-Update wird die Patch-Baseline als Filter verwendet, der nur den qualifizierten Paketen die Aufnahme in das Update erlaubt. Wenn mehrere Pakete zutreffen, wird die aktuelle Version nach Anwenden der Patch-Baseline-Definition verwendet.
Benutzerdefinierte Patch-Baselines, bei denen das Kontrollkästchen Funktionsupdates einschließen aktiviert ist	Neben den unter Verwendung von `InstallHistory.plist` identifizierten Sicherheits-Updates wendet Patchmanager auch nicht sicherheitsrelevante Updates an, die ansonsten den Patch-Filterregeln entsprechen.

Weitere Informationen über Patch-Compliance-Statuswerte finden Sie unter Statuswerte der Patch-Compliance.

Funktionsweise von Patch-Baseline-Regeln auf Oracle Linux

In Oracle Linux erfolgt die Patch-Auswahl folgendermaßen:

1. Auf dem verwalteten Knoten ruft die YUM-Bibliothek die `updateinfo.xml`-Datei für jedes konfigurierte Repo auf.

   > **Note**
   >
   > Die `updateinfo.xml`-Datei ist möglicherweise nicht verfügbar, wenn das Repo nicht von Oracle verwaltet wird. Wenn keine `updateinfo.xml`-Datei gefunden wird, hängt es von den Einstellungen für Funktionsupdates einschließen und Automatische Genehmigung ab, ob Patches installiert werden. Wenn beispielsweise nicht sicherheitsrelevante Updates zulässig sind, werden sie installiert, wenn die automatische Genehmigung eintrifft.

2. Jeder Update-Hinweis in `updateinfo.xml` enthält mehrere Attribute, die die Eigenschaften der Pakete im Hinweis kennzeichnen, wie in der folgenden Tabelle beschrieben.

Update-Hinweis-Attribute

Attribut	Description
Typ	Entspricht dem Wert des Klassifizierungsschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Typ des im Update-Hinweis enthaltenen Pakets. Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation anzeigenDescribePatchProperties. Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.
severity	Entspricht dem Wert des Schweregradschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Schweregrad der im Update-Hinweis enthaltenen Pakete. Gilt in der Regel nur für Update-Hinweise im Hinblick auf die Sicherheit. Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation anzeigenDescribePatchProperties. Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.

Attribut	Description
update_id	Kennzeichnet die Advisory ID, wie beispielsweise CVE-2019-17055. Die Advisory ID kann in der Patch-Baseline im ApprovedPatches oder RejectedPatches-Attribut verwendet werden.
Referenzen	Enthält weitere Informationen über den Update-Hinweis, wie beispielsweise eine CVE-ID (Format: CVE-2019-17055) oder eine Bugzilla-ID (Format: 1463241). Die CVE ID und die Bugzilla ID können in der Patch-Baseline im ApprovedPatches- oder im RejectedPatches-Attribut verwendet werden.
Aktualisiert	Entspricht ApproveAfterDays in der Patch-Baseline. Kennzeichnet das Veröffentlichungsdatum (Aktualisierungsdatum) der im Update-Hinweis enthaltenen Pakete. Ein Vergleich zwischen dem aktuellen Zeitstempel und dem Wert dieses Attributs plus `ApproveAfterDays` wird verwendet, um zu bestimmen, ob der Patch für die Bereitstellung genehmigt wurde.

Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen.

3. Das Produkt des verwalteten Knotens wird durch SSM Agent bestimmt. Dieses Attribut entspricht dem Wert des Produktschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline.
4. Pakete für das Update werden gemäß den folgenden Richtlinien ausgewählt.

Sicherheitsoption	Patch-Auswahl
Vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, und benutzerd	Für jeden Update-Hinweis in `updateinfo.xml` wird die Patch-Baseline als Filter

Sicherheitsoption	Patch-Auswahl

efinierte Patch-Baselines, bei denen das Kontrollkästchen Funktionsupdates einschlie ßen nicht ausgewählt wurde	verwendet, der nur den qualifizierten Paketen die Aufnahme in das Update erlaubt. Wenn mehrere Pakete zutreffen, wird die aktuelle Version nach Anwenden der Patch-Baseline-Definition verwendet.
	
	Für von Version 7 verwaltete Knoten lautet der entsprechende Yum-Befehl für diesen Workflow:
	
	`sudo yum update-minimal --sec-sev erity=Important,Moderate --bugfix - y`
	
	Für von Version 8 und 9 verwaltete Knoten lautet der entsprechende DNF-Befehl für diesen Workflow:
	
	`sudo dnf upgrade-minimal --security --sec-severity=Moderate --sec-sev erity=Important`

Sicherheitsoption	Patch-Auswahl

Benutzerdefinierte Patch-Baselines, bei denen das Kontrollkästchen Funktions updates einschließen aktiviert ist, mit einer SCHWEREGRAD-Liste von `[Critical, Important]` und einer KLASSIFIZIERUNG-Liste von `[Security, Bugfix]`	Neben den aus `updateinfo.xml` ausgewählten Sicherheits-Updates wendet Patch Manager auch nicht sicherheitsrelevante Updates an, die ansonsten den Patch-Filterregeln entsprechen.
	
	Für von Version 7 verwaltete Knoten lautet der entsprechende Yum-Befehl für diesen Workflow:
	
	`sudo yum update --security --sec-severity=Critical,Important --bugfix -y`
	
	Für von Version 8 und 9 verwaltete Knoten lautet der entsprechende DNF-Befehl für diesen Workflow:
	
	`sudo dnf upgrade --security --sec-severity=Critical, --sec-severity=Important --bugfix y`
	
	Note
	Neue Pakete, die veraltete Pakete mit anderen Namen ersetzen, werden installiert, wenn Sie diese `yum`- oder `dnf`-Befehle außerhalb von Patch Manager ausführen. Sie werden jedoch nicht mit den entsprechenden Patch Manager-Vorgängen installiert.

Weitere Informationen über Patch-Compliance-Statuswerte finden Sie unter Statuswerte der Patch-Compliance.

So funktionieren Patch-Basisregeln für AlmaLinuxRHEL, und Rocky Linux

Bei AlmaLinux, Red Hat Enterprise Linux (RHEL) und läuft Rocky Linux der Prozess der Patch-Auswahl wie folgt ab:

1. Auf dem verwalteten Knoten greift die YUM-Bibliothek (RHEL7) oder die DNF-Bibliothek (AlmaLinux 8 und 9, RHEL 8, 9 und 10 sowie Rocky Linux 8 und 9) auf die `updateinfo.xml` Datei für jedes konfigurierte Repository zu.

   > **Note**
   >
   > Die `updateinfo.xml`-Datei ist möglicherweise nicht verfügbar, wenn das Repo nicht von Red Hat verwaltet wird. Falls keine `updateinfo.xml` gefunden werden, wird kein Patch angewendet.

2. Jeder Update-Hinweis in `updateinfo.xml` enthält mehrere Attribute, die die Eigenschaften der Pakete im Hinweis kennzeichnen, wie in der folgenden Tabelle beschrieben.

   Update-Hinweis-Attribute

Attribut	Description
Typ	Entspricht dem Wert des Klassifizierungsschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Typ des im Update-Hinweis enthaltenen Pakets. Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation anzeigen. DescribePatchProperties Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.
severity	Entspricht dem Wert des Schweregradschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Schweregrad der

Attribut	Description
	im Update-Hinweis enthaltenen Pakete. Gilt in der Regel nur für Update-Hinweise im Hinblick auf die Sicherheit. Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation anzeigenDescribePatchProperties. Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.
update_id	Kennzeichnet die Advisory ID, wie etwa RHSA-2017:0864. Die Advisory ID kann in der Patch-Baseline im ApprovedPatches oder RejectedPatches-Attribut verwendet werden.
Referenzen	Enthält weitere Informationen über den Update-Hinweis, wie etwa eine CVE ID (Format: CVE-2017-1000371) oder eine Bugzilla ID (Format: 1463241). Die CVE ID und die Bugzilla ID können in der Patch-Baseline im ApprovedPatches- oder im RejectedPatches-Attribut verwendet werden.
Aktualisiert	Entspricht ApproveAfterDays in der Patch-Baseline. Kennzeichnet das Veröffentlichungsdatum (Aktualisierungsdatum) der im Update-Hinweis enthaltenen Pakete. Ein Vergleich zwischen dem aktuellen Zeitstempel und dem Wert dieses Attributs plus `ApproveAfterDays` wird verwendet, um zu bestimmen, ob der Patch für die Bereitstellung genehmigt wurde.

Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen.

3. Das Produkt des verwalteten Knotens wird durch SSM Agent bestimmt. Dieses Attribut entspricht dem Wert des Produktschlüsselattributs im PatchFilter-Datentyp der Patch-Baseline.
4. Pakete für das Update werden gemäß den folgenden Richtlinien ausgewählt.

Sicherheitsoption	Patch-Auswahl

Vordefinierte Standard-Patch-Baselines, die von AWS bereitgestellt werden, und benutzerdefinierte Patch-Baselines, bei denen das Kontrollkästchen Funktionsupdates einschließen nicht ausgewählt wurde	Für jeden Update-Hinweis in `updateinfo.xml` wird die Patch-Baseline als Filter verwendet, der nur den qualifizierten Paketen die Aufnahme in das Update erlaubt. Wenn mehrere Pakete zutreffen, wird die aktuelle Version nach Anwenden der Patch-Baseline-Definition verwendet.
	
	Für RHEL 7 lautet der entsprechende YUM-Befehl für diesen Workflow:
	
	`sudo yum update-minimal --security --severity=Critical,Important --bugfix -y`
	
	Für AlmaLinux 8 und 9, RHEL 8, 9 und 10 sowie Rocky Linux 8 und 9 lautet der entsprechende dnf-Befehl für diesen Workflow:
	
	`sudo dnf upgrade-minimal --security --severity=Critical --sec-severity=Important --bugfix -y`

Benutzerdefinierte Patch-Baselines, bei denen das Kontrollkästchen Funktionsupdates einschließen aktiviert ist, mit einer SCHWEREGRAD-Liste von `[Critical,`	Neben den aus `updateinfo.xml` ausgewählten Sicherheits-Updates wendet Patch Manager auch nicht sicherheitsrelevan

Sicherheitsoption	Patch-Auswahl

`Important]` und einer KLASSIFIZIERUNG-Liste von `[Security, Bugfix]`	te Updates an, die ansonsten den Patch-Fil terregeln entsprechen.
	
	Für RHEL 7 lautet der entsprechende YUM-Befehl für diesen Workflow:
	
	`sudo yum update --security --sec-sev erity=Critical,Important --bugfix - y`
	
	Für AlmaLinux 8 und 9, RHEL 8, 9 und 10 sowie Rocky Linux 8 und 9 lautet der entsprechende dnf-Befehl für diesen Workflow:
	
	`sudo dnf upgrade --sec-severity=Cri tical --sec-severity=Important -- bugfix -y`
	
	Note
	Neue Pakete, die veraltete Pakete mit anderen Namen ersetzen, werden installiert, wenn Sie diese `yum`- oder `dnf`-Befehle außerhalb von Patch Manager ausführen. Sie werden jedoch nicht mit den entsprechenden Patch Manager-Vorgängen installiert.

Weitere Informationen über Patch-Compliance-Statuswerte finden Sie unter Statuswerte der Patch-Compliance.

Funktionsweise von Patch-Baseline-Regeln auf SUSE Linux Enterprise Server

Auf SLES enthält jeder Patch die folgenden Attribute, mit denen die Eigenschaften der Pakete im Patch gekennzeichnet werden:

- Category: Entspricht dem Wert des Klassifizierungs-Schlüsselattributs im PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Typ des im Update-Hinweis enthaltenen Patches.

  Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation DescribePatchProperties anzeigen. Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.
- Schweregrad: Entspricht dem Wert des Schweregrads-Schlüsselattributs in dem PatchFilter-Datentyp der Patch-Baseline. Kennzeichnet den Schweregrad der Patches.

  Sie können die Liste der unterstützten Werte mithilfe des AWS CLI Befehls describe-patch-properties oder der API-Operation anzeigenDescribePatchProperties. Sie können die Liste auch im Bereich Genehmigungsregeln der Seite Erstellen einer Patch-Baseline der Seite Patch-Baseline bearbeiten in der Systems Manager-Konsole anzeigen.

Das Produkt des verwalteten Knotens wird durch SSM Agent bestimmt. Dieses Attribut entspricht dem Wert des Produkt-Schlüsselattributs im PatchFilter-Datentyp der Patch-Baseline.

Für jeden Patch wird die Patch-Baseline als Filter verwendet, der nur den qualifizierten Paketen die Aufnahme in das Update erlaubt. Wenn mehrere Pakete zutreffen, wird die aktuelle Version nach Anwenden der Patch-Baseline-Definition verwendet.

Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen.

Funktionsweise von Patch-Baseline-Regeln auf Ubuntu Server

Auf Ubuntu Server bietet der Patch-Baseline-Service Filter in den Feldern Priorität und Abschnitt. Diese Felder sind normalerweise für alle Ubuntu Server-Pakete vorhanden. Um zu bestimmen, ob ein Patch von der Patch-Baseline ausgewählt wird, geht Patch Manager folgendermaßen vor:

1. Auf Ubuntu Server-Systemen wird das Äquivalent von `sudo apt-get update` ausgeführt, um die Liste der verfügbaren Pakete zu aktualisieren. Repos sind nicht konfiguriert und die Daten werden aus Repos abgerufen, die in einer `sources`-Liste konfiguriert sind.

2. Wenn eine Aktualisierung für `python3-apt` (eine Python-Bibliotheks-Schnittstelle zu `libapt`) verfügbar ist, wird es auf die neueste Version aktualisiert. (Dieses nicht sicherheitsrelevante Paket wird aktualisiert, auch wenn Sie die Option Mit nicht sicherheitsrelevanten Updates nicht ausgewählt haben.)
3. Als Nächstes werden die Listen GlobalFilters, ApprovalRules, ApprovedPatches und RejectedPatches angewendet.

> **Note**
>
> Da es nicht möglich ist, die Veröffentlichungsdaten von Updatepaketen für Ubuntu Server zuverlässig zu bestimmen, werden die Optionen für die automatische Genehmigung für dieses Betriebssystem nicht unterstützt.

Genehmigungsregeln sind jedoch auch davon abhängig, ob das Kästchen Mit nicht sicherheitsrelevanten Updates beim Erstellen oder letzten Aktualisieren einer Patch-Baseline aktiviert wurde.

Wenn nicht sicherheitsrelevante Updates ausgeschlossen werden, wird eine implizite Regel angewendet, um nur Pakete mit Upgrades in Sicherheits-Repos auszuwählen. Für jedes Paket muss die Kandidatenversion des Pakets (in der Regel die neueste Version) Teil eines Sicherheits-Repos sein. Für Ubuntu Server sind Patch-Kandidaten-Versionen in diesem Fall auf Patches beschränkt, die in den folgenden Repos enthalten sind:

- Ubuntu Server 16.04 LTS: `xenial-security`
- Ubuntu Server 18.04 LTS: `bionic-security`
- Ubuntu Server 20.04 LTS: `focal-security`
- Ubuntu Server 22.04 LTS (`jammy-security`)
- Ubuntu Server 24.04 LTS (`noble-security`)
- Ubuntu Server 25.04 (`plucky-security`)

Wenn nicht sicherheitsrelevante Updates enthalten sind, werden auch Patches aus anderen Repositorys berücksichtigt.

Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen.

Zum Anzeigen der Inhalte der Felder Priorität und Abschnitt führen Sie den folgenden `aptitude`-Befehl aus:

> **Note**
>
> Möglicherweise müssen Sie zuerst Aptitude auf Ubuntu Server 16-Systemen installieren.

```
aptitude search -F '%p %P %s %t %V#' '~U'
```

In der Antwort auf diesen Befehl werden alle Pakete, für die ein Upgrade durchgeführt werden kann, in diesem Format gemeldet:

```
name, priority, section, archive, candidate version
```

Weitere Informationen über Patch-Compliance-Statuswerte finden Sie unter Statuswerte der Patch-Compliance.

Unterschiede beim Patchvorgang zwischen Linux und Windows Server

In diesem Thema werden wichtige Unterschiede zwischen Linux- und Windows Server-Patching in Patch Manager, einem Tool in AWS Systems Manager, beschrieben.

> **Note**
>
> Um von Linux verwaltete Knoten zu patchen, müssen Ihre Knoten SSM Agent der Version 2.0.834.0 oder höher ausführen.
> Wenn Systems Manager neue Tools hinzugefügt oder Aktualisierungen an den vorhandenen Tools vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Tools und Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent in Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die SSM Agent-Versionshinweise-Seite in GitHub, um Benachrichtigungen über SSM Agent-Updates zu erhalten.

Unterschied 1: Patch-Bewertung

Patch Manager verwendet auf Windows-verwalteten Knoten und Linux-verwalteten Knoten jeweils andere Prozesse, um zu ermitteln, welche Patches installiert sein sollten.

Linux

Bei Linux-Patches wertet Systems Manager auf jedem verwalteten Knoten einzeln zuerst die Patch-Baseline-Regeln und dann die Liste der genehmigten bzw. abgelehnten Patches aus. Systems Manager muss die Patches auf jedem Knoten gesondert auswerten, weil der Service die Liste der bekannten Patches und Updates von den Repositorys abruft, die auf dem verwalteten Knoten konfiguriert sind.

Windows

Für Windows-Patches wertet Systems Manager direkt im Service zuerst die Patch-Baseline-Regeln und dann die Liste der genehmigten bzw. abgelehnten Patches aus. Dies ist möglich, weil Windows-Patches aus einem einzigen Repository (Windows Update) abgerufen werden.

Unterschied 2: **`Not Applicable`**-Patches

Aufgrund der großen Anzahl der verfügbaren Pakete für Linux-Betriebssysteme, meldet Systems Manager keine Details zu Patches mit dem Status Nicht anwendbar. Ein `Not Applicable`-Patch ist beispielsweise ein Patch für Apache-Software, wenn auf der Instance Apache nicht installiert ist. Systems Manager meldet zwar die Anzahl der `Not Applicable` Patches in der Zusammenfassung, aber wenn Sie die DescribeInstancePatchesAPI für einen verwalteten Knoten aufrufen, enthalten die zurückgegebenen Daten keine Patches mit dem Status von`Not Applicable`. Dieses Verhalten unterscheidet sich von dem bei Windows.

Unterschied 3: Unterstützung von SSM-Dokumenten

Das Systems-Manager-Dokument (SSM-Dokument) `AWS-ApplyPatchBaseline` unterstützt keine Linux-verwalteten Knoten. Um Patch-Baselines auf Linux-, macOS- und Windows Server-verwalteten Knoten anzuwenden, wird das SSM-Dokument `AWS-RunPatchBaseline` empfohlen. Weitere Informationen erhalten Sie unter SSM-Befehlsdokumente zum Patchen verwalteter Knoten und SSM-Befehlsdokument zum Patchen: `AWS-RunPatchBaseline`.

Unterschied 4: Anwendungspatches

Der Hauptfokus von Patch Manager liegt auf dem Patchen von Betriebssystemen. Sie können mit Patch Manager jedoch auch Patches für bestimmte Anwendungen auf Ihren verwalteten Knoten anwenden.

Linux

Auf Linux-Betriebssystemen verwendet Patch Manager die konfigurierten Repositorys für Updates und unterscheidet dabei nicht zwischen Betriebssystem- und Anwendungs-Patches. Sie können mit Patch Manager festlegen, aus welchen Repositorys Updates abgerufen werden. Weitere Informationen finden Sie unter So geben Sie ein alternatives Patch-Quell-Repository an (Linux).

Windows

Auf von Windows Server verwaltete Knoten können Sie für Microsoft-Anwendungen wie Microsoft Word 2016 und Microsoft Exchange Server 2016 Genehmigungsregeln sowie die Patch-Ausnahmen Approved (Freigegeben) und Rejected (Abgelehnt) anwenden. Weitere Informationen finden Sie unter Arbeiten mit benutzerdefinierten Patch-Baselines.

Unterschied 5: Optionen für abgelehnte Patch-Listen in benutzerdefinierten Patch-Baselines

Wenn Sie eine benutzerdefinierte Patch-Baseline erstellen, können Sie für eine Liste Abgelehnte Patches einen oder mehrere Patches angeben. Bei verwalteten Linux-Knoten können Sie auch festlegen, dass sie installiert werden dürfen, wenn es sich um Abhängigkeiten für einen anderen Patch handelt, der laut Baseline zulässig ist.

Windows Server unterstützt jedoch das Konzept der Patch-Abhängigkeiten nicht. Sie können einen Patch zur Liste der abgelehnten Patches in einer benutzerdefinierten Baseline für Windows Server hinzufügen. Das Ergebnis hängt jedoch davon ab, (1) ob der abgelehnte Patch bereits auf einem verwalteten Knoten installiert ist oder nicht, und (2) welche Option Sie für die Aktion Abgelehnte Patches wählen.

In der folgenden Tabelle finden Sie Einzelheiten zu den Optionen für abgelehnte Patches in Windows Server.

Status der installation	Option: „Als Abhängigkeit zulassen“	Option: „Blockieren“
Der Patch ist bereits installiert	Gemeldeter Status: `INSTALLED_OTHER`	Gemeldeter Status: `INSTALLED_REJECTED`
Der Patch ist noch nicht installiert	Patch übersprungen	Patch übersprungen

Jeder von Microsoft veröffentlichte Patch für Windows Server enthält normalerweise alle Informationen, die für eine erfolgreiche Installation erforderlich sind. Gelegentlich kann jedoch ein Paket erforderlich sein, das Sie manuell installieren müssen. Patch Manager meldet keine Informationen zu diesen Voraussetzungen. Weitere Informationen finden Sie auf der Microsoft-Website unter Problembehandlung bei Windows Update.

SSM-Befehlsdokumente zum Patchen verwalteter Knoten

In diesem Thema werden die neun derzeit verfügbaren Systems-Manager-Dokumente (SSM-Dokumente) beschrieben, die Ihnen dabei helfen, Ihre verwalteten Knoten mit den neuesten sicherheitsrelevanten Updates zu patchen.

Wir empfehlen Ihnen, nur fünf dieser Dokumente für Ihre Patches zu verwenden. Zusammen bieten Ihnen diese fünf SSM-Dokumente eine breite Palette an Patch-Optionen mit AWS Systems Manager. Vier dieser Dokumente wurden später veröffentlicht als die vier alten SSM-Dokumente, die sie ersetzen und Erweiterungen oder Konsolidierungen der Funktion darstellen.

Empfohlene SSM-Dokumente zum Patchen

Wir empfehlen, bei Ihren Patchvorgängen die folgenden fünf SSM-Dokumente zu verwenden.

- `AWS-ConfigureWindowsUpdate`
- `AWS-InstallWindowsUpdates`
- `AWS-RunPatchBaseline`
- `AWS-RunPatchBaselineAssociation`
- `AWS-RunPatchBaselineWithHooks`

Ältere SSM-Dokumente zum Patchen

Die folgenden vier älteren SSM-Dokumente sind in einigen AWS-Regionen weiterhin verfügbar, werden jedoch nicht mehr aktualisiert oder unterstützt. Es kann nicht garantiert werden, dass diese Dokumente nur in IPv6 Umgebungen und im Support funktionieren. IPv4 Es kann nicht garantiert werden, dass sie in allen Szenarien funktionieren, und sie könnten in Zukunft den Support verlieren. Es wird empfohlen, diese Dokumente nicht für Patch-Vorgänge zu verwenden.

- `AWS-ApplyPatchBaseline`
- `AWS-FindWindowsUpdates`

- `AWS-InstallMissingWindowsUpdates`
- `AWS-InstallSpecificWindowsUpdates`

Anweisungen zum Einrichten von Patching-Vorgängen in einer Umgebung, die nur Support bietet, finden Sie unter IPv6. the section called “Tutorial: Einen Server in einer IPv6 einzigen Umgebung patchen”

In den folgenden Abschnitten finden Sie weitere Informationen zur Verwendung dieser SSM-Dokumente bei Ihren Patching-Vorgängen.

Themen

- Empfohlene SSM-Dokumente für das Patchen von verwalteten Knoten
- Legacy-SSM-Dokumente für das Patchen von verwalteten Knoten
- Bekannte Einschränkungen der SSM-Dokumente für das Patchen verwalteter Knoten
- SSM-Befehlsdokument zum Patchen: AWS-RunPatchBaseline
- SSM-Befehlsdokument zum Patchen: AWS-RunPatchBaselineAssociation
- SSM-Befehlsdokument zum Patchen: AWS-RunPatchBaselineWithHooks
- Beispielszenario für die Verwendung des InstallOverrideList Parameters in AWS-RunPatchBaseline oder AWS-RunPatchBaselineAssociation
- Verwenden des BaselineOverride -Parameters

Empfohlene SSM-Dokumente für das Patchen von verwalteten Knoten

Die folgenden fünf SSM-Dokumente werden für die Verwendung bei Ihren Patching-Operationen für Ihre verwalteten Knoten empfohlen.

Empfohlene SSM-Dokumente

- AWS-ConfigureWindowsUpdate
- AWS-InstallWindowsUpdates
- AWS-RunPatchBaseline
- AWS-RunPatchBaselineAssociation
- AWS-RunPatchBaselineWithHooks

AWS-ConfigureWindowsUpdate

Unterstützt die Konfiguration grundlegender Funktionen für Windows Update und deren Verwendung zum automatischen Installieren von Updates (oder zum Deaktivieren automatischer Updates). In allen AWS-Regionen verfügbar.

Mit diesem SSM-Dokument wird Windows Update aufgefordert, die angegebenen Updates herunterzuladen und zu installieren und die verwalteten Knoten bei Bedarf neu zu starten. Verwenden Sie dieses Dokument zusammen mit einem Tool inState Manager, um sicherzustellen AWS Systems Manager, dass Windows Update seine Konfiguration beibehält. Sie können es auch manuell ausführenRun Command, indem Sie ein Tool in verwenden AWS Systems Manager, um die Windows Update-Konfiguration zu ändern.

Die in diesem Dokument verfügbaren Parameter unterstützen die Angabe einer Kategorie von Updates, die installiert werden sollen (oder ob automatische Updates deaktiviert werden sollen), sowie die Angabe des Wochentages und der Tageszeit für die Ausführung von Patch-Vorgängen. Dieses SSM-Dokument ist besonders dann von Vorteil, wenn Sie keine strenge Kontrolle über Windows Updates benötigen und keine Compliance-Informationen sammeln müssen.

Replaces legacy SSM documents:

- Keine

AWS-InstallWindowsUpdates

Installiert Updates auf einem von Windows Server verwalteten Knoten. In allen AWS-Regionen verfügbar.

Dieses SSM-Dokument bietet grundlegende Patch-Funktion für den Fall, dass Sie entweder ein bestimmtes Update (mit Hilfe des `Include Kbs`-Parameters) installieren möchten oder Patches mit bestimmten Klassifizierungen oder Kategorien installieren möchten, aber keine Informationen zur Patch-Compliance benötigen.

Replaces legacy SSM documents:

- `AWS-FindWindowsUpdates`
- `AWS-InstallMissingWindowsUpdates`
- `AWS-InstallSpecificWindowsUpdates`

Die drei alten Dokumente erfüllen zwar unterschiedliche Funktionen, aber Sie können die gleichen Ergebnisse erzielen, indem Sie unterschiedliche Parametereinstellungen mit dem neueren SSM-Dokument `AWS-InstallWindowsUpdates` verwenden. Diese Parametereinstellungen werden in Legacy-SSM-Dokumente für das Patchen von verwalteten Knoten beschrieben.

`AWS-RunPatchBaseline`

Installiert Patches auf Ihren verwalteten Knoten oder scannt Knoten, um festzustellen, ob qualifizierte Patches fehlen. In allen AWS-Regionen verfügbar.

Mit `AWS-RunPatchBaseline` können Sie Patch-Genehmigungen mithilfe der Patch-Baseline steuern, die als „Standard“ für einen Betriebssystemtyp angegeben ist. Stellt Informationen zur Patch-Compliance dar, die Sie mit den Systems Manager-Compliance-Tools einsehen können. Mit diesen Tools erhalten Sie Erkenntnisse in den Zustand der Patch-Compliance Ihrer verwalteten Knoten, z. B. bei welchen Knoten Patches fehlen und was diese Patches sind. Wenn Sie `AWS-RunPatchBaseline` verwenden, werden Patch-Compliance-Informationen mit dem API-Befehl `PutInventory` aufgezeichnet. Für Linux-Betriebssysteme werden Compliance-Informationen für Patches sowohl über das in einem verwalteten Knoten konfigurierte Standard-Quell-Repository bereitgestellt, als auch von einem beliebigen alternativen Quell-Repository aus, das Sie in einer benutzerdefinierten Patch-Baseline angeben. Weitere Informationen über alternative Quell-Repositorys finden Sie unter So geben Sie ein alternatives Patch-Quell-Repository an (Linux). Weitere Informationen zu den Systems Manager-Compliance-Tools finden Sie unter AWS Systems Manager-Compliance.

Ersetzt alte Dokumente:

- `AWS-ApplyPatchBaseline`

Das Legacy-Dokument `AWS-ApplyPatchBaseline` gilt nur für von Windows Server verwaltete Knoten und bietet keinen Support für Anwendungs-Patches. Das neue Dokument `AWS-RunPatchBaseline` bietet die gleiche Unterstützung für sowohl Windows- als auch Linux-Systeme. Version 2.0.834.0 oder höher von SSM Agent ist erforderlich, um das Dokument `AWS-RunPatchBaseline` zu verwenden.

Weitere Informationen über das SSM-Dokument `AWS-RunPatchBaseline` finden Sie unter SSM-Befehlsdokument zum Patchen: `AWS-RunPatchBaseline`.

AWS-RunPatchBaselineAssociation

Installiert Patches auf Ihren Instances oder scannt Instances, um festzustellen, ob qualifizierte Patches fehlen. In allen kommerziellen AWS-Regionen verfügbar.

`AWS-RunPatchBaselineAssociation` unterscheidet sich in verschiedenen wichtigen Punkten von `AWS-RunPatchBaseline`:

- `AWS-RunPatchBaselineAssociation` ist hauptsächlich für die Verwendung mit State Manager-Zuordnungen vorgesehen, die mithilfe von Quick Setup, einem Tool in AWS Systems Manager, erstellt wurden. Insbesondere, wenn Sie den Quick Setup-Host-Management-Konfigurationstyp verwenden und die Option Scan instances for missing patches daily (Instances täglich auf fehlende Patches scannen) auswählen, verwendet das System `AWS-RunPatchBaselineAssociation` für diese Operation.

  In den meisten Fällen sollten Sie jedoch beim Einrichten eigener Patching-Vorgänge `AWS-RunPatchBaseline` oder `AWS-RunPatchBaselineWithHooks` anstelle von `AWS-RunPatchBaselineAssociation` auswählen.

  Weitere Informationen finden Sie unter den folgenden Themen:

  - AWS Systems Manager Quick Setup
  - SSM-Befehlsdokument zum Patchen: `AWS-RunPatchBaselineAssociation`
- `AWS-RunPatchBaselineAssociation` unterstützt die Verwendung von Tags, um zu identifizieren, welche Patch-Baseline bei der Ausführung mit einer Reihe von Zielen verwendet werden soll.
- Für Patching-Vorgänge, die `AWS-RunPatchBaselineAssociation` verwenden, werden Patch-Compliance-Daten in Bezug auf eine bestimmte State Manager-Zuordnung gesammelt. Die Patch-Compliance-Daten, die gesammelt werden, wenn `AWS-RunPatchBaselineAssociation` ausgeführt wird, werden mit dem API-Befehl `PutComplianceItems` anstelle des Befehls `PutInventory` aufgezeichnet. Dies verhindert, dass Compliance-Daten, die nicht mit dieser bestimmten Zuordnung verknüpft sind, überschrieben werden.

  Für Linux-Betriebssysteme werden Compliance-Informationen für Patches sowohl über das in einer Instance konfigurierte Standard-Quell-Repository bereitgestellt, als auch von einem beliebigen alternativen Quell-Repository aus, das Sie in einer benutzerdefinierten Patch-Baseline angeben. Weitere Informationen über alternative Quell-Repositorys finden Sie unter So geben Sie ein alternatives Patch-Quell-Repository an (Linux). Weitere Informationen zu den Systems Manager-Compliance-Tools finden Sie unter AWS Systems Manager-Compliance.

Ersetzt alte Dokumente:

- Keine

Weitere Informationen über das SSM-Dokument `AWS-RunPatchBaselineAssociation` finden Sie unter SSM-Befehlsdokument zum Patchen: `AWS-RunPatchBaselineAssociation`.

`AWS-RunPatchBaselineWithHooks`

Installiert Patches auf Ihren verwalteten Knoten oder scannt Knoten, um festzustellen, ob qualifizierte Patches fehlen. Mit optionalen Hooks können Sie SSM-Dokumente an drei Punkten während des Patch-Zyklus ausführen. In allen kommerziellen AWS-Regionen verfügbar. Wird nicht unterstützt auf macOS.

`AWS-RunPatchBaselineWithHooks` unterscheidet sich von `AWS-RunPatchBaseline` in seiner `Install`-Operation.

`AWS-RunPatchBaselineWithHooks` unterstützt Lebenszyklus-Hooks, die während dem Patching von verwalteten Knoten an festgelegten Punkten ausgeführt werden. Da Patch-Installationen manchmal den Neustart von verwalteten Knoten erfordern, ist die Patch-Operation in zwei Ereignisse unterteilt, wobei insgesamt drei Hooks enthalten sind, die benutzerdefinierte Funktion unterstützen. Der erste Hook ist vor der `Install with NoReboot`-Operation. Der zweite Hook ist nach der `Install with NoReboot`-Operation. Der dritte Hook ist nach dem Neustart des Knoten verfügbar.

Ersetzt alte Dokumente:

- Keine

Weitere Informationen über das SSM-Dokument `AWS-RunPatchBaselineWithHooks` finden Sie unter SSM-Befehlsdokument zum Patchen: `AWS-RunPatchBaselineWithHooks`.

Legacy-SSM-Dokumente für das Patchen von verwalteten Knoten

Die folgenden vier SSM-Dokumente sind in einigen AWS-Regionen noch verfügbar. Sie werden jedoch nicht mehr aktualisiert und möglicherweise in Zukunft nicht mehr unterstützt. Daher empfehlen wir ihre Verwendung nicht. Stattdessen verwenden Sie bitte die unter Empfohlene SSM-Dokumente für das Patchen von verwalteten Knoten beschriebenen Dokumente.

Alte SSM-Dokumente

- AWS-ApplyPatchBaseline

- AWS-FindWindowsUpdates
- AWS-InstallMissingWindowsUpdates
- AWS-InstallSpecificWindowsUpdates

AWS-ApplyPatchBaseline

Unterstützt nur von Windows Server verwaltete Knoten, enthält jedoch, anders als der Nachfolger `AWS-RunPatchBaseline`, keinen Support für Anwendungs-Patches. Nicht verfügbar in Versionen, die nach August 2017 AWS-Regionen veröffentlicht wurden.

> **Note**
>
> Um dieses SSM-Dokument, `AWS-RunPatchBaseline`, zu ersetzen, wird die Version 2.0.834.0 oder eine neuere Version von SSM Agent benötigt. Sie können das Dokument `AWS-UpdateSSMAgent` verwenden, um Ihre verwalteten Knoten auf die neueste Version des Agenten zu aktualisieren.

AWS-FindWindowsUpdates

Ersetzt durch `AWS-InstallWindowsUpdates`, die alle die gleichen Aktionen ausführen können. Nicht verfügbar bei AWS-Regionen Markteinführungen nach April 2017.

Um das gleiche Ergebnis wie bei diesem alten SSM-Dokument zu erzielen, verwenden Sie die folgende Parameterkonfiguration mit dem empfohlenen Ersatzdokument, `AWS-InstallWindowsUpdates`:

- `Action` = `Scan`
- `Allow Reboot` = `False`

AWS-InstallMissingWindowsUpdates

Ersetzt durch `AWS-InstallWindowsUpdates`, die alle die gleichen Aktionen ausführen können. Nicht verfügbar bei Produkten, die nach April 2017 auf den AWS-Regionen Markt gebracht wurden.

Um das gleiche Ergebnis wie bei diesem alten SSM-Dokument zu erzielen, verwenden Sie die folgende Parameterkonfiguration mit dem empfohlenen Ersatzdokument, `AWS-InstallWindowsUpdates`:

- `Action` = `Install`
- `Allow Reboot` = `True`

`AWS-InstallSpecificWindowsUpdates`

Ersetzt durch `AWS-InstallWindowsUpdates`, die alle die gleichen Aktionen ausführen können. Nicht verfügbar bei Produkten, die nach April 2017 auf den AWS-Regionen Markt gebracht wurden.

Um das gleiche Ergebnis wie bei diesem alten SSM-Dokument zu erzielen, verwenden Sie die folgende Parameterkonfiguration mit dem empfohlenen Ersatzdokument, `AWS-InstallWindowsUpdates`:

- `Action` = `Install`
- `Allow Reboot` = `True`
- `Include Kbs` = *`comma-separated list of KB articles`*

Bekannte Einschränkungen der SSM-Dokumente für das Patchen verwalteter Knoten

Unterbrechungen des externen Neustarts

Wenn das System auf dem Knoten während der Patch-Installation einen Neustart initiiert (z. B. um Firmware-Updates oder andere Funktionen anzuwenden SecureBoot), kann die Ausführung des Patch-Dokuments unterbrochen und als fehlgeschlagen markiert werden, obwohl Patches erfolgreich installiert wurden. Dies ist darauf zurückzuführen, dass der SSM-Agent den Status der Dokumentausführung bei externen Neustarts nicht beibehalten und wieder aufnehmen kann.

Um den Status der Patch-Installation nach einer fehlgeschlagenen Ausführung zu überprüfen, führen Sie einen `Scan` Patch-Vorgang aus und überprüfen Sie anschließend die Patch-Kompatibilitätsdaten in Patch Manager, um den aktuellen Kompatibilitätsstatus zu beurteilen.

SSM-Befehlsdokument zum Patchen: **`AWS-RunPatchBaseline`**

AWS Systems Manager unterstütztAWS-RunPatchBaseline, ein Systems Manager Manager-Dokument (SSM-Dokument) fürPatch Manager, ein Tool in AWS Systems Manager. Dieses SSM-Dokument führt Patch-Operationen auf verwaltete Knoten sowohl für sicherheitsrelevante als auch für andere Arten von Updates durch. Wenn das Dokument ausgeführt wird, verwendet es die Patch-Baseline, die der „Standard“ für einen Betriebssystemtyp ist, wenn keine Patch-Gruppe angegeben ist. Andernfalls wird die Patch-Baseline verwendet, die der Patch-Gruppe zugeordnet ist. Informationen zu Patch-Gruppen finden Sie unter Patch-Gruppen.

Sie können das Dokument `AWS-RunPatchBaseline` verwenden, um Patches sowohl für Betriebssysteme als auch für Anwendungen durchzuführen. (Unter Windows Server ist der Anwendungssupport auf Updates für Microsoft-Anwendungen beschränkt.)

Dieses Dokument unterstützt von Linux, macOS und Windows Server verwaltete Knoten. Das Dokument führt die entsprechenden Aktionen für jede Plattform durch.

> **Note**
>
> Patch Manager unterstützt auch das veraltete SSM-Dokument `AWS-ApplyPatchBaseline`. Dieses Dokument unterstützt jedoch nur das Patchen von Windows-verwalteten Knoten. Wir empfehlen Ihnen, stattdessen `AWS-RunPatchBaseline` zu verwenden, da es das Patchen auf Linux-, macOS- und Windows Server-verwaltete Knoten unterstützt. Version 2.0.834.0 oder höher von SSM Agent ist erforderlich, um das Dokument `AWS-RunPatchBaseline` zu verwenden.

Windows Server

Auf Windows Server verwalteten Knoten lädt das `AWS-RunPatchBaseline` Dokument ein PowerShell Modul herunter und ruft es auf, das wiederum einen Snapshot der Patch-Baseline herunterlädt, die für den verwalteten Knoten gilt. Dieser Patch-Baseline-Snapshot enthält eine Liste genehmigter Patches, die kompiliert werden, indem die Patch-Baseline auf einem WSUS-Server (Windows Server Update Services) abgefragt wird. Diese Liste wird an die Windows Update-API weitergeleitet, die das Herunterladen und Installieren des genehmigten Patches entsprechend steuert.

Linux

Auf Linux-verwalteten Knoten ruft das Dokument `AWS-RunPatchBaseline` ein Python-Modul auf, das wiederum einen entsprechenden Snapshot der Patch-Baseline für den verwalteten Knoten herunterlädt. Dieser Patch-Baseline-Snapshot verwendet die definierten Regeln und Listen der genehmigten und gesperrten Patches, um den entsprechenden Paketmanager für jeden Knoten-Typ anzutreiben:

- Von Amazon Linux 2, Oracle Linux und RHEL 7 verwaltete Knoten verwenden YUM. Für YUM-Operationen ist eine `Python 2.6` oder eine neuere unterstützte Version Patch Manager erforderlich (2.6 — 3.12). Amazon Linux 2023 verwendet DNF. Für DNF-Operationen Patch Manager ist eine unterstützte Version von `Python 2` oder `Python 3` (2.6 - 3.12) erforderlich.

- Von RHEL 8 verwaltete Knoten verwenden DNF. Für DNF-Operationen Patch Manager ist eine unterstützte Version von `Python 2` or `Python 3` (2.6 - 3.12) erforderlich. (Keine der beiden Versionen ist standardmäßig auf RHEL 8 installiert. Sie müssen die eine oder andere Version manuell installieren.)
- Debian Server, und Ubuntu Server Instanzen verwenden APT. Für APT-Operationen Patch Manager ist eine unterstützte Version von `Python 3` (3.0 - 3.12) erforderlich.

macOS

Auf macOS-verwalteten Knoten ruft das Dokument `AWS-RunPatchBaseline` ein Python-Modul auf, das wiederum einen entsprechenden Snapshot der Patch-Baseline für den verwalteten Knoten herunterlädt. Als Nächstes ruft ein Python-Unterprozess die AWS Command Line Interface (AWS CLI) auf dem Knoten auf, um die Installations- und Aktualisierungsinformationen für die angegebenen Paketmanager abzurufen und den entsprechenden Paketmanager für jedes Aktualisierungspaket zu steuern.

Jeder Snapshot ist spezifisch für eine Patchgruppe AWS-Konto, ein Betriebssystem und eine Snapshot-ID. Der Snapshot wird über eine vorsignierte Amazon Simple Storage Service (Amazon S3)-URL bereitgestellt, die 24 Stunden nach Erstellung des Snapshots abläuft. Wenn Sie jedoch denselben Snapshot-Inhalt auf andere verwaltete Knoten anwenden möchten, können Sie nach Ablauf der URL bis zu drei Tage nach Erstellung des Snapshots eine neue vorsignierte Amazon-S3-URL generieren. Verwenden Sie dazu den Befehl get-deployable-patch-snapshot-for-instance.

Nachdem alle genehmigten und zutreffenden Updates installiert und je nach Bedarf Neustarts durchgeführt wurden, werden Patch-Compliance-Informationen auf einem verwalteten Knoten generiert und wieder an Patch Manager gemeldet.

> **Note**
>
> Wenn der Parameter `RebootOption` im Dokument `AWS-RunPatchBaseline` auf `NoReboot` gesetzt ist, wird der verwaltete Knoten nach dem Ausführen von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.

Weitere Informationen zum Anzeigen von Patch-Compliance-Daten finden Sie unter Info zu Patch Compliance.

AWS-RunPatchBaseline-Parameter

AWS-RunPatchBaseline unterstützt sechs Parameter. Der Parameter Operation muss angegeben werden. Die Parameter InstallOverrideList, BaselineOverride und RebootOption sind optional. Snapshot-ID ist technisch optional, wir empfehlen allerdings, dass Sie einen benutzerdefinierten Wert dafür angeben, wenn Sie AWS-RunPatchBaseline außerhalb von einem Wartungsfenster ausführen, und Patch Manager den Wert benutzerdefinierten automatisch angeben lassen, wenn das Dokument als Teil eines Wartungsfenstervorgangs ausgeführt wird.

Parameter

- Parametername: Operation
- Parametername: AssociationId
- Parametername: Snapshot ID
- Parametername: InstallOverrideList
- Parametername: RebootOption
- Parametername: BaselineOverride
- Parametername: StepTimeoutSeconds

Parametername: **Operation**

Nutzung: erforderlich.

Optionen: Scan | Install.

Scan

Wenn Sie die Option Scan wählen, bestimmt AWS-RunPatchBaseline den Patch-Compliance-Status des verwalteten Knoten und meldet diese Informationen an Patch Manager. Scan fordert nicht zum Installieren von Updates oder zum Neustarten von verwalteten Knoten auf. Stattdessen erkennt die Operation, wo für den Knoten genehmigte und geeignete Updates fehlen.

Installieren

Bei Auswahl der Option Install versucht AWS-RunPatchBaseline, die genehmigten und geeigneten Updates zu installieren, die auf dem verwalteten Knoten fehlen. Patch-Compliance-Informationen, die als Teil eines Install-Vorgangs generiert wurden, enthalten keine fehlenden Updates, melden allerdings möglicherweise Updates im Fehlerzustand, wenn die Installation

des Updates aus einem beliebigen Grund nicht erfolgreich war. Immer wenn ein Update auf einem verwalteten Knoten installiert wird, wird der Knoten neu gestartet, um sicherzustellen, dass das Update installiert und aktiviert ist. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

> **Note**
>
> Wenn ein von den Basisregeln festgelegter Patch installiert wird, bevor der Patch Manager den verwalteten Knoten aktualisiert, wird das System möglicherweise nicht wie erwartet neu gestartet. Dies kann passieren, wenn ein Patch manuell von einem Benutzer oder automatisch von einem anderen Programm, z. B. dem `unattended-upgrades`-Paket auf Ubuntu Server, installiert wird.

Parametername: **`AssociationId`**

Nutzung: optional.

`AssociationId` ist die ID einer Zuordnung in State Manager, einem Tool in AWS Systems Manager. Es wird von Patch Manager verwendet, um einer angegebenen Zuordnung Compliance-Daten hinzuzufügen. Diese Zuordnung bezieht sich auf einen Patching-Vorgang, der in einer Patch-Richtlinie in Quick Setup eingerichtet ist.

> **Note**
>
> Wenn mit der `AWS-RunPatchBaseline` ein `AssociationId`-Wert zusammen mit einer Baseline-Überschreibung der Patch-Richtlinie bereitgestellt wird, wird das Patchen als eine `PatchPolicy`-Operation durchgeführt und der `ExecutionType`-Wert, der in `AWS:ComplianceItem` gemeldet wird, ist ebenfalls `PatchPolicy`. Wenn kein `AssociationId`-Wert angegeben wird, wird das Patchen als eine `Command`-Operation durchgeführt, und der `ExecutionType`-Wert, der in `AWS:ComplianceItem` übermittelt wird, ist ebenfalls `Command`.

Wenn Sie noch keine Zuordnung erstellt haben, die Sie verwenden möchten, können Sie eine erstellen, indem Sie den Befehl create-association ausführen.

Parametername: **`Snapshot ID`**

Nutzung: optional.

`Snapshot ID` ist eine eindeutige ID (GUID), die von Patch Manager verwendet wird, um sicherzustellen, dass ein Satz von verwalteten Knoten, für die in einer einzelnen Operation Patches durchgeführt werden, den genau gleichen Satz genehmigter Patches aufweist. Auch wenn der Parameter als optional definiert ist, hängen unsere Empfehlungen für bewährte Methoden davon ab, ob Sie `AWS-RunPatchBaseline` in einem Wartungsfenster, wie in der folgenden Tabelle beschrieben, ausführen.

Bewährte Methoden für **`AWS-RunPatchBaseline`**

Mode	Bewährte Methode	Details
Ausführen von `AWS-RunPatchBaseline` innerhalb eines Wartungsfensters	Geben Sie keine Snapshot ID an. Patch Manager wird sie für Sie bereitstellen.	Falls Sie ein Wartungsfenster zum Ausführen von `AWS-RunPatchBaseline` verwenden, dürfen Sie Ihre eigene generierte Snapshot ID nicht angeben. In diesem Szenario stellt Systems Manager einen GUID-Wert auf Grundlage der Wartungsfensterausführungs-ID bereit. Auf diese Weise wird sichergestellt, dass eine richtige ID für alle Aufrufe von `AWS-RunPatchBaseline` in diesem Wartungsfenster verwendet wird. Wenn Sie einen Wert in diesem Szenario angeben, beachten Sie, dass der Snapshot der Patch-Baseline möglicherweise nicht länger als drei Tagen erhalten bleibt. Danach wird ein neuer

Mode	Bewährte Methode	Details
		Snapshot erstellt, auch wenn Sie dieselbe ID angeben, nachdem der Snapshot abgelaufen ist.

Mode	Bewährte Methode	Details
Ausführen von `AWS-RunPatchBaseline` außerhalb eines Wartungsfensters	Generieren Sie einen benutzerdefinierten GUID-Wert für die Snapshot-ID und geben Sie ihn an.[1]	Wenn Sie kein Wartungsfenster zum Ausführen von `AWS-RunPatchBaseline` verwenden, empfehlen wir, dass Sie eine eindeutige Snapshot-ID für jede Patch-Baseline generieren und angeben, insbesondere wenn Sie das Dokument `AWS-RunPatchBaseline` auf mehreren verwaltete Knoten in derselben Operation ausführen. Wenn Sie keine ID in diesem Szenario angeben, generiert Systems Manager eine andere Snapshot-ID für jeden verwalteten Knoten, an den der Befehl gesendet wird. Dies kann zu unterschiedlichen Sätzen von Patches führen, die auf den jeweiligen verwalteten Knoten angegeben sind. Angenommen, Sie führen das Dokument `AWS-RunPatchBaseline` direkt über Run Command aus, ein Tool in AWS Systems Manager, und richten es auf eine Gruppe von 50 verwaltet en Knoten aus. Das Angeben einer benutzerdefinierten Snapshot-ID führt zur Erstellung eines einzelnen Baseline-

Mode	Bewährte Methode	Details
		Snapshots, der verwendet wird, um alle Knoten zu bewerten und zu patchen. Dadurch wird gewährlei stet, dass sie letztendlich einen konsistenten Zustand aufweisen.

[1] Sie können jedes beliebige Tool zum Generieren eines Werts für den Snapshot-ID-Parameter verwenden, das eine GUID generieren kann. In können Sie PowerShell beispielsweise das `New-Guid` Cmdlet verwenden, um eine GUID im Format von zu generieren. `12345699-9405-4f69-bc5e-9315aEXAMPLE`

Parametername: **`InstallOverrideList`**

Nutzung: optional.

Mit `InstallOverrideList` können Sie eine https-URL oder eine Amazon S3-PathStyle-URL zu einer Liste mit zu installierenden Patches angeben. Diese im YAML-Format geführte Patch-Installationsliste überschreibt die von der Standard-Patch-Baseline angegebenen Patches. Dies bietet Ihnen eine detalliertere Kontrolle darüber, welche Patches auf Ihren verwalteten Knoten installiert sind.

> **Important**
>
> Der `InstallOverrideList`-Dateiname darf die folgenden Zeichen nicht enthalten: Backtick (`), einfaches Anführungszeichen ('), doppeltes Anführungszeichen („) und Dollarzeichen ($).

Das Verhalten des Patch-Vorgangs bei Verwendung des `InstallOverrideList`-Parameters unterscheidet sich zwischen Linux- und macOS-verwalteten Knoten und Windows Server-verwalteten Knoten. Unter Linux und macOS versucht Patch Manager, Patches aus der Patchliste `InstallOverrideList` anzuwenden, die in einem auf dem Knoten aktivierten Repository vorhanden sind, unabhängig davon, ob die Patches den Patch-Baseline-Regeln entsprechen oder

nicht. Auf Windows Server-Knoten werden Patches in der `InstallOverrideList`-Patch-Liste jedoch nur angewendet, wenn sie auch den Patch-Baseline-Regeln entsprechen.

Beachten Sie, dass Compliance-Berichte Patch-Status entsprechend den Angaben in der Patch-Baseline wiedergeben, nicht entsprechend Ihren Angaben in einer `InstallOverrideList`-Liste von Patches. Mit anderen Worten: Scan-Operationen ignorieren den Parameter `InstallOverrideList`. Auf diese Weise wird sichergestellt, dass Compliance-Berichte den Patch-Status konsistent entsprechend der Richtlinie wiedergeben und nicht danach, was für eine bestimmte Patching-Operation genehmigt wurde.

> **Note**
>
> Wenn Sie einen Knoten patchen, der nur verwendet, stellen Sie sicher IPv6, dass die angegebene URL vom Knoten aus erreichbar ist. Wenn die SSM Agent Konfigurationsoption auf gesetzt `UseDualStackEndpoint` ist`true`, wird ein Dual-Stack-S3-Client verwendet, wenn eine S3-URL bereitgestellt wird. Weitere Informationen the section called “Tutorial: Einen Server in einer IPv6 einzigen Umgebung patchen” zur Konfiguration des Agenten für die Verwendung von Dualstack finden Sie unter.

Eine Beschreibung, wie Sie den Parameter `InstallOverrideList` verwenden können, um verschiedene Patch-Typen in verschiedenen Wartungsfenster-Zeitplänen auf eine Zielgruppe anzuwenden und gleichzeitig eine einzelne Patch-Baseline zu verwenden, finden Sie unter Beispielszenario für die Verwendung des InstallOverrideList Parameters in `AWS-RunPatchBaseline` oder `AWS-RunPatchBaselineAssociation`.

Gültige URL-Formate

> **Note**
>
> Wenn Ihre Datei in einem öffentlich zugänglichen Bucket gespeichert ist, können Sie entweder ein HTTPS-URL-Format oder eine URL im Amazon S3-Pfadstil angeben. Wenn Ihre Datei in einem privaten Bucket gespeichert ist, müssen Sie eine URL im Amazon S3-Pfadstil angeben.

- HTTPS-URL-Format:

```
https://s3.aws-api-domain/amzn-s3-demo-bucket/my-windows-override-list.yaml
```

- URL im Amazon S3-Pfadstil:

```
s3://amzn-s3-demo-bucket/my-windows-override-list.yaml
```

Gültige YAML-Inhaltsformate

Die Formate, die Sie verwenden, um Patches in Ihrer Liste anzugeben, hängen von dem Betriebssystem Ihres verwalteten Knoten ab. Das allgemeine Format lautet jedoch folgendermaßen:

```
patches:
    -
        id: '{patch-d}'
        title: '{patch-title}'
        {additional-fields}:{values}
```

Sie können zwar zusätzliche Felder in der YAML-Datei bereitstellen, diese werden jedoch während der Patch-Operationen ignoriert.

Darüber hinaus empfehlen wir zu überprüfen, ob das Format Ihrer YAML-Datei gültig ist, bevor Sie die Liste in Ihrem S3-Bucket hinzufügen oder aktualisieren. Weitere Informationen zum YAML-Format finden Sie unter yaml.org. Für Validierungstool-Optionen suchen Sie im Internet nach „yaml format validators“ durch.

Linux

id

Das Feld id ist ein Pflichtfeld. Verwenden Sie es, um Patches mit Paketnamen und Architektur anzugeben. Beispiel: `'dhclient.x86_64'`. Sie können Platzhalter in der ID verwenden, um mehrere Pakete anzugeben. Zum Beispiel `'dhcp*'` und `'dhcp*1.*'`.

Title

Das Feld Titel ist optional, es bietet jedoch auf Linux-Systemen zusätzliche Filterfunktionen. Wenn Sie Titel verwenden, sollte er die Versionsinformationen des Pakets in einem der folgenden Formate enthalten:

YUM/SUSE Linux Enterprise Server (SLES):

```
{name}.{architecture}:{epoch}:{version}-{release}
```

APT

```
{name}.{architecture}:{version}
```

Für Linux-Patch-Titel können Sie einen oder mehrere Platzhalter in beliebigen Positionen verwenden, um die Anzahl der Paketzuordnungen zu erhöhen. Beispiel: `'*32:9.8.2-0.*.rc1.57.amzn1'`.

Zum Beispiel:

- apt-Paketversion 1.2.25 ist derzeit auf Ihrem verwalteten Knoten installiert, aber Version 1.2.27 ist jetzt verfügbar.
- Fügen Sie die apt.amd64-Version 1.2.27 der Liste hinzu. Sie ist abhängig von apt utils.amd64 Version 1.2.27, aber apt-utils.amd64 Version 1.2.25 ist in der Liste angegeben.

In diesem Fall wird die Installation der APT-Version 1.2.27 blockiert und als „Fehlgeschlagen-“ gemeldet. NonCompliant

Windows Server

id

Das Feld id ist ein Pflichtfeld. Verwenden Sie es, um Patches mithilfe von Microsoft Knowledge Base IDs (z. B. KB2736693) und Microsoft Security Bulletin IDs (z. B. MS17 -023) anzugeben.

Alle anderen Felder, die Sie in einer Patch-Liste für Windows bereitstellen möchten, sind optional und dienen nur zu Ihrer eigenen Information. Sie können zusätzlichen Felder verwenden, wie z. B. Titel, Klassifizierung, Schweregrad oder andere Angaben für detailliertere Informationen über die angegebenen Patches.

macOS

id

Das Feld id ist ein Pflichtfeld. Der Wert für das Feld id kann entweder mit einem `{package-name}.{package-version}`-Format oder einem {package_name}-Format bereitgestellt werden.

Patch-Beispiellisten

- Amazon Linux 2

```
patches:
    -
        id: 'kernel.x86_64'
    -
        id: 'bind*.x86_64'
        title: '39.11.4-26.P2.amzn2.5.2'

        id: 'glibc*'
    -
        id: 'dhclient*'
        title: '*4.2.5-58.amzn2'
    -
        id: 'dhcp*'
        title: '*4.2.5-77.amzn2'
```

- Debian Server

```
patches:
    -
        id: 'apparmor.amd64'
        title: '2.10.95-0ubuntu2.9'
    -
        id: 'cryptsetup.amd64'
        title: '*2:1.6.6-5ubuntu2.1'
    -
        id: 'cryptsetup-bin.*'
        title: '*2:1.6.6-5ubuntu2.1'
    -
        id: 'apt.amd64'
        title: '*1.2.27'
    -
        id: 'apt-utils.amd64'
        title: '*1.2.25'
```

- macOS

```
patches:
    -
        id: 'XProtectPlistConfigData'
```

```
    -
        id: 'MRTConfigData.1.61'
    -
        id: 'Command Line Tools for Xcode.11.5'
    -
        id: 'Gatekeeper Configuration Data'
```

- Oracle Linux

```
patches:
    -
        id: 'audit-libs.x86_64'
        title: '*2.8.5-4.el7'
    -
        id: 'curl.x86_64'
        title: '*.el7'
    -
        id: 'grub2.x86_64'
        title: 'grub2.x86_64:1:2.02-0.81.0.1.el7'
    -
        id: 'grub2.x86_64'
        title: 'grub2.x86_64:1:*-0.81.0.1.el7'
```

- Red Hat Enterprise Linux (RHEL)

```
patches:
    -
        id: 'NetworkManager.x86_64'
        title: '*1:1.10.2-14.el7_5'
    -
        id: 'NetworkManager-*.x86_64'
        title: '*1:1.10.2-14.el7_5'
    -
        id: 'audit.x86_64'
        title: '*0:2.8.1-3.el7'
    -
        id: 'dhclient.x86_64'
        title: '*.el7_5.1'
    -
        id: 'dhcp*.x86_64'
        title: '*12:5.2.5-68.el7'
```

- SUSE Linux Enterprise Server (SLES)

```
patches:
    -
        id: 'amazon-ssm-agent.x86_64'
    -
        id: 'binutils'
        title: '*0:2.26.1-9.12.1'
    -
        id: 'glibc*.x86_64'
        title: '*2.19*'
    -
        id: 'dhcp*'
        title: '0:4.3.3-9.1'
    -
        id: 'lib*'
```

- Ubuntu Server

```
patches:
    -
        id: 'apparmor.amd64'
        title: '2.10.95-0ubuntu2.9'
    -
        id: 'cryptsetup.amd64'
        title: '*2:1.6.6-5ubuntu2.1'
    -
        id: 'cryptsetup-bin.*'
        title: '*2:1.6.6-5ubuntu2.1'
    -
        id: 'apt.amd64'
        title: '*1.2.27'
    -
        id: 'apt-utils.amd64'
        title: '*1.2.25'
```

- Windows

```
patches:
    -
        id: 'KB4284819'
```

```
        title: '2018-06 Cumulative Update for Windows Server 2016 (1709) for x64-
based Systems (KB4284819)'
    -
        id: 'KB4284833'
    -
        id: 'KB4284835'
        title: '2018-06 Cumulative Update for Windows Server 2016 (1803) for x64-
based Systems (KB4284835)'
    -
        id: 'KB4284880'
    -
        id: 'KB4338814'
```

Parametername: **`RebootOption`**

Nutzung: optional.

Optionen: `RebootIfNeeded` | `NoReboot`

Standardwert: `RebootIfNeeded`

> **Warning**
>
> Die Standardoption ist `RebootIfNeeded`. Stellen Sie sicher, dass Sie die richtige Option für Ihren Anwendungsfall auswählen. Wenn Ihre verwalteten Knoten beispielsweise sofort neu gestartet werden müssen, um einen Konfigurationsprozess abzuschließen, wählen Sie `RebootIfNeeded` aus. Oder wenn Sie die Verfügbarkeit von verwalteten Knoten bis zu einer geplanten Neustartzeit beibehalten müssen, wählen Sie `NoReboot` aus.

> **Important**
>
> Wir empfehlen nicht, Cluster-Instances in Amazon EMR (früher Amazon Elastic MapReduce genannt) zum Patchen zu verwendenPatch Manager. Wählen Sie insbesondere nicht die Option `RebootIfNeeded` für den Parameter `RebootOption` aus. (Diese Option ist in den SSM-Befehlsdokumenten für das Patchen von `AWS-RunPatchBaseline`, `AWS-RunPatchBaselineAssociation` und `AWS-RunPatchBaselineWithHooks` verfügbar.) Die zugrunde liegenden Befehle für das Patchen mithilfe von Patch Manager verwenden `yum`- und `dnf`-Befehle. Daher führen die Operationen aufgrund der Art und Weise, wie Pakete

installiert werden, zu Inkompatibilitäten. Informationen zu den bevorzugten Methoden für die Aktualisierung von Software auf Amazon-EMR-Clustern finden Sie unter Verwenden des Standard-AMI für Amazon EMR im Amazon EMR Management Guide.

RebootIfNeeded

Wenn Sie die Option `RebootIfNeeded` auswählen, wird der verwaltete Knoten in einem der folgenden Fälle neu gestartet:

- Patch Manager ist auf einem oder mehreren Patches installiert.

  Patch Manager wertet nicht aus, ob ein Neustart vom Patch erfordert wird. Das System wird neu gestartet, auch wenn der Patch keinen Neustart erfordert.
- Patch Manager erkennt ein oder mehrere Patches mit dem Status `INSTALLED_PENDING_REBOOT` während der `Install`-Operation.

  Der `INSTALLED_PENDING_REBOOT`-Status kann bedeuten, dass die Option `NoReboot` ausgewählt wurde, als der `Install`-Vorgang das letzte Mal ausgeführt wurde, oder dass ein Patch außerhalb von Patch Manager seit dem letzten Neustart des verwalteten Knotens installiert wurde.

Durch den Neustart von verwalteten Knoten wird in diesen beiden Fällen sichergestellt, dass aktualisierte Pakete aus dem Speicher gelöscht werden und das Patch- und Neustartverhalten über alle Betriebssysteme hinweg konsistent bleibt.

NoReboot

Wenn Sie die Option `NoReboot` auswählen, startet Patch Manager einen verwalteten Knoten nicht neu, selbst wenn über ihn während der `Install`-Operation Patches installiert wurden. Diese Option ist nützlich, wenn Sie wissen, dass für Ihre verwalteten Knoten nach dem Anwenden von Patches kein Neustart erforderlich ist oder Anwendungen bzw. Prozesse auf einem Knoten ausgeführt werden, die nicht durch einen Neustart beim Patchen unterbrochen werden sollten. Sie ist auch nützlich, wenn Sie mehr Kontrolle über das Timing von Neustarts von verwalteten Knoten wünschen, z. B. durch die Verwendung eines Wartungsfensters.

Wenn Sie die Option `NoReboot` auswählen und ein Patch installiert ist, wird dem Patch der Status `InstalledPendingReboot` zugewiesen. Der verwaltete Knoten selbst wird jedoch als `Non-Compliant` gekennzeichnet. Nach einem Neustart und Ausführung einer `Scan`-Operation wird der Status des verwalteten Knoten auf `Compliant` aktualisiert.

> **Important**
>
> Die Option `NoReboot` verhindert nur Neustarts auf Betriebssystemebene. Im Rahmen des Patch-Vorgangs können weiterhin Neustarts auf Service-Ebene erfolgen. Wenn Docker beispielsweise aktualisiert wird, können abhängige Services wie Amazon Elastic Container Service automatisch neu gestartet werden, auch wenn `NoReboot` aktiviert ist. Wenn Sie wichtige Services haben, die nicht unterbrochen werden dürfen, sollten Sie zusätzliche Maßnahmen in Betracht ziehen. Sie könnten z. B. Instances vorübergehend außer Betrieb nehmen oder Patches während Wartungsfenstern planen.

Datei zum Nachverfolgen der Patch-Installation: Um die Patch-Installation nachzuverfolgen, insbesondere von Patches, die seit dem letzten Neustart des Systems installiert wurden, erstellt Systems Manager eine Datei auf dem verwalteten Knoten.

> **Important**
>
> Löschen oder ändern Sie die Tracking-Datei nicht. Wenn diese Datei gelöscht oder beschädigt wird, ist der Patch-Compliance-Bericht für den verwalteten Knoten ungenau. Starten Sie in diesem Fall den Knoten neu und führen Sie eine Patch-Scan-Operation aus, um die Datei wiederherzustellen.

Diese Tracking-Datei wird an den folgenden Speicherorten auf Ihren verwalteten Knoten gespeichert:

- Linux-Betriebssysteme:
  - `/var/log/amazon/ssm/patch-configuration/patch-states-configuration.json`
  - `/var/log/amazon/ssm/patch-configuration/patch-inventory-from-last-operation.json`
- Windows Server-Betriebssystem:
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchStatesConfiguration.json`
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchInventoryFromLastOperation.json`

Parametername: **`BaselineOverride`**

Nutzung: optional.

Sie können Patching-Voreinstellungen zur Laufzeit mit dem `BaselineOverride`-Parameter definieren. Diese Baseline-Überschreibung wird als JSON-Objekt in einem S3-Bucket beibehalten. Sie stellt sicher, dass Patchvorgänge die bereitgestellten Baselines verwenden, die dem Host-Betriebssystem entsprechen, anstatt die Regeln aus der Standard-Patch-Baseline anzuwenden

> **Important**
>
> Der `BaselineOverride`-Dateiname darf die folgenden Zeichen nicht enthalten: Backtick (`), einfaches Anführungszeichen ('), doppeltes Anführungszeichen („) und Dollarzeichen ($).

Weitere Informationen zur Verwendung des Parameters `BaselineOverride` finden Sie unter Verwenden des BaselineOverride -Parameters.

Parametername: **`StepTimeoutSeconds`**

Nutzung: erforderlich.

Die Zeit in Sekunden – zwischen 1 und 36 000 Sekunden (10 Stunden) –, die ein Befehl in Anspruch nehmen darf, bevor er als fehlgeschlagen betrachtet wird.

SSM-Befehlsdokument zum Patchen: **`AWS-RunPatchBaselineAssociation`**

Wie das `AWS-RunPatchBaseline`-Dokument führt auch `AWS-RunPatchBaselineAssociation` Patching-Operationen auf Instances für sicherheitsrelevante und andere Arten von Updates aus. Sie können das Dokument `AWS-RunPatchBaselineAssociation` auch verwenden, um Patches sowohl für Betriebssysteme als auch für Anwendungen durchzuführen. (Unter Windows Server ist der Anwendungssupport auf Updates für Microsoft-Anwendungen beschränkt.)

Dieses Dokument unterstützt Amazon Elastic Compute Cloud (Amazon EC2) -Instances für LinuxmacOS, undWindows Server. Es unterstützt keine EC2 Nicht-Knoten in einer Hybrid- und Multi-Cloud-Umgebung. Das Dokument führt die entsprechenden Aktionen für jede Plattform durch und ruft ein Python-Modul auf Linux und macOS Instanzen und ein PowerShell Modul auf Windows-Instanzen auf.

`AWS-RunPatchBaselineAssociation` unterscheidet sich jedoch auf folgende Weise von `AWS-RunPatchBaseline`:

- `AWS-RunPatchBaselineAssociation` ist hauptsächlich für die Verwendung mit State Manager-Zuordnungen vorgesehen, die mithilfe von Quick Setup, einem Tool in AWS Systems Manager, erstellt wurden. Insbesondere, wenn Sie den Quick Setup-Host-Management-Konfigurationstyp verwenden und die Option Scan instances for missing patches daily (Instances täglich auf fehlende Patches scannen) auswählen, verwendet das System `AWS-RunPatchBaselineAssociation` für diese Operation.

  In den meisten Fällen sollten Sie jedoch beim Einrichten eigener Patching-Vorgänge `AWS-RunPatchBaseline` oder `AWS-RunPatchBaselineWithHooks` anstelle von `AWS-RunPatchBaselineAssociation` auswählen.

- Wenn Sie das `AWS-RunPatchBaselineAssociation`-Dokument verwenden, können Sie ein Tag-Schlüssel-Paar im `BaselineTags`-Parameterfeld des Dokuments angeben. Wenn eine benutzerdefinierte Patch-Baseline in Ihrem AWS-Konto System diese Tags verwendetPatch Manager, verwendet ein Tool in diese markierte Baseline AWS Systems Manager, wenn es auf den Ziel-Instances ausgeführt wird, und nicht die aktuell angegebene „Standard“ -Patch-Baseline für den Betriebssystemtyp.

  > **Note**
  >
  > Wenn Sie andere Patch-Operationen als die, mit denen Sie eingerichtet wurdenQuick Setup, verwenden möchten und Sie den zugehörigen optionalen `BaselineTags` Parameter verwenden möchten, müssen Sie einige zusätzliche Berechtigungen für das Instance-Profil für Amazon Elastic Compute Cloud (Amazon EC2) -Instances bereitstellen. `AWS-RunPatchBaselineAssociation` Weitere Informationen finden Sie unter Parametername: `BaselineTags`.

  Die beiden folgenden Formate sind gültig für Ihre `BaselineTags`-Parameter:

  `Key=tag-key,Values=tag-value`

  `Key=tag-key,Values=tag-value1,tag-value2,tag-value3`

  > **Important**
  >
  > Tag-Schlüssel und -Werte dürfen die folgenden Zeichen nicht enthalten: Backtick (`), einfaches Anführungszeichen ('), doppeltes Anführungszeichen („) und Dollarzeichen ($).

- Wenn `AWS-RunPatchBaselineAssociation` ausgeführt wird, werden die Patch-Compliance-Daten, die es sammelt, mit dem API-Befehl `PutComplianceItems` anstelle des Befehls `PutInventory`, der von `AWS-RunPatchBaseline` verwendet wird, aufgezeichnet. Dieser Unterschied bedeutet, dass die Patch-Compliance-Informationen gemäß einer bestimmten Zuordnung gespeichert und gemeldet werden. Patch-Compliance-Daten, die außerhalb dieser Zuordnung generiert wurden, werden nicht überschrieben.
- Die Patch-Compliance-Informationen, die nach der Ausführung von `AWS-RunPatchBaselineAssociation` gemeldet werden, geben an, ob eine Instance konform ist oder nicht. Es enthält keine Details auf Patch-Ebene, wie die Ausgabe des folgenden AWS Command Line Interface ()AWS CLI-Befehls zeigt. Der Befehl filtert auf `Association` als Compliance-Typ:

```
aws ssm list-compliance-items \
    --resource-ids "i-02573cafcfEXAMPLE" \
    --resource-types "ManagedInstance" \
    --filters "Key=ComplianceType,Values=Association,Type=EQUAL" \
    --region us-east-2
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "ComplianceItems": [
        {
            "Status": "NON_COMPLIANT",
            "Severity": "UNSPECIFIED",
            "Title": "MyPatchAssociation",
            "ResourceType": "ManagedInstance",
            "ResourceId": "i-02573cafcfEXAMPLE",
            "ComplianceType": "Association",
            "Details": {
                "DocumentName": "AWS-RunPatchBaselineAssociation",
                "PatchBaselineId": "pb-0c10e65780EXAMPLE",
                "DocumentVersion": "1"
            },
            "ExecutionSummary": {
                "ExecutionTime": 1590698771.0
            },
            "Id": "3e5d5694-cd07-40f0-bbea-040e6EXAMPLE"
        }
    ]
```

```
}
```

Wenn ein Tag-Schlüssel-Paar-Wert als Parameter für das `AWS-RunPatchBaselineAssociation`-Dokument angegeben wurde, sucht Patch Manager nach einer benutzerdefinierten Patch-Baseline, die mit dem Betriebssystemtyp übereinstimmt und mit demselben Tag-Schlüssel-Paar gekennzeichnet wurde. Diese Suche ist nicht auf die aktuell angegebene Standard-Patch-Baseline oder die Baseline beschränkt, die einer Patch-Gruppe zugewiesen ist. Wenn keine Baseline mit den angegebenen Tags gefunden wird, sucht Patch Manager als Nächstes nach einer Patch-Gruppe, wenn eine in dem Befehl angegeben wurde, der `AWS-RunPatchBaselineAssociation` ausführt. Wenn keine Patch-Gruppe übereinstimmt, wird Patch Manager auf die aktuelle Standard-Patch-Baselines für das Betriebssystemkonto zurückgesetzt.

Wenn mehr als eine Patch-Baseline mit den Tags gefunden wird, die im `AWS-RunPatchBaselineAssociation`-Dokument angegeben sind, gibt Patch Manager eine Fehlermeldung zurück, die angibt, dass nur eine Patch-Baseline mit diesem Schlüssel-Wert-Paar gekennzeichnet werden kann, damit der Vorgang fortgesetzt wird.

> **Note**
>
> Auf Linux-Knoten wird der entsprechende Paketmanager für jeden Knotentyp verwendet, um Pakete zu installieren:
>
> - Amazon-Linux-2-, Oracle Linux- und RHEL-Instances verwenden YUM. Für YUM-Operationen ist eine `Python 2.6` oder eine Patch Manager neuere unterstützte Version (2.6 - 3.12) erforderlich. Amazon Linux 2023 verwendet DNF. Für DNF-Vorgänge erfordert Patch Manager eine unterstützte Version von `Python 2` oder `Python 3` (2.6–3.12).
> - Debian Serverund Ubuntu Server Instanzen verwenden APT. Für APT-Operationen Patch Manager ist eine unterstützte Version von `Python 3` (3.0 - 3.12) erforderlich.

Nachdem der Scan abgeschlossen wurde oder alle genehmigten und zutreffenden Updates installiert und je nach Bedarf Neustarts durchgeführt wurden, werden Patch-Compliance-Informationen auf einer Instance generiert und wieder an den Patchmanager-Service gemeldet.

> **Note**
>
> Wenn der Parameter `RebootOption` im Dokument `AWS-RunPatchBaselineAssociation` auf `NoReboot` gesetzt ist, wird die Instance nach dem Ausführen von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.

Weitere Informationen zum Anzeigen von Patch-Compliance-Daten finden Sie unter Info zu Patch Compliance.

`AWS-RunPatchBaselineAssociation`-Parameter

`AWS-RunPatchBaselineAssociation` unterstützt fünf Parameter. Die Parameter `Operation` und `AssociationId` müssen angegeben werden. Die Parameter `InstallOverrideList`, `RebootOption` und `BaselineTags` sind optional.

Parameter

- Parametername: Operation
- Parametername: BaselineTags
- Parametername: AssociationId
- Parametername: InstallOverrideList
- Parametername: RebootOption

Parametername: **`Operation`**

Nutzung: erforderlich.

Optionen: `Scan` | `Install`.

Scan

Wenn Sie die Option `Scan` wählen, bestimmt `AWS-RunPatchBaselineAssociation` den Patch-Compliance-Status der Instance und meldet diese Informationen an Patch Manager. `Scan` fordert nicht zum Installieren von Updates oder zum Neustarten von Instances auf. Stattdessen erkennt der Vorgang, wo für die Instance genehmigte und geeignete Updates fehlen.

Installieren

Bei Auswahl der Option `Install` versucht `AWS-RunPatchBaselineAssociation`, die genehmigten und geeigneten Updates zu installieren, die auf der Instance fehlen. Patch-Compliance-Informationen, die als Teil eines `Install`-Vorgangs generiert wurden, enthalten keine fehlenden Updates, melden allerdings möglicherweise Updates im Fehlerzustand, wenn die Installation des Updates aus einem beliebigen Grund nicht erfolgreich war. Immer wenn ein Update auf einer Instance installiert wird, wird die Instance neu gestartet, um sicherzustellen, dass es installiert und aktiviert ist. (Ausnahme: Wenn der `RebootOption`-Parameter im `AWS-RunPatchBaselineAssociation`-Dokument auf `NoReboot` gesetzt ist, wird die Instance nach der Ausführung von Patch Manager nicht neugestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

> **Note**
>
> Wenn ein von den Basisregeln festgelegter Patch installiert wird, bevor der Patch Manager die Instance aktualisiert, wird das System möglicherweise nicht wie erwartet neu gestartet. Dies kann passieren, wenn ein Patch manuell von einem Benutzer oder automatisch von einem anderen Programm, z. B. dem `unattended-upgrades`-Paket auf Ubuntu Server, installiert wird.

Parametername: **`BaselineTags`**

Nutzung: optional.

`BaselineTags` ist ein eindeutiges Tag-Schlüssel-Wert-Paar, das Sie auswählen und einer individuellen benutzerdefinierten Patch-Baseline zuweisen. Sie können einen oder mehrere Werte für diesen Parameter angeben. Beider der folgenden Formate sind gültig:

`Key=`*`tag-key`*`,Values=`*`tag-value`*

`Key=`*`tag-key`*`,Values=`*`tag-value1`*`,`*`tag-value2`*`,`*`tag-value3`*

> **Important**
>
> Tag-Schlüssel und -Werte dürfen die folgenden Zeichen nicht enthalten: Backtick (`), einfaches Anführungszeichen ('), doppeltes Anführungszeichen („) und Dollarzeichen ($).

Der `BaselineTags`-Wert wird von Patch Manager verwendet, um sicherzustellen, dass eine Gruppe von Instances, für die in einem Vorgang Patches durchgeführt werden, den genau gleichen Satz genehmigter Patches aufweist. Wenn der Patching-Vorgang ausgeführt wird, überprüft Patch Manager, ob eine Patch-Baseline für den Betriebssystemtyp mit demselben Schlüssel-Wert-Paar versehen ist, das Sie für `BaselineTags` angeben. Wenn eine Übereinstimmung vorliegt, wird diese benutzerdefinierte Patch-Baseline verwendet. Wenn keine Übereinstimmung vorliegt, wird eine Patch-Baseline anhand einer beliebigen Patchgruppe identifiziert, die für die Patching-Operation angegeben wurde. Wenn keine vorhanden ist, wird die AWS verwaltete vordefinierte Patch-Baseline für dieses Betriebssystem verwendet.

Zusätzliche Berechtigungsanforderungen

Wenn Sie andere Patch-Operationen als die, mit denen Sie eingerichtet wurdenQuick Setup, verwenden und den optionalen `BaselineTags` Parameter verwenden möchten, müssen Sie dem Instance-Profil für Amazon Elastic Compute Cloud (Amazon EC2) -Instances die folgenden Berechtigungen hinzufügen. `AWS-RunPatchBaselineAssociation`

> **Note**
>
> Quick Setupund unterstützen `AWS-RunPatchBaselineAssociation` keine lokalen Server und virtuellen Maschinen ()VMs.

```
{
    "Effect": "Allow",
    "Action": [
        "ssm:DescribePatchBaselines",
        "tag:GetResources"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ssm:GetPatchBaseline",
        "ssm:DescribeEffectivePatchesForPatchBaseline"
    ],
    "Resource": "patch-baseline-arn"
}
```

*patch-baseline-arn*Ersetzen Sie es durch den Amazon-Ressourcennamen (ARN) der Patch-Baseline, auf die Sie Zugriff gewähren möchten, im folgenden Format`arn:aws:ssm:us-east-2:123456789012:patchbaseline/pb-0c10e65780EXAMPLE`.

Parametername: **`AssociationId`**

Nutzung: erforderlich.

`AssociationId` ist die ID einer Zuordnung in State Manager, einem Tool in AWS Systems Manager. Es wird von Patch Manager verwendet, um einer angegebenen Zuordnung Compliance-Daten hinzuzufügen. Diese Zuordnung bezieht sich auf einen `Scan`-Patching-Vorgang, der in einer in Quick Setup erstellten Host-Management-Konfiguration aktiviert ist. Indem Sie die Patching-Ergebnisse als Zuordnungs-Compliance-Daten statt als Inventar-Compliance-Daten senden, werden die vorhandenen Inventar-Compliance-Informationen für Ihre Instances nach einem Patchvorgang oder bei anderen Zuordnungen nicht überschrieben. IDs Wenn Sie noch keine Zuordnung erstellt haben, die Sie verwenden möchten, können Sie eine erstellen, indem Sie den Befehl create-association ausführen. Zum Beispiel:

Linux & macOS

```
aws ssm create-association \
    --name "AWS-RunPatchBaselineAssociation" \
    --association-name "MyPatchHostConfigAssociation" \
    --targets
 "Key=instanceids,Values=[i-02573cafcfEXAMPLE,i-07782c72faEXAMPLE,i-07782c72faEXAMPLE]"
 \
    --parameters "Operation=Scan" \
    --schedule-expression "cron(0 */30 * * * ? *)" \
    --sync-compliance "MANUAL" \
    --region us-east-2
```

Windows Server

```
aws ssm create-association ^
    --name "AWS-RunPatchBaselineAssociation" ^
    --association-name "MyPatchHostConfigAssociation" ^
    --targets
 "Key=instanceids,Values=[i-02573cafcfEXAMPLE,i-07782c72faEXAMPLE,i-07782c72faEXAMPLE]"
 ^
    --parameters "Operation=Scan" ^
    --schedule-expression "cron(0 */30 * * * ? *)" ^
```

```
    --sync-compliance "MANUAL" ^
    --region us-east-2
```

Parametername: **`InstallOverrideList`**

Nutzung: optional.

Mit `InstallOverrideList` können Sie eine https-URL oder eine Amazon Simple Storage Service (Amazon S3)-URL im Pfadstil zu einer Liste mit zu installierenden Patches angeben. Diese im YAML-Format geführte Patch-Installationsliste überschreibt die von der Standard-Patch-Baseline angegebenen Patches. Dies bietet Ihnen eine differenziertere Kontrolle darüber, welche Patches auf Ihren Instances installiert sind.

> **Important**
>
> Der `InstallOverrideList`-Dateiname darf die folgenden Zeichen nicht enthalten: Backtick (`), einfaches Anführungszeichen ('), doppeltes Anführungszeichen („) und Dollarzeichen ($).

Das Verhalten des Patch-Vorgangs bei Verwendung des `InstallOverrideList`-Parameters unterscheidet sich zwischen Linux- und macOS-verwalteten Knoten und Windows Server-verwalteten Knoten. Unter Linux und macOS versucht Patch Manager, Patches aus der Patchliste `InstallOverrideList` anzuwenden, die in einem auf dem Knoten aktivierten Repository vorhanden sind, unabhängig davon, ob die Patches den Patch-Baseline-Regeln entsprechen oder nicht. Auf Windows Server-Knoten werden Patches in der `InstallOverrideList`-Patch-Liste jedoch nur angewendet, wenn sie auch den Patch-Baseline-Regeln entsprechen.

Beachten Sie, dass Compliance-Berichte Patch-Status entsprechend den Angaben in der Patch-Baseline wiedergeben, nicht entsprechend Ihren Angaben in einer `InstallOverrideList`-Liste von Patches. Mit anderen Worten: Scan-Operationen ignorieren den Parameter `InstallOverrideList`. Auf diese Weise wird sichergestellt, dass Compliance-Berichte den Patch-Status konsistent entsprechend der Richtlinie wiedergeben und nicht danach, was für eine bestimmte Patching-Operation genehmigt wurde.

Gültige URL-Formate

> **Note**
>
> Wenn Ihre Datei in einem öffentlich zugänglichen Bucket gespeichert ist, können Sie entweder ein HTTPS-URL-Format oder eine URL im Amazon S3-Pfadstil angeben. Wenn Ihre Datei in einem privaten Bucket gespeichert ist, müssen Sie eine URL im Amazon S3-Pfadstil angeben.

- Beispiel des HTTPS-URL-Formats:

```
https://s3.amazonaws.com/amzn-s3-demo-bucket/my-windows-override-list.yaml
```

- Beispiel-URL im Amazon-S3-Pfadstil:

```
s3://amzn-s3-demo-bucket/my-windows-override-list.yaml
```

Gültige YAML-Inhaltsformate

Die Formate, die Sie verwenden, um Patches in Ihrer Liste anzugeben, hängen von dem Betriebssystem Ihrer Instance ab. Das allgemeine Format lautet jedoch folgendermaßen:

```
patches:
    -
        id: '{patch-d}'
        title: '{patch-title}'
        {additional-fields}:{values}
```

Sie können zwar zusätzliche Felder in der YAML-Datei bereitstellen, diese werden jedoch während der Patch-Operationen ignoriert.

Darüber hinaus empfehlen wir zu überprüfen, ob das Format Ihrer YAML-Datei gültig ist, bevor Sie die Liste in Ihrem S3-Bucket hinzufügen oder aktualisieren. Weitere Informationen zum YAML-Format finden Sie unter yaml.org. Für Validierungstool-Optionen suchen Sie im Internet nach „yaml format validators“ durch.

- Microsoft Windows

  id

Das Feld id ist ein Pflichtfeld. Verwenden Sie es, um Patches mithilfe von Microsoft Knowledge Base IDs (z. B. KB2736693) und Microsoft Security Bulletin IDs (z. B. MS17 -023) anzugeben.

Alle anderen Felder, die Sie in einer Patch-Liste für Windows bereitstellen möchten, sind optional und dienen nur zu Ihrer eigenen Information. Sie können zusätzlichen Felder verwenden, wie z. B. Titel, Klassifizierung, Schweregrad oder andere Angaben für detailliertere Informationen über die angegebenen Patches.

- Linux

  id

  Das Feld id ist ein Pflichtfeld. Verwenden Sie es, um Patches mit Paketnamen und Architektur anzugeben. Beispiel: `'dhclient.x86_64'`. Sie können Platzhalter in der ID verwenden, um mehrere Pakete anzugeben. Zum Beispiel `'dhcp*'` und `'dhcp*1.*'`.

  Titel

  Das Feld Titel ist optional, es bietet jedoch auf Linux-Systemen zusätzliche Filterfunktionen. Wenn Sie Titel verwenden, sollte er die Versionsinformationen des Pakets in einem der folgenden Formate enthalten:

  YUM/Red Hat Enterprise Linux (RHEL):

  ```
  {name}.{architecture}:{epoch}:{version}-{release}
  ```

  APT

  ```
  {name}.{architecture}:{version}
  ```

  Für Linux-Patch-Titel können Sie einen oder mehrere Platzhalter in beliebigen Positionen verwenden, um die Anzahl der Paketzuordnungen zu erhöhen. Beispiel: `'*32:9.8.2-0.*.rc1.57.amzn1'`.

  Zum Beispiel:

  - apt-Paketversion 1.2.25 ist derzeit auf Ihrer Instance installiert, aber Version 1.2.27 ist jetzt verfügbar.
  - Fügen Sie die apt.amd64-Version 1.2.27 der Liste hinzu. Sie ist abhängig von apt utils.amd64 Version 1.2.27, aber apt-utils.amd64 Version 1.2.25 ist in der Liste angegeben.

In diesem Fall wird die Installation der APT-Version 1.2.27 blockiert und als „Fehlgeschlagen-“ gemeldet. NonCompliant

Andere Felder

Alle anderen Felder, die Sie in einer Patch-Liste für Linux bereitstellen möchten, sind optional und dienen nur zu Ihrer eigenen Information. Sie können zusätzlichen Felder verwenden, wie z. B. Klassifizierung, Schweregrad oder andere Angaben für detailliertere Informationen über die angegebenen Patches.

Patch-Beispiellisten

- Windows

```
patches:
    -
        id: 'KB4284819'
        title: '2018-06 Cumulative Update for Windows Server 2016 (1709) for x64-
based Systems (KB4284819)'
    -
        id: 'KB4284833'
    -
        id: 'KB4284835'
        title: '2018-06 Cumulative Update for Windows Server 2016 (1803) for x64-
based Systems (KB4284835)'
    -
        id: 'KB4284880'
    -
        id: 'KB4338814'
```

- APT

```
patches:
    -
        id: 'apparmor.amd64'
        title: '2.10.95-0ubuntu2.9'
    -
        id: 'cryptsetup.amd64'
        title: '*2:1.6.6-5ubuntu2.1'
    -
        id: 'cryptsetup-bin.*'
```

```
        title: '*2:1.6.6-5ubuntu2.1'
    -
        id: 'apt.amd64'
        title: '*1.2.27'
    -
        id: 'apt-utils.amd64'
        title: '*1.2.25'
```

- Amazon Linux 2

```
patches:
    -
        id: 'kernel.x86_64'
    -
        id: 'bind*.x86_64'
        title: '39.11.4-26.P2.amzn2.5.2'

        id: 'glibc*'
    -
        id: 'dhclient*'
        title: '*4.2.5-58.amzn2'
    -
        id: 'dhcp*'
        title: '*4.2.5-77.amzn2'
```

- Red Hat Enterprise Linux (RHEL)

```
patches:
    -
        id: 'NetworkManager.x86_64'
        title: '*1:1.10.2-14.el7_5'
    -
        id: 'NetworkManager-*.x86_64'
        title: '*1:1.10.2-14.el7_5'
    -
        id: 'audit.x86_64'
        title: '*0:2.8.1-3.el7'
    -
        id: 'dhclient.x86_64'
        title: '*.el7_5.1'
    -
        id: 'dhcp*.x86_64'
        title: '*12:5.2.5-68.el7'
```

- SUSE Linux Enterprise Server (SLES)

```
patches:
    -
        id: 'amazon-ssm-agent.x86_64'
    -
        id: 'binutils'
        title: '*0:2.26.1-9.12.1'
    -
        id: 'glibc*.x86_64'
        title: '*2.19*'
    -
        id: 'dhcp*'
        title: '0:4.3.3-9.1'
    -
        id: 'lib*'
```

- Ubuntu Server

```
patches:
    -
        id: 'apparmor.amd64'
        title: '2.10.95-0ubuntu2.9'
    -
        id: 'cryptsetup.amd64'
        title: '*2:1.6.6-5ubuntu2.1'
    -
        id: 'cryptsetup-bin.*'
        title: '*2:1.6.6-5ubuntu2.1'
    -
        id: 'apt.amd64'
        title: '*1.2.27'
    -
        id: 'apt-utils.amd64'
        title: '*1.2.25'
```

- Windows

```
patches:
    -
        id: 'KB4284819'
```

```
        title: '2018-06 Cumulative Update for Windows Server 2016 (1709) for x64-
based Systems (KB4284819)'
    -
        id: 'KB4284833'
    -
        id: 'KB4284835'
        title: '2018-06 Cumulative Update for Windows Server 2016 (1803) for x64-
based Systems (KB4284835)'
    -
        id: 'KB4284880'
    -
        id: 'KB4338814'
```

Parametername: **`RebootOption`**

Nutzung: optional.

Optionen: `RebootIfNeeded` | `NoReboot`

Standardwert: `RebootIfNeeded`

> **Warning**
>
> Die Standardoption ist `RebootIfNeeded`. Stellen Sie sicher, dass Sie die richtige Option für Ihren Anwendungsfall auswählen. Wenn Ihre Instances beispielsweise sofort neu starten müssen, um einen Konfigurationsprozess abzuschließen, wählen Sie `RebootIfNeeded` aus. Oder wenn Sie die Verfügbarkeit von Instances bis zu einer geplanten Neustartzeit beibehalten müssen, wählen Sie `NoReboot` aus.

> **Important**
>
> Die Option `NoReboot` verhindert nur Neustarts auf Betriebssystemebene. Im Rahmen des Patch-Vorgangs können weiterhin Neustarts auf Service-Ebene erfolgen. Wenn Docker beispielsweise aktualisiert wird, können abhängige Services wie Amazon Elastic Container Service automatisch neu gestartet werden, auch wenn `NoReboot` aktiviert ist. Wenn Sie wichtige Services haben, die nicht unterbrochen werden dürfen, sollten Sie zusätzliche Maßnahmen in Betracht ziehen. Sie könnten z. B. Instances vorübergehend außer Betrieb nehmen oder Patches während Wartungsfenstern planen.

> **Important**
>
> Wir empfehlen nicht, Cluster-Instances in Amazon EMR (früher Amazon Elastic MapReduce genannt) zum Patchen zu verwendenPatch Manager. Wählen Sie insbesondere nicht die Option `RebootIfNeeded` für den Parameter `RebootOption` aus. (Diese Option ist in den SSM-Befehlsdokumenten für das Patchen von `AWS-RunPatchBaseline`, `AWS-RunPatchBaselineAssociation` und `AWS-RunPatchBaselineWithHooks` verfügbar.) Die zugrunde liegenden Befehle für das Patchen mithilfe von Patch Manager verwenden `yum`- und `dnf`-Befehle. Daher führen die Operationen aufgrund der Art und Weise, wie Pakete installiert werden, zu Inkompatibilitäten. Informationen zu den bevorzugten Methoden für die Aktualisierung von Software auf Amazon-EMR-Clustern finden Sie unter Verwenden des Standard-AMI für Amazon EMR im Amazon EMR Management Guide.

RebootIfNeeded

Wenn Sie die Option `RebootIfNeeded` auswählen, wird die Instance in einem der folgenden Fälle neu gestartet:

- Patch Manager ist auf einem oder mehreren Patches installiert.

  Patch Manager wertet nicht aus, ob ein Neustart vom Patch erfordert wird. Das System wird neu gestartet, auch wenn der Patch keinen Neustart erfordert.
- Patch Manager erkennt ein oder mehrere Patches mit dem Status `INSTALLED_PENDING_REBOOT` während der `Install`-Operation.

  Der `INSTALLED_PENDING_REBOOT`-Status kann bedeuten, dass die Option `NoReboot` ausgewählt wurde, als der `Install`-Vorgang das letzte Mal ausgeführt wurde, oder dass ein Patch außerhalb von Patch Manager seit dem letzten Neustart des verwalteten Knotens installiert wurde.

Durch den Neustart von Instances wird in diesen beiden Fällen sichergestellt, dass aktualisierte Pakete aus dem Speicher gelöscht werden und das Patch- und Neustartverhalten über alle Betriebssysteme hinweg konsistent bleibt.

NoReboot

Wenn Sie die Option `NoReboot` auswählen, startet Patch Manager eine Instance nicht neu, selbst wenn über ihn während der `Install`-Operation Patches installiert wurden. Diese Option ist nützlich, wenn Sie wissen, dass für Ihre Instances nach dem Anwenden von Patches kein

Neustart erforderlich ist oder Anwendungen bzw. Prozesse auf einer Instance ausgeführt werden, die nicht durch einen Neustart des Patches unterbrochen werden sollten. Sie ist auch nützlich, wenn Sie mehr Kontrolle über das Timing von Instance-Neustarts wünschen, z. B. durch die Verwendung eines Wartungsfensters.

Datei zum Nachverfolgen der Patch-Installation (Tracking-Datei): Um die Patch-Installation nachzuverfolgen, insbesondere von Patches, die seit dem letzten Neustart des Systems installiert wurden, erstellt Systems Manager eine Datei auf der verwalteten Instance.

> **Important**
>
> Löschen oder ändern Sie die Tracking-Datei nicht. Wenn diese Datei gelöscht oder beschädigt wird, ist der Patch-Compliance-Bericht für die Instance ungenau. Starten Sie in diesem Fall die Instance neu und führen Sie einen Patch-Scan-Vorgang aus, um die Datei wiederherzustellen.

Diese Tracking-Datei wird an den folgenden Speicherorten auf Ihren verwalteten Instances gespeichert:

- Linux-Betriebssysteme:
  - `/var/log/amazon/ssm/patch-configuration/patch-states-configuration.json`
  - `/var/log/amazon/ssm/patch-configuration/patch-inventory-from-last-operation.json`
- Windows Server-Betriebssystem:
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchStatesConfiguration.json`
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchInventoryFromLastOperation.json`

SSM-Befehlsdokument zum Patchen: **`AWS-RunPatchBaselineWithHooks`**

AWS Systems Manager unterstütztAWS-RunPatchBaselineWithHooks, ein Systems Manager Manager-Dokument (SSM-Dokument) fürPatch Manager, ein Tool in AWS Systems Manager. Dieses SSM-Dokument führt Patch-Operationen auf verwaltete Knoten sowohl für sicherheitsrelevante als auch für andere Arten von Updates durch.

`AWS-RunPatchBaselineWithHooks` unterscheidet sich auf folgende Weise von `AWS-RunPatchBaseline`:

- Ein Wrapper-Dokument – `AWS-RunPatchBaselineWithHooks` ist ein Wrapper für `AWS-RunPatchBaseline` und setzt für einige seiner Operationen auf`AWS-RunPatchBaseline`.
- Die **`Install`**-Operation – `AWS-RunPatchBaselineWithHooks` unterstützt Lebenszyklus-Hooks, die während dem Patchen von verwalteten Knoten an festgelegten Punkten ausgeführt werden. Da Patch-Installationen manchmal den Neustart von verwalteten Knoten erfordern, ist die Patch-Operation in zwei Ereignisse unterteilt, wobei insgesamt drei Hooks enthalten sind, die benutzerdefinierte Funktion unterstützen. Der erste Hook ist vor der `Install with NoReboot`-Operation. Der zweite Hook ist nach der `Install with NoReboot`-Operation. Der dritte Hook ist nach dem Neustart des verwalteten Knoten verfügbar.
- Keine Unterstützung für benutzerdefinierte Patchlisten – `AWS-RunPatchBaselineWithHooks` unterstützt den `InstallOverrideList`-Parameter nicht.
- SSM Agent-Support – `AWS-RunPatchBaselineWithHooks` erfordert die Installation von SSM Agent 3.0.502 oder höher auf dem zu patchenden verwalteten Knoten.

Wenn das Dokument ausgeführt wird, verwendet es die Patch-Baseline, die aktuell der „Standard“ für einen Betriebssystemtyp ist, wenn keine Patch-Gruppe angegeben ist. Andernfalls werden die Patch-Baselines verwendet, die der Patch-Gruppe zugeordnet sind. Informationen zu Patch-Gruppen finden Sie unter Patch-Gruppen.

Sie können das Dokument `AWS-RunPatchBaselineWithHooks` verwenden, um Patches sowohl für Betriebssysteme als auch für Anwendungen durchzuführen. (Unter Windows Server ist der Anwendungssupport auf Updates für Microsoft-Anwendungen beschränkt.)

Dieses Dokument unterstützt von Linux und von Windows Server verwaltete Knoten. Das Dokument führt die entsprechenden Aktionen für jede Plattform durch.

> **Note**
>
> `AWS-RunPatchBaselineWithHooks` wird in macOS nicht unterstützt.

Linux

Auf Linux-verwalteten Knoten ruft das Dokument `AWS-RunPatchBaselineWithHooks` ein Python-Modul auf, das wiederum einen entsprechenden Snapshot der Patch-Baseline

für den verwalteten Knoten herunterlädt. Dieser Patch-Baseline-Snapshot verwendet die definierten Regeln und Listen der genehmigten und gesperrten Patches, um den entsprechenden Paketmanager für jeden Knoten-Typ anzutreiben:

- Von Amazon Linux 2, Oracle Linux und RHEL 7 verwaltete Knoten verwenden YUM. Für YUM-Operationen ist eine `Python 2.6` oder eine Patch Manager neuere unterstützte Version (2.6 - 3.12) erforderlich. Amazon Linux 2023 verwendet DNF. Für DNF-Operationen Patch Manager ist eine unterstützte Version von `Python 2` oder `Python 3` (2.6 - 3.12) erforderlich.
- Von RHEL 8 verwaltete Knoten verwenden DNF. Für DNF-Operationen Patch Manager ist eine unterstützte Version von `Python 2` or `Python 3` (2.6 - 3.12) erforderlich. (Keine der beiden Versionen ist standardmäßig auf RHEL 8 installiert. Sie müssen die eine oder andere Version manuell installieren.)
- Debian Serverund Ubuntu Server Instanzen verwenden APT. Für APT-Operationen Patch Manager ist eine unterstützte Version von `Python 3` (3.0 - 3.12) erforderlich.

Windows Server

Auf Windows Server verwalteten Knoten lädt das `AWS-RunPatchBaselineWithHooks` Dokument ein PowerShell Modul herunter und ruft es auf, das wiederum einen Snapshot der Patch-Baseline herunterlädt, die für den verwalteten Knoten gilt. Dieser Patch-Baseline-Snapshot enthält eine Liste genehmigter Patches, die kompiliert werden, indem die Patch-Baseline auf einem WSUS-Server (Windows Server Update Services) abgefragt wird. Diese Liste wird an die Windows Update-API weitergeleitet, die das Herunterladen und Installieren des genehmigten Patches entsprechend steuert.

Jeder Snapshot ist spezifisch für eine AWS-Konto Patch-Gruppe, ein Betriebssystem und eine Snapshot-ID. Der Snapshot wird über eine vorsignierte Amazon Simple Storage Service (Amazon S3)-URL bereitgestellt, die 24 Stunden nach Erstellung des Snapshots abläuft. Wenn Sie jedoch denselben Snapshot-Inhalt auf andere verwaltete Knoten anwenden möchten, können Sie nach Ablauf der URL bis zu drei Tage nach Erstellung des Snapshots eine neue vorsignierte Amazon-S3-URL generieren. Verwenden Sie dazu den Befehl get-deployable-patch-snapshot-for-instance.

Nachdem alle genehmigten und zutreffenden Updates installiert und je nach Bedarf Neustarts durchgeführt wurden, werden Patch-Compliance-Informationen auf einem verwalteten Knoten generiert und wieder an Patch Manager gemeldet.

Wenn der Parameter `RebootOption` im Dokument `AWS-RunPatchBaselineWithHooks` auf `NoReboot` gesetzt ist, wird der verwaltete Knoten nach dem Ausführen von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.

> **Important**
>
> Die Option `NoReboot` verhindert zwar Neustarts des Betriebssystems, aber keine Neustarts auf Service-Ebene, die beim Aktualisieren bestimmter Pakete erfolgen können. Beispielsweise kann die Aktualisierung von Paketen wie Docker automatische Neustarts abhängiger Services auslösen (etwa Container-Orchestrierungsservices), selbst wenn `NoReboot` angegeben ist.

Weitere Informationen zum Anzeigen von Patch-Compliance-Daten finden Sie unter Info zu Patch Compliance.

`AWS-RunPatchBaselineWithHooks`-Betriebsschritte

Wenn `AWS-RunPatchBaselineWithHooks` ausgeführt wird, werden die folgenden Schritte durchgeführt:

1. Scan – Eine `Scan`-Operation mit `AWS-RunPatchBaseline` wird auf dem verwalteten Knoten ausgeführt und ein Compliance-Bericht wird generiert und hochgeladen.
2. Überprüfen der lokalen Patch-Zustände – Ein Skript wird ausgeführt, um zu bestimmen, welche Schritte auf der Grundlage der ausgewählten Operation und dem `Scan`-Ergebnis aus Schritt 1 ausgeführt werden.
   a. Wenn die ausgewählte Operation `Scan` ist, wird die Operation als abgeschlossen markiert. Die Operation ist abgeschlossen.
   b. Wenn die ausgewählte Operation `Install` ist, werte Patch Manager das `Scan`-Ergebnis aus Schritt 1, um zu bestimmen, was als nächstes ausgeführt werden soll:
      i. Wenn keine fehlenden Patches erkannt werden und keine ausstehenden Neustarts erforderlich sind, fährt die Operation direkt mit dem letzten Schritt (Schritt 8) fort, der einen von Ihnen bereitgestellten Hook enthält. Alle Schritte dazwischen werden übersprungen.
      ii. Wenn keine fehlenden Patches erkannt werden, aber ausstehende Neustarts erforderlich sind und die Neustartoption `NoReboot` ist, fährt die Operation direkt mit dem letzten Schritt (Schritt 8) fort, der einen von Ihnen bereitgestellten Hook enthält. Alle Schritte dazwischen werden übersprungen.

iii. Andernfalls fährt die Operation mit dem nächsten Schritt fort.

3. Hook-Operation vor dem Patchen – Das SSM-Dokument, das Sie für den ersten Lebenszyklus-Hook bereitgestellt haben, `PreInstallHookDocName` wird auf dem verwalteten Knoten ausgeführt.
4. Installation mit NoReboot — Auf dem verwalteten Knoten `AWS-RunPatchBaseline` wird ein `Install` Vorgang `NoReboot` mit der Neustartoption ausgeführt, und es wird ein Konformitätsbericht generiert und hochgeladen.
5. Hook-Operation nach der Installation – Das SSM-Dokument, das Sie für den zweiten Lebenszyklus-Hook bereitgestellt haben, `PostInstallHookDocName` wird auf dem verwalteten Knoten ausgeführt.
6. Überprüfen des Neustarts – Ein Skript wird ausgeführt, um festzustellen, ob ein Neustart für den verwalteten Knoten erforderlich ist und welche Schritte ausgeführt werden sollen:
   a. Wenn die ausgewählte Neustartoption `NoReboot` ist, geht die Operation direkt zum letzten Schritt (Schritt 8) über, der einen von Ihnen bereitgestellten Hook enthält. Alle Schritte dazwischen werden übersprungen.
   b. Wenn die ausgewählte Neustartoption `RebootIfNeeded` ist, prüft Patch Manager, ob ausstehende Neustarts aus dem in Schritt 4 erfassten Bestand erforderlich sind. Dies bedeutet, dass der Vorgang mit Schritt 7 fortgesetzt wird und der verwaltete Knoten in einem der folgenden Fälle neu gestartet wird:
      i. Patch Manager hat einen oder mehrere Patches installiert. (Patch Manager wertet nicht aus, ob für den Patch ein Neustart erforderlich ist. Das System wird auch dann neu gestartet, wenn der Patch keinen Neustart erfordert.)
      ii. Patch Manager erkennt ein oder mehrere Patches mit dem Status `INSTALLED_PENDING_REBOOT` während des Installationsvorgangs. Der `INSTALLED_PENDING_REBOOT`-Status kann bedeuten, dass die Option `NoReboot` ausgewählt wurde, als der Installations-Vorgang das letzte Mal ausgeführt wurde, oder dass ein Patch außerhalb von Patch Manager seit dem letzten Neustart des verwalteten Knotens installiert wurde.

   Wenn keine Patches gefunden werden, die diese Kriterien erfüllen, ist der Patch-Vorgang für verwaltete Knoten abgeschlossen, und der Vorgang fährt direkt mit dem letzten Schritt (Schritt 8) fort, der einen von Ihnen bereitgestellten Hook enthält. Alle Schritte dazwischen werden übersprungen.

7. Neustart und Bericht – Eine Installations-Operation mit der Neustart-Option `RebootIfNeeded` wird auf dem verwalteten Knoten unter Verwendung von `AWS-RunPatchBaseline` ausgeführt und ein Compliance-Bericht wird generiert und hochgeladen.
8. Hook-Operation nach Neustart – Das SSM-Dokument, das Sie für den dritten Lebenszyklus-Hook bereitgestellt haben, `OnExitHookDocName` wird auf dem verwalteten Knoten ausgeführt.

Bei einer `Scan`-Operation wird der Prozess der Ausführung des Dokuments beendet, wenn Schritt 1 fehlschlägt, und der Schritt wird als fehlgeschlagen gemeldet, obwohl nachfolgende Schritte als erfolgreich gemeldet werden.

Wenn bei einem `Install`-Vorgang einer der `aws:runDocument`-Schritte während des Vorgangs fehlschlagen, werden diese Schritte als fehlgeschlagen gemeldet, und der Vorgang fährt direkt mit dem letzten Schritt (Schritt 8) fort, der einen von Ihnen bereitgestellten Hook enthält. Alle Schritte dazwischen werden übersprungen. Dieser Schritt wird als fehlgeschlagen gemeldet, der letzte Schritt meldet den Status des Vorgangsergebnisses, und alle dazwischen liegenden Schritte werden als erfolgreich gemeldet.

`AWS-RunPatchBaselineWithHooks`-Parameter

`AWS-RunPatchBaselineWithHooks` unterstützt sechs Parameter.

Der Parameter `Operation` muss angegeben werden.

Die Parameter `RebootOption`, `PreInstallHookDocName`, `PostInstallHookDocName` und `OnExitHookDocName` sind optional.

`Snapshot-ID` ist eigentlich optional, wir empfehlen jedoch, einen benutzerdefinierten Wert dafür anzugeben, wenn Sie `AWS-RunPatchBaselineWithHooks` außerhalb eines Wartungsfensters ausführen. Lassen Sie Patch Manager den Wert automatisch angeben, wenn das Dokument als Teil einer Wartungsfenster-Operation ausgeführt wird.

Parameter

- Parametername: Operation
- Parametername: Snapshot ID
- Parametername: RebootOption
- Parametername: PreInstallHookDocName
- Parametername: PostInstallHookDocName
- Parametername: OnExitHookDocName

Parametername: **`Operation`**

Nutzung: erforderlich.

Optionen: `Scan` | `Install`.

Scan

Wenn Sie die Option `Scan` wählen, verwende das System das Dokument `AWS-RunPatchBaseline`, um den Patch-Compliance-Zustand des verwalteten Knoten zu bestimmen und diese Informationen an Patch Manager zu melden. `Scan` fordert nicht zum Installieren von Updates oder zum Neustarten von verwalteten Knoten auf. Stattdessen erkennt die Operation, wo für den Knoten genehmigte und geeignete Updates fehlen.

Installieren

Bei Auswahl der Option `Install` versucht `AWS-RunPatchBaselineWithHooks`, die genehmigten und geeigneten Updates zu installieren, die auf dem verwalteten Knoten fehlen. Patch-Compliance-Informationen, die als Teil eines `Install`-Vorgangs generiert wurden, enthalten keine fehlenden Updates, melden allerdings möglicherweise Updates im Fehlerzustand, wenn die Installation des Updates aus einem beliebigen Grund nicht erfolgreich war. Immer wenn ein Update auf einem verwalteten Knoten installiert wird, wird der Knoten neu gestartet, um sicherzustellen, dass das Update installiert und aktiviert ist. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaselineWithHooks` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

> **Note**
>
> Wenn ein von den Basisregeln festgelegter Patch installiert wird, bevor der Patch Manager den verwalteten Knoten aktualisiert, wird das System möglicherweise nicht wie erwartet neu gestartet. Dies kann passieren, wenn ein Patch manuell von einem Benutzer oder automatisch von einem anderen Programm, z. B. dem `unattended-upgrades`-Paket auf Ubuntu Server, installiert wird.

Parametername: **`Snapshot ID`**

Nutzung: optional.

`Snapshot ID` ist eine eindeutige ID (GUID), die von Patch Manager verwendet wird, um sicherzustellen, dass ein Satz von verwalteten Knoten, für die in einer einzelnen Operation Patches durchgeführt werden, den genau gleichen Satz genehmigter Patches aufweist. Auch wenn der Parameter als optional definiert ist, hängen unsere Empfehlungen für bewährte Methoden davon ab, ob Sie `AWS-RunPatchBaselineWithHooks` in einem Wartungsfenster, wie in der folgenden Tabelle beschrieben, ausführen.

Bewährte Methoden für **`AWS-RunPatchBaselineWithHooks`**

Mode	Bewährte Methode	Details
Ausführen von `AWS-RunPatchBaselineWithHooks` innerhalb eines Wartungsfensters	Geben Sie keine Snapshot ID an. Patch Manager wird sie für Sie bereitstellen.	Falls Sie ein Wartungsfenster zum Ausführen von `AWS-RunPatchBaselineWithHooks` verwenden, dürfen Sie Ihre eigene generierte Snapshot ID nicht angeben. In diesem Szenario stellt Systems Manager einen GUID-Wert auf Grundlage der Wartungsfensterausführungs-ID bereit. Auf diese Weise wird sichergestellt, dass eine richtige ID für alle Aufrufe von `AWS-RunPatchBaselineWithHooks` in diesem Wartungsfenster verwendet wird. Wenn Sie einen Wert in diesem Szenario angeben, beachten Sie, dass der Snapshot der Patch-Baseline möglicherweise nicht länger als drei Tagen erhalten bleibt. Danach wird ein neuer Snapshot erstellt, auch wenn Sie dieselbe ID angeben,

Mode	Bewährte Methode	Details
		nachdem der Snapshot abgelaufen ist.

Mode	Bewährte Methode	Details
Ausführen von `AWS-RunPatchBaselineWithHooks` außerhalb eines Wartungsfensters	Generieren Sie einen benutzerdefinierten GUID-Wert für die Snapshot-ID und geben Sie ihn an.[1]	Wenn Sie kein Wartungsfenster zum Ausführen von `AWS-RunPatchBaselineWithHooks` verwenden, empfehlen wir, dass Sie eine eindeutige Snapshot-ID für jede Patch-Baseline generieren und angeben, insbesondere wenn Sie das Dokument `AWS-RunPatchBaselineWithHooks` auf mehreren verwaltete Knoten in derselben Operation ausführen. Wenn Sie keine ID in diesem Szenario angeben, generiert Systems Manager eine andere Snapshot-ID für jeden verwalteten Knoten, an den der Befehl gesendet wird. Dies kann zu unterschiedlichen Sätzen von Patches führen, die auf den Knoten angegeben sind. Angenommen, Sie führen das Dokument `AWS-RunPatchBaselineWithHooks` direkt über Run Command aus, ein Tool in AWS Systems Manager, und richten es auf eine Gruppe von 50 verwalteten Knoten aus. Das Angeben einer benutzerdefinierten Snapshot-ID führt zur Erstellung eines

Mode	Bewährte Methode	Details
		einzelnen Baseline-Snapshots , der verwendet wird, um alle verwaltete Knoten zu bewerten und zu patchen. Dadurch wird gewährlei stet, dass sie letztendlich einen konsistenten Zustand aufweisen.

[1] Sie können jedes beliebige Tool zum Generieren eines Werts für den Snapshot-ID-Parameter verwenden, das eine GUID generieren kann. In können Sie PowerShell beispielsweise das `New-Guid` Cmdlet verwenden, um eine GUID im Format von zu generieren. `12345699-9405-4f69-bc5e-9315aEXAMPLE`

Parametername: **`RebootOption`**

Nutzung: optional.

Optionen: `RebootIfNeeded` | `NoReboot`

Standardwert: `RebootIfNeeded`

> **Warning**
>
> Die Standardoption ist `RebootIfNeeded`. Stellen Sie sicher, dass Sie die richtige Option für Ihren Anwendungsfall auswählen. Wenn Ihre verwalteten Knoten beispielsweise sofort neu gestartet werden müssen, um einen Konfigurationsprozess abzuschließen, wählen Sie `RebootIfNeeded` aus. Oder wenn Sie die Verfügbarkeit von verwalteten Knoten bis zu einer geplanten Neustartzeit beibehalten müssen, wählen Sie `NoReboot` aus.

> **Important**
>
> Wir empfehlen nicht, Cluster-Instances in Amazon EMR (früher Amazon Elastic MapReduce genannt) zum Patchen zu verwendenPatch Manager. Wählen Sie insbesondere nicht die Option `RebootIfNeeded` für den Parameter `RebootOption` aus. (Diese Option ist

in den SSM-Befehlsdokumenten für das Patchen von `AWS-RunPatchBaseline`, `AWS-RunPatchBaselineAssociation` und `AWS-RunPatchBaselineWithHooks` verfügbar.) Die zugrunde liegenden Befehle für das Patchen mithilfe von Patch Manager verwenden `yum`- und `dnf`-Befehle. Daher führen die Operationen aufgrund der Art und Weise, wie Pakete installiert werden, zu Inkompatibilitäten. Informationen zu den bevorzugten Methoden für die Aktualisierung von Software auf Amazon-EMR-Clustern finden Sie unter Verwenden des Standard-AMI für Amazon EMR im Amazon EMR Management Guide.

RebootIfNeeded

Wenn Sie die Option `RebootIfNeeded` auswählen, wird der verwaltete Knoten in einem der folgenden Fälle neu gestartet:

- Patch Manager ist auf einem oder mehreren Patches installiert.

  Patch Manager wertet nicht aus, ob ein Neustart vom Patch erfordert wird. Das System wird neu gestartet, auch wenn der Patch keinen Neustart erfordert.

- Patch Manager erkennt ein oder mehrere Patches mit dem Status `INSTALLED_PENDING_REBOOT` während der `Install`-Operation.

  Der `INSTALLED_PENDING_REBOOT`-Status kann bedeuten, dass die Option `NoReboot` ausgewählt wurde, als der `Install`-Vorgang das letzte Mal ausgeführt wurde, oder dass ein Patch außerhalb von Patch Manager seit dem letzten Neustart des verwalteten Knotens installiert wurde.

Durch den Neustart von verwalteten Knoten wird in diesen beiden Fällen sichergestellt, dass aktualisierte Pakete aus dem Speicher gelöscht werden und das Patch- und Neustartverhalten über alle Betriebssysteme hinweg konsistent bleibt.

NoReboot

Wenn Sie die Option `NoReboot` auswählen, startet Patch Manager einen verwalteten Knoten nicht neu, selbst wenn über ihn während der `Install`-Operation Patches installiert wurden. Diese Option ist nützlich, wenn Sie wissen, dass für Ihre verwalteten Knoten nach dem Anwenden von Patches kein Neustart erforderlich ist oder Anwendungen bzw. Prozesse auf einem Knoten ausgeführt werden, die nicht durch einen Neustart beim Patchen unterbrochen werden sollten. Sie ist auch nützlich, wenn Sie mehr Kontrolle über das Timing von Neustarts von verwalteten Knoten wünschen, z. B. durch die Verwendung eines Wartungsfensters.

> **Note**
>
> Wenn Sie die Option `NoReboot` auswählen und ein Patch installiert ist, wird dem Patch der Status `InstalledPendingReboot` zugewiesen. Der verwaltete Knoten selbst wird jedoch als `Non-Compliant` gekennzeichnet. Nach einem Neustart und Ausführung einer `Scan`-Operation wird der Knoten-Status in `Compliant` aktualisiert.

Datei zum Nachverfolgen der Patch-Installation: Um die Patch-Installation nachzuverfolgen, insbesondere von Patches, die seit dem letzten Neustart des Systems installiert wurden, erstellt Systems Manager eine Datei auf dem verwalteten Knoten.

> **Important**
>
> Löschen oder ändern Sie die Tracking-Datei nicht. Wenn diese Datei gelöscht oder beschädigt wird, ist der Patch-Compliance-Bericht für den verwalteten Knoten ungenau. Starten Sie in diesem Fall den Knoten neu und führen Sie eine Patch-Scan-Operation aus, um die Datei wiederherzustellen.

Diese Tracking-Datei wird an den folgenden Speicherorten auf Ihren verwalteten Knoten gespeichert:

- Linux-Betriebssysteme:
  - `/var/log/amazon/ssm/patch-configuration/patch-states-configuration.json`
  - `/var/log/amazon/ssm/patch-configuration/patch-inventory-from-last-operation.json`
- Windows Server-Betriebssystem:
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchStatesConfiguration.json`
  - `C:\ProgramData\Amazon\PatchBaselineOperations\State\PatchInventoryFromLastOperation.json`

Parametername: **`PreInstallHookDocName`**

Nutzung: optional.

Standard: `AWS-Noop`.

Der Wert, der für den `PreInstallHookDocName`-Parameter anzugeben ist, ist der Name oder der Amazon-Ressourcenname (ARN) eines SSM-Dokuments Ihrer Wahl. Sie können den Namen eines AWS verwalteten Dokuments oder den Namen oder ARN eines benutzerdefinierten SSM-Dokuments angeben, das Sie erstellt haben oder das für Sie freigegeben wurde. (Für ein SSM-Dokument, das von einem anderen für Sie freigegeben wurde AWS-Konto, müssen Sie den vollständigen Ressourcen-ARN angeben, z. `arn:aws:ssm:us-east-2:123456789012:document/MySharedDocument` B.)

Das von Ihnen angegebene SSM-Dokument wird vor der `Install`-Operation ausgeführt und führt alle Aktionen aus, die von SSM Agent unterstützt werden, z. B. ein Shell-Skript, um die Zustandsprüfung der Anwendung zu überprüfen, bevor der verwaltete Knoten gepatcht wird. (Eine Liste der Aktionen finden Sie unter Referenz für Befehlsdokument-Plugins). Der SSM-Dokumentname ist standardmäßig `AWS-Noop`, was keine Operation für den verwalteten Knoten ausführt.

Informationen zum Erstellen eines benutzerdefinierten SSM-Dokuments finden Sie unter Erstellen von SSM-Dokumentinhalten.

Parametername: **`PostInstallHookDocName`**

Nutzung: optional.

Standard: `AWS-Noop`.

Der Wert, der für den `PostInstallHookDocName`-Parameter anzugeben ist, ist der Name oder der Amazon-Ressourcenname (ARN) eines SSM-Dokuments Ihrer Wahl. Sie können den Namen eines AWS verwalteten Dokuments oder den Namen oder ARN eines benutzerdefinierten SSM-Dokuments angeben, das Sie erstellt haben oder das für Sie freigegeben wurde. (Für ein SSM-Dokument, das von einem anderen für Sie freigegeben wurde AWS-Konto, müssen Sie den vollständigen Ressourcen-ARN angeben, z. `arn:aws:ssm:us-east-2:123456789012:document/MySharedDocument` B.)

Das von Ihnen angegebene SSM-Dokument wird nach der `Install with NoReboot`-Operation ausgeführt und führt alle Aktionen aus, die von SSM Agent unterstützt werden, z. B. ein Shell-Skript zum Installieren von Updates von Drittanbietern vor dem Neustart. (Eine Liste der Aktionen finden Sie unter Referenz für Befehlsdokument-Plugins). Der SSM-Dokumentname ist standardmäßig `AWS-Noop`, was keine Operation für den verwalteten Knoten ausführt.

Informationen zum Erstellen eines benutzerdefinierten SSM-Dokuments finden Sie unter Erstellen von SSM-Dokumentinhalten.

Parametername: **`OnExitHookDocName`**

Nutzung: optional.

Standard: `AWS-Noop`.

Der Wert, der für den `OnExitHookDocName`-Parameter anzugeben ist, ist der Name oder der Amazon-Ressourcenname (ARN) eines SSM-Dokuments Ihrer Wahl. Sie können den Namen eines AWS verwalteten Dokuments oder den Namen oder ARN eines benutzerdefinierten SSM-Dokuments angeben, das Sie erstellt haben oder das für Sie freigegeben wurde. (Für ein SSM-Dokument, das aus einem anderen AWS-Konto freigegeben wurde, müssen Sie den vollständigen Ressourcen-ARN angeben, z. B. `arn:aws:ssm:us-east-2:123456789012:document/MySharedDocument`.)

Das von Ihnen angegebene SSM-Dokument wird nach dem Neustart des verwalteten Knoten ausgeführt und führt alle Aktionen aus, die von SSM Agent unterstützt werden, z. B. ein Shell-Skript, um den Knoten-Zustand nach Abschluss des Patchvorgangs zu überprüfen. (Eine Liste der Aktionen finden Sie unter Referenz für Befehlsdokument-Plugins). Der SSM-Dokumentname ist standardmäßig `AWS-Noop`, was keine Operation für den verwalteten Knoten ausführt.

Informationen zum Erstellen eines benutzerdefinierten SSM-Dokuments finden Sie unter Erstellen von SSM-Dokumentinhalten.

Beispielszenario für die Verwendung des InstallOverrideList Parameters in **`AWS-RunPatchBaseline`** oder **`AWS-RunPatchBaselineAssociation`**

Sie können den Parameter `InstallOverrideList` verwenden, wenn Sie die von der aktuellen Standard-Patch-Baseline in Patch Manager, ein Tool in AWS Systems Manager, angegebenen Patches überschreiben möchten. In diesem Thema finden Sie Beispiele, die zeigen, wie Sie diesen Parameter verwenden, um Folgendes zu erreichen:

- Anwendung verschiedener Sätzen von Patches auf eine Zielgruppe von verwalteten Knoten.
- Anwendung dieser Patch-Sets auf verschiedene Häufigkeiten
- Verwendung derselben Patch-Baseline für beide Operationen

Angenommen, Sie möchten zwei verschiedene Kategorien von Patches auf Ihren von Amazon Linux 2 verwalteten Knoten installieren. Sie möchten diese Patches mithilfe von Wartungsfenstern nach verschiedenen Zeitplänen installieren. Sie möchten, dass jede Woche ein Wartungsfenster ausgeführt wird und alle `Security`-Patches installiert werden. Sie möchten, dass einmal im Monat

ein weiteres Wartungsfenster ausgeführt wird und dabei alle verfügbaren Patches oder Kategorien von Patches außer `Security` installiert werden.

Es kann jedoch nur jeweils eine Patch-Baseline als Standard für ein Betriebssystem definiert werden. Diese Anforderung hilft, Situationen zu vermeiden, in denen eine Patch-Baseline einen Patch genehmigt, während eine andere ihn blockiert, was zu Problemen zwischen in Konflikt stehenden Versionen führen kann.

Mit der folgenden Strategie können Sie den Parameter `InstallOverrideList` verwenden, um verschiedene Patch-Typen nach verschiedenen Zeitplänen auf eine Zielgruppe anzuwenden und dabei dennoch dieselbe Patch-Baseline zu verwenden:

1. Stellen Sie in der Standard-Patch-Baseline sicher, dass nur `Security`-Updates angegeben sind.
2. Erstellen Sie ein Wartungsfenster, das `AWS-RunPatchBaseline` oder `AWS-RunPatchBaselineAssociation` jede Woche ausführt. Geben Sie keine Überschreibungsliste an.
3. Erstellen Sie eine Überschreibungsliste der Patches aller Typen, die Sie monatlich anwenden möchten, und speichern Sie sie in einem Amazon Simple Storage Service (Amazon S3)-Bucket.
4. Erstellen Sie ein zweites Wartungsfenster, das einmal im Monat ausgeführt wird. Geben Sie für die Run Command-Aufgabe, die Sie für dieses Wartungsfenster registrieren, jedoch den Speicherort Ihrer Überschreibungsliste an.

Das Ergebnis: Jede Woche werden nur `Security`-Patches installiert, wie in Ihrer Standard-Patch-Baseline definiert. Die Installation aller verfügbaren Patches oder einer von Ihnen definierten Teilmenge von Patches erfolgt jeden Monat.

> **Note**
>
> Wenn Sie einen Knoten patchen, der nur verwendet IPv6, stellen Sie sicher, dass die angegebene URL vom Knoten aus erreichbar ist. Wenn die SSM Agent Konfigurationsoption auf gesetzt `UseDualStackEndpoint` ist`true`, wird ein Dual-Stack-S3-Client verwendet, wenn eine S3-URL bereitgestellt wird. Weitere Informationen the section called “Tutorial: Einen Server in einer IPv6 einzigen Umgebung patchen” zur Konfiguration des Agenten für die Verwendung von Dualstack finden Sie unter.

Weitere Informationen und Beispiellisten finden Sie unter Parametername: `InstallOverrideList`.

Verwenden des BaselineOverride -Parameters

Sie können Patching-Voreinstellungen zur Laufzeit mit der Baseline-Überschreibungsfunktion in Patch Manager definieren, einem Tool in AWS Systems Manager. Geben Sie dazu einen Amazon Simple Storage Service (Amazon S3)-Bucket an, der ein JSON-Objekt mit einer Liste mit Patch-Baselines enthält. Beim Patchvorgang werden die im JSON-Objekt bereitgestellten Baselines verwendet, die mit dem Hostbetriebssystem übereinstimmen, anstatt die Regeln aus der Standard-Patch-Baseline anzuwenden.

> **Important**
>
> Der `BaselineOverride`-Dateiname darf die folgenden Zeichen nicht enthalten: Backtick (`), einfaches Anführungszeichen ('), doppeltes Anführungszeichen („) und Dollarzeichen ($).

Wenn Sie Patch-Policy verwenden, wird die Patch-Compliance der im `BaselineOverride`-Parameter angegebenen Baseline nicht überschrieben. Die Ausgabe wird in den Stdout-Protokollen von Run Command aufgezeichnet, einem Tool in AWS Systems Manager. Die Ergebnisse drucken nur Pakete aus, die als `NON_COMPLIANT` gekennzeichnet sind. Das bedeutet, dass das Paket als `Missing`, `Failed`, `InstalledRejected` oder `InstalledPendingReboot` gekennzeichnet ist.

Wenn ein Patch-Vorgang jedoch eine Patch-Richtlinie verwendet, übergibt das System den Override-Parameter aus dem zugehörigen S3-Bucket, und der Compliance-Wert wird für den verwalteten Knoten aktualisiert. Weitere Informationen zum Patch-Richtlinienverhalten finden Sie unter Patch-Richtlinienkonfigurationen in Quick Setup.

> **Note**
>
> Wenn Sie einen Knoten patchen, der nur verwendet IPv6, stellen Sie sicher, dass die angegebene URL vom Knoten aus erreichbar ist. Wenn die SSM Agent Konfigurationsoption auf gesetzt `UseDualStackEndpoint` ist`true`, wird ein Dual-Stack-S3-Client verwendet, wenn eine S3-URL bereitgestellt wird. Weitere Informationen the section called “Tutorial: Einen Server in einer IPv6 einzigen Umgebung patchen” zur Konfiguration des Agenten für die Verwendung von Dualstack finden Sie unter.

Verwenden der Patch-Baseline-Überschreibung mit den Parametern „Snapshot-ID“ oder „Install Override List“

Es gibt zwei Fälle, in denen die Patch-Baseline-Überschreibung ein bemerkenswertes Verhalten aufweist.

Gleichzeitiges Verwenden von Baseline-Überschreiben und Snapshot-ID

Snapshot-IDs stellen sicher, dass alle verwalteten Knoten in einem bestimmten Patching-Befehl dasselbe anwenden. Wenn Sie beispielsweise 1 000 Knoten gleichzeitig patchen, sind die Patches identisch.

Wenn Sie sowohl eine Snapshot-ID als auch eine Patch-Baseline-Überschreibung verwenden, hat die Snapshot-ID Vorrang vor der Patch-Baseline-Überschreibung. Die Baseline-Überschreibungsregeln werden weiterhin verwendet, aber sie werden nur einmal ausgewertet. Im vorangegangenen Beispiel sind die Patches für Ihre 1 000 verwaltete Knoten immer gleich. Wenn Sie in der Mitte des Patching-Vorgangs die JSON-Datei im referenzierten S3-Bucket auf etwas anderes geändert haben, sind die angewendeten Patches immer noch identisch. Dies liegt daran, dass die Snapshot-ID bereitgestellt wurde.

Gleichzeitiges Verwenden der Baseline-Überschreibung und des Parameters „Override List“

Sie können diese beiden Parameter nicht gleichzeitig verwenden. Das Patching-Dokument schlägt fehl, wenn beide Parameter angegeben sind, und es führt keine Scans oder Installationen auf dem verwalteten Knoten durch.

Codebeispiele

Das folgende Codebeispiel für Python zeigt, wie die Patch-Baseline-Überschreibung generiert wird.

```
import boto3
import json

ssm = boto3.client('ssm')
s3 = boto3.resource('s3')
s3_bucket_name = 'my-baseline-override-bucket'
s3_file_name = 'MyBaselineOverride.json'
baseline_ids_to_export = ['pb-0000000000000000', 'pb-0000000000000001']

baseline_overrides = []
for baseline_id in baseline_ids_to_export:
    baseline_overrides.append(ssm.get_patch_baseline(
```

```
        BaselineId=baseline_id
    ))

json_content = json.dumps(baseline_overrides, indent=4, sort_keys=True, default=str)
s3.Object(bucket_name=s3_bucket_name, key=s3_file_name).put(Body=json_content)
```

So wird eine Patch-Baseline-Überschreibung wie die folgende erstellt.

```
[
    {
        "ApprovalRules": {
            "PatchRules": [
                {
                    "ApproveAfterDays": 0,
                    "ComplianceLevel": "UNSPECIFIED",
                    "EnableNonSecurity": false,
                    "PatchFilterGroup": {
                        "PatchFilters": [
                            {
                                "Key": "PRODUCT",
                                "Values": [
                                    "*"
                                ]
                            },
                            {
                                "Key": "CLASSIFICATION",
                                "Values": [
                                    "*"
                                ]
                            },
                            {
                                "Key": "SEVERITY",
                                "Values": [
                                    "*"
                                ]
                            }
                        ]
                    }
                }
            ]
        },
        "ApprovedPatches": [],
        "ApprovedPatchesComplianceLevel": "UNSPECIFIED",
```

```
        "ApprovedPatchesEnableNonSecurity": false,
        "GlobalFilters": {
            "PatchFilters": []
        },
        "OperatingSystem": "AMAZON_LINUX_2",
        "RejectedPatches": [],
        "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY",
        "Sources": []
    },
    {
        "ApprovalRules": {
            "PatchRules": [
                {
                    "ApproveUntilDate": "2021-01-06",
                    "ComplianceLevel": "UNSPECIFIED",
                    "EnableNonSecurity": true,
                    "PatchFilterGroup": {
                        "PatchFilters": [
                            {
                                "Key": "PRODUCT",
                                "Values": [
                                    "*"
                                ]
                            },
                            {
                                "Key": "CLASSIFICATION",
                                "Values": [
                                    "*"
                                ]
                            },
                            {
                                "Key": "SEVERITY",
                                "Values": [
                                    "*"
                                ]
                            }
                        ]
                    }
                }
            ]
        },
        "ApprovedPatches": [
            "open-ssl*"
        ],
```

```
        "ApprovedPatchesComplianceLevel": "UNSPECIFIED",
        "ApprovedPatchesEnableNonSecurity": false,
        "GlobalFilters": {
            "PatchFilters": []
        },
        "OperatingSystem": "SUSE",
        "RejectedPatches": [
            "python*"
        ],
        "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY",
        "Sources": []
    }
]
```

Patch-Baselines

In den Themen in diesem Abschnitt finden Sie Informationen zur Funktionsweise von Patch-Baselines in Patch Manager, einem Tool in AWS Systems Manager, wenn Sie einen `Scan`- oder `Install`-Vorgang auf Ihren verwalteten Knoten ausführen.

Themen

- Vordefinierte und benutzerdefinierte Patch-Baselines
- Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen
- Patch-Gruppen
- Informationen zum Patchen von Anwendungen, die von Microsoft unter Windows Server veröffentlicht wurden

Vordefinierte und benutzerdefinierte Patch-Baselines

Patch Manager, ein Tool in AWS Systems Manager, bietet vordefinierte Patch-Baselines für jedes der Betriebssysteme, die von unterstützt werden. Patch Manager Sie können diese Baselines in ihrer aktuellen Konfiguration verwenden (eine Anpassung ist nicht möglich) oder eigene benutzerdefinierte Patch-Baselines erstellen. Benutzerdefinierte Patch-Baselines ermöglichen Ihnen eine bessere Kontrolle darüber, welche Patches für Ihre Umgebung genehmigt oder abgelehnt werden. Außerdem weisen die vordefinierten Baselines allen Patches, die mit diesen Baselines installiert wurden, die Compliance-Ebene `Unspecified` zu. Für die Zuweisung von Compliance-Werten können Sie eine Kopie einer vordefinierten Baseline erstellen und die Compliance-Werte angeben, die Patches zugewiesen werden sollen. Weitere Informationen erhalten Sie unter Benutzerdefinierte Baselines und Arbeiten mit benutzerdefinierten Patch-Baselines.

> **Note**
>
> Die Informationen in diesem Thema gelten unabhängig davon, welche Methode oder Art der Konfiguration Sie für Ihren Patching-Vorgang verwenden:
>
> - Eine in Quick Setup konfigurierte Patch-Richtlinie
> - Eine in Quick Setup konfigurierte Host-Management-Option
> - Ein Wartungsfenster zum Ausführen eines Patch-`Scan` oder einer `Install`-Aufgabe
> - Ein On-Demand-Jetzt patchen-Vorgang

Themen

- Vordefinierte Baselines
- Benutzerdefinierte Baselines

Vordefinierte Baselines

Die folgende Tabelle beschreibt die mit Patch Manager bereitgestellten vordefinierten Patch-Baselines.

Informationen dazu, welche Versionen der einzelnen Betriebssysteme von Patch Manager unterstützt werden, finden Sie unter Patch Manager-Voraussetzungen.

Name	Unterstütztes Betriebssystem	Details
`AWS-AlmaLinuxDefaultPatchBaseline`	AlmaLinux	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“ und dem Schweregrad „Critical“ oder „Important“. Genehmigt außerdem alle Patches mit der Klassifizierung „Bugfix“. Patches werden sieben Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[1]

Name	Unterstütztes Betriebssystem	Details
`AWS-AmazonLinux2DefaultPatchBaseline`	Amazon Linux 2	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“ und dem Schweregrad „Critical“ oder „Important“. Genehmigt außerdem alle Patches mit der Klassifizierung „Bugfix“. Patches werden automatisch sieben Tage nach der Veröffentlichung genehmigt.[1]
`AWS-AmazonLinux2023DefaultPatchBaseline`	Amazon Linux 2023	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“ und dem Schweregrad „Critical“ oder „Important“. Patches werden automatisch sieben Tage nach der Veröffentlichung genehmigt. Genehmigt außerdem alle Patches mit einer Klassifizierung „Bugfix“ sieben Tage nach der Veröffentlichung.
`AWS-CentOSDefaultPatchBaseline`	CentOS Stream	Genehmigt alle Aktualisierungen sieben Tage nach ihrer Verfügbarkeit, einschließlich nicht sicherheitsrelevanter Aktualisierungen.

Name	Unterstütztes Betriebssystem	Details
`AWS-DebianDefaultPatchBaseline`	Debian Server	Genehmigt sofort alle sicherheitsrelevanten Patches für Betriebssysteme mit der Priorität „Required“, „Important“, „Standard“, „Optional“ oder „Extra“. Die Genehmigung erfolgt unverzüglich, weil in den Repositorys keine zuverlässigen Datumsangaben zur Veröffentlichung verfügbar sind.
`AWS-MacOSDefaultPatchBaseline`	macOS	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“. Genehmigt auch alle Pakete mit einem aktuellen Update.
`AWS-OracleLinuxDefaultPatchBaseline`	Oracle Linux	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“ und dem Schweregrad „Important“ oder „Moderate“. Genehmigt außerdem alle als „Bugfix“ eingestuften Patches 7 Tage nach Veröffentlichung. Patches werden sieben Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[1]

Name	Unterstütztes Betriebssystem	Details
`AWS-RedHatDefaultPatchBaseline`	Red Hat Enterprise Linux (RHEL)	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“ und dem Schweregrad „Critical“ oder „Important“. Genehmigt außerdem alle Patches mit der Klassifizierung „Bugfix“. Patches werden sieben Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[1]
`AWS-RockyLinuxDefaultPatchBaseline`	Rocky Linux	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“ und dem Schweregrad „Critical“ oder „Important“. Genehmigt außerdem alle Patches mit der Klassifizierung „Bugfix“. Patches werden sieben Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[1]
`AWS-SuseDefaultPatchBaseline`	SUSE Linux Enterprise Server (SLES)	Genehmigt alle Betriebssystem-Patches mit der Klassifizierung „Security“ und dem Schweregrad „Critical“ oder „Important“. Patches werden sieben Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[1]

Name	Unterstütztes Betriebssystem	Details
`AWS-UbuntuDefaultPatchBaseline`	Ubuntu Server	Genehmigt sofort alle sicherheitsrelevanten Patches für Betriebssysteme mit der Priorität „Required“, „Important“, „Standard“, „Optional“ oder „Extra“. Die Genehmigung erfolgt unverzüglich, weil in den Repositorys keine zuverlässigen Datumsangaben zur Veröffentlichung verfügbar sind.
`AWS-DefaultPatchBaseline`	Windows Server	Genehmigt alle Windows Server Betriebssystem-Patches, die als "" oder CriticalUpdates SecurityUpdates "eingestuft sind und die den MSRC-Schweregrad „Kritisch“ oder „Wichtig“ haben. Patches werden 7 Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[2]
`AWS-WindowsPredefinedPatchBaseline-OS`	Windows Server	Genehmigt alle Windows Server Betriebssystem-Patches, die als "oder" CriticalUpdates "eingestuft sind und die den MSRC-Schweregrad „Kritisch“ oder „Wichtig“ haben. SecurityUpdates Patches werden 7 Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[2]

Name	Unterstütztes Betriebssystem	Details
`AWS-WindowsPredefinedPatchBaseline-OS-Applications`	Windows Server	Genehmigt für das Windows Server Betriebssystem alle Patches, die als "oder CriticalUpdatesSecurityUpdates" eingestuft sind und die den MSRC-Schweregrad „Kritisch“ oder „Wichtig“ haben. Genehmigt für von Microsoft veröffentlichte Anwendungen alle Patches. Patches für Betriebssysteme und Anwendungen werden 7 Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt.[2]

[1] Für Amazon Linux 2 wird die 7-Tage-Wartezeit, bevor Patches automatisch genehmigt werden, aus einem `Updated Date`-Wert in `updateinfo.xml` und nicht aus einem `Release Date`-Wert berechnet. Verschiedene Faktoren können den `Updated Date`-Wert beeinflussen. Andere Betriebssysteme behandeln Veröffentlichungs- und Aktualisierungsdaten unterschiedlich. Informationen dazu, wie Sie unerwartete Ergebnisse durch Verzögerungen bei der automatischen Genehmigung vermeiden können, finden Sie unter So werden Veröffentlichungs- und Aktualisierungsdaten von Paketen berechnet.

[2] Für Windows Server enthalten die Standard-Baselines eine Verzögerung von 7 Tagen für die automatische Genehmigung. Um einen Patch innerhalb von 7 Tagen nach der Veröffentlichung zu installieren, müssen Sie eine benutzerdefinierte Baseline erstellen.

Benutzerdefinierte Baselines

Mithilfe der folgenden Informationen können Sie benutzerdefinierte Patch-Baselines erstellen, um Ihre Patching-Ziele zu erreichen.

Themen

- Automatische Genehmigungen in benutzerdefinierten Baselines verwenden

- Zusätzliche Informationen zum Erstellen von Patch-Baselines

Automatische Genehmigungen in benutzerdefinierten Baselines verwenden

Wenn Sie eine eigene Patch-Baseline herstellen, können Sie die Patches wahlweise automatisch genehmigen, indem Sie die folgenden Kategorien verwenden.

- Betriebssystem: Unterstützte Versionen von Windows Server, Amazon Linux, Ubuntu Server, usw.
- Produktname (für Betriebssysteme): Beispielsweise RHEL 7.5, Amazon Linux 2023 2023.8.20250808, Windows Server 2012, Windows Server 2012 R2 usw.
- Produktname (Windows Servernur für von Microsoft veröffentlichte Anwendungen): Zum Beispiel Word 2016, BizTalk Server usw.
- Klassifizierung: Beispielsweise kritische Updates, Sicherheitsupdates usw.
- Schweregrad: Beispielsweise kritisch, wichtig usw.

Für jede von Ihnen erstellte Genehmigungsregel können Sie eine Verzögerung für die automatische Genehmigung oder ein Stichdatum für die Patch-Genehmigung angeben.

> **Note**
>
> Da es nicht möglich ist, die Veröffentlichungsdaten von Updatepaketen für Ubuntu Server zuverlässig zu bestimmen, werden die Optionen für die automatische Genehmigung für dieses Betriebssystem nicht unterstützt.

Eine Verzögerung der automatischen Genehmigung ist die Anzahl an Tagen, die gewartet werde soll, nachdem die Patch veröffentlicht oder zuletzt aktualisiert wurde, bevor der Patch automatisch genehmigt wird. Wenn Sie beispielsweise eine Regel mit der `CriticalUpdates`-Klassifizierung erstellen und für sie für eine Verzögerung der automatischen Genehmigung von sieben Tagen konfigurieren, wird ein neuer kritischer Patch, der am 7. Juli veröffentlicht wird, am 14. Juli automatisch genehmigt.

Wenn ein Linux-Repository keine Informationen zum Veröffentlichungsdatum von Paketen bereitstellt, verwendet Patch Manager den Erstellungszeitpunkt des Pakets als Datum zur automatischen Genehmigung der Datumsangaben für Amazon Linux 2, Amazon Linux 2023 und Red Hat Enterprise Linux (RHEL). Wenn die Erstellungszeit des Pakets nicht bestimmt werden kann, verwendet Patch Manager als Standarddatum den 1. Januar 1970. Dies führt dazu, dass alle Angaben zum Datum

der automatischen Genehmigung in Patch-Baselines von Patch Manager umgangen werden, die so konfiguriert sind, dass Patches für jedes Datum nach dem 1. Januar 1970 genehmigt werden.

Wenn Sie ein Stichdatum für die automatische Genehmigung angeben, wendet Patch Manager automatisch alle Patches an, die an oder vor diesem Datum veröffentlicht oder zuletzt aktualisiert wurden. Wenn Sie beispielsweise den 07. Juli 2023 als Stichtag angeben, werden keine Patches automatisch installiert, die an oder nach dem 08. Juli 2023 veröffentlicht oder zuletzt aktualisiert wurden.

Wenn Sie eine benutzerdefinierte Patch-Baseline erstellen, können Sie für Patches, die von dieser Patch-Baseline genehmigt wurden, einen Schweregrad für die Konformität angeben, beispielsweise `Critical` oder `High`. Wenn der Patch-Status eines genehmigten Patches als `Missing` gemeldet wird, dann ist der insgesamt gemeldete Konformitätsschweregrad der Patch-Baseline der von Ihnen angegebene Schweregrad.

Zusätzliche Informationen zum Erstellen von Patch-Baselines

Beachten Sie bei der Erstellung einer Patch-Baseline Folgendes:

- Patch Manager stellt eine vordefinierte Patch-Baseline für jedes unterstützte Betriebssystem bereit. Diese vordefinierten Patch-Baselines werden als Standard-Patch-Baselines für alle Betriebssystemtypen verwendet, wenn Sie nicht eigene Patch-Baselines erstellen und diese als Standard für den jeweiligen Betriebssystemtyp festlegen.

  > **Note**
  >
  > Für Windows Server werden drei vordefinierte Patch-Baselines bereitgestellt. Die Patch-Baselines `AWS-DefaultPatchBaseline` und `AWS-WindowsPredefinedPatchBaseline-OS` unterstützen nur Betriebssystemupdates auf dem Windows-Betriebssystem selbst. `AWS-DefaultPatchBaseline` wird als Standard-Patch-Baseline für von Windows Server verwalteten Knoten verwendet, es sei denn, Sie geben eine andere Patch-Baseline an. Die Konfigurationseinstellungen in diesen beiden Patch-Baselines sind identisch. Die neuere der beiden, `AWS-WindowsPredefinedPatchBaseline-OS`, wurde erstellt, um sie von der dritten vordefinierten Patch-Baseline für Windows Server zu unterscheiden. Diese Patch-Baseline, `AWS-WindowsPredefinedPatchBaseline-OS-Applications`, kann verwendet werden, um Patches sowohl auf das Windows Server-Betriebssystem als auch auf unterstützte Anwendungen, die von Microsoft veröffentlicht wurden, anzuwenden.

- Standardmäßig entfernen Windows Server 2019 und Windows Server 2022 Updates, die durch spätere Updates ersetzt werden. Wenn Sie also den `ApproveUntilDate`-Parameter in einer Windows Server-Patch-Baseline verwenden, das im `ApproveUntilDate`-Parameter ausgewählte Datum jedoch vor dem Datum des neuesten Patches liegt, wird der neue Patch nicht installiert, wenn der Patching-Vorgang ausgeführt wird. Weitere Informationen zu Windows Server-Patching-Regeln, finden Sie in der Registerkarte Windows Server in Wie Sicherheitspatches ausgewählt werden.

  Das bedeutet, dass der verwaltete Knoten die Anforderungen des Systems-Manager-Betriebs erfüllt, auch wenn ein kritischer Patch aus dem Vormonat möglicherweise nicht installiert wurde. Das gleiche Szenario kann bei Verwendung des `ApproveAfterDays`-Parameters auftreten. Aufgrund des Verhaltens von Microsoft bei abgelösten Patches ist es möglich, eine Zahl (im Allgemeinen größer als 30 Tage) festzulegen, sodass Patches für Windows Server nie installiert werden, wenn der neueste verfügbare Patch von Microsoft veröffentlicht wird, bevor die Anzahl der Tage in `ApproveAfterDays` verstrichen ist.
- Nur für Windows Server kann ein verfügbarer Sicherheitsupdate-Patch, der nicht von der Patch-Baseline genehmigt wurde, den Konformitätswert `Compliant` oder `Non-Compliant` haben, wie in einer benutzerdefinierten Patch-Baseline definiert.

  Wenn Sie eine Patch-Baseline erstellen oder aktualisieren, wählen Sie den Status, den Sie Sicherheitspatches zuweisen möchten, die zwar verfügbar, aber nicht genehmigt sind, weil sie die in der Patch-Baseline angegebenen Installationskriterien nicht erfüllen. Beispielsweise können Sicherheitspatches, die Sie möglicherweise installieren möchten, übersprungen werden, wenn Sie eine lange Wartezeit für die Installation nach der Veröffentlichung eines Patches angegeben haben. Wenn während der von Ihnen angegebenen Wartezeit ein Update für den Patch veröffentlicht wird, beginnt die Wartezeit für die Installation des Patches von vorne. Wenn die Wartezeit zu lang ist, können mehrere Versionen des Patches veröffentlicht, aber nie installiert werden.

  Wenn Sie die Konsole verwenden, um eine Patch-Baseline zu erstellen oder zu aktualisieren, geben Sie diese Option im Feld Compliance-Status für verfügbare Sicherheitsupdates an. Mit AWS CLI dem update-patch-baselineBefehl create-patch-baselineoder geben Sie diese Option im `available-security-updates-compliance-status` Parameter an.
- Patch ManagerVersucht bei lokalen Servern und virtuellen Maschinen (VMs), Ihre benutzerdefinierte Standard-Patch-Baseline zu verwenden. Wenn keine benutzerdefinierte Standard-Patch-Baseline vorhanden ist, verwendet das System die vordefinierte Patch-Baseline für das entsprechende Betriebssystem.

- Wenn ein Patch sowohl als genehmigt als auch als abgelehnt aufgelistet ist, wird der Patch abgelehnt.
- Für einen verwalteten Knoten kann nur eine einzige Patch-Baseline definiert werden.
- Die Formate der Paketnamen, die Sie zu den Listen der genehmigten und abgelehnten Patches für eine Patch-Baseline hinzufügen können, hängen von der Art des Betriebssystems ab, das gepatcht wird.

  Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen.
- Wenn Sie eine Patch-Richtlinienkonfiguration in Quick Setup verwenden, werden Aktualisierungen, die Sie an benutzerdefinierten Patch-Baselines vornehmen, einmal pro Stunde mit Quick Setup synchronisiert.

  Wenn eine benutzerdefinierte Patch-Baseline gelöscht wird, auf die in einer Patch-Richtlinie verwiesen wurde, wird auf der Seite mit den Quick Setup-Konfigurationsdetails ein Banner für Ihre Patch-Richtlinie angezeigt. Das Banner informiert Sie darüber, dass die Patch-Richtlinie auf eine nicht mehr vorhandene Patch-Baseline verweist und nachfolgende Patching-Vorgänge fehlschlagen werden. Kehren Sie in diesem Fall zur Seite Quick Setup-Konfigurationen zurück, wählen Sie die Patch Manager-Konfiguration aus und wählen Sie Aktionen, Konfiguration bearbeiten. Der Name der gelöschten Patch-Baseline wird hervorgehoben, und Sie müssen eine neue Patch-Baseline für das betroffene Betriebssystem auswählen.
- Wenn Sie eine Genehmigungsregel mit mehreren `Classification`- und `Severity`-Werten erstellen, werden Patches auf der Grundlage ihrer verfügbaren Attribute genehmigt. Pakete mit beiden `Classification`- und `Severity`-Attributen und entsprechen den ausgewählten Basiswerten für beide Felder. Pakete, die nur `Classification`-Attribute enthalten, werden nur mit den ausgewählten `Classification`-Basiswerten verglichen. Schweregradanforderungen in derselben Regel werden für Pakete ohne `Severity`-Attribute ignoriert.

Informationen zum Erstellen einer Patch-Baseline finden Sie unter Arbeiten mit benutzerdefinierten Patch-Baselines und Tutorial: Patchen Sie eine Serverumgebung mit dem AWS CLI.

Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen

Die Formate der Paketnamen, die Sie zu den Listen der genehmigten und abgelehnten Patches hinzufügen können, hängen von der Art des Betriebssystems ab, das gepatcht wird.

Paketnamen-Formate für Linux-Betriebssysteme

Die Formate, die Sie für genehmigte und abgelehnte Patches in der Patch-Baseline festlegen können, variieren je nach Linux-Typ. Genauer gesagt hängen die unterstützten Formate von dem Paket-Manager ab, der vom Linux-Betriebssystemtyp verwendet wird.

Themen

- Amazon Linux 2, Amazon Linux 2023, Oracle Linux und Red Hat Enterprise Linux (RHEL)
- Debian Server und Ubuntu Server
- SUSE Linux Enterprise Server (SLES)

Amazon Linux 2, Amazon Linux 2023, Oracle Linux und Red Hat Enterprise Linux (RHEL)

Paket-Manager: YUM, außer unter Amazon Linux 2023 und RHEL 8, die DNF als Paket-Manager verwenden.

Genehmigte Patches: Für genehmigte Patches können Sie Folgendes festlegen:

- Bugzilla IDs, im Format `1234567` (Das System verarbeitet Zeichenketten, die nur Zahlen enthalten, als Bugzilla.) IDs
- CVE IDs, im Format `CVE-2018-1234567`
- Beratung IDs, in Formaten wie und `RHSA-2017:0864` `ALAS-2018-123`
- Paketnamen, die unter Verwendung einer oder mehrerer der verfügbaren Komponenten für die Paketbenennung erstellt wurden. Zur Veranschaulichung lauten die Komponenten für das genannte `dbus.x86_64:1:1.12.28-1.amzn2023.0.1`-Paket wie folgt:
  - `name`: `dbus`
  - `architecture`: `x86_64`
  - `epoch`: `1`
  - `version`: `1.12.28`
  - `release`: `1.amzn2023.0.1`

  Paketnamen mit den folgenden Konstruktionen werden unterstützt:
  - `name`
  - `name.arch`
  - `name-version`

  - `name-version-release`
  - `name-version-release.arch`
  - `version`
  - `version-release`
  - `epoch:version-release`
  - `name-epoch:version-release`
  - `name-epoch:version-release.arch`
  - `epoch:name-version-release.arch`
  - `name.arch:epoch:version-release`

  Hier einige Beispiele:

  - `dbus.x86_64`
  - `dbus-1.12.28`
  - `dbus-1.12.28-1.amzn2023.0.1`
  - `dbus-1:1.12.28-1.amzn2023.0.1.x86_64`
- Wir unterstützen auch Paketnamenkomponenten mit einem einzigen Platzhalter in den oben genannten Formaten, z. B. in den folgenden:
  - `dbus*`
  - `dbus-1.12.2*`
  - `dbus-*:1.12.28-1.amzn2023.0.1.x86_64`

Abgelehnte Patches: Für abgelehnte Patches können Sie Folgendes festlegen:

- Paketnamen, die unter Verwendung einer oder mehrerer der verfügbaren Komponenten für die Paketbenennung erstellt wurden. Zur Veranschaulichung lauten die Komponenten für das genannte `dbus.x86_64:1:1.12.28-1.amzn2023.0.1`-Paket wie folgt:
  - `name`: `dbus`
  - `architecture`; `x86_64`
  - `epoch`: `1`
  - `version`: `1.12.28`
  - `release`: `1.amzn2023.0.1`

  Paketnamen mit den folgenden Konstruktionen werden unterstützt:

- `name`
- `name.arch`
- `name-version`
- `name-version-release`
- `name-version-release.arch`
- `version`
- `version-release`
- `epoch:version-release`
- `name-epoch:version-release`
- `name-epoch:version-release.arch`
- `epoch:name-version-release.arch`
- `name.arch:epoch:version-release`

  Hier einige Beispiele:

  - `dbus.x86_64`
  - `dbus-1.12.28`
  - `dbus-1.12.28-1.amzn2023.0.1`
  - `dbus-1:1.12.28-1.amzn2023.0.1.x86_64`
- Wir unterstützen auch Paketnamenkomponenten mit einem einzigen Platzhalter in den oben genannten Formaten, z. B. in den folgenden:
  - `dbus*`
  - `dbus-1.12.2*`
  - `dbus-*:1.12.28-1.amzn2023.0.1.x86_64`

Debian Server und Ubuntu Server

Paket-Manager: APT

Genehmigte Patches und abgelehnte Patches: Legen Sie für genehmigte sowie abgelehnte Patches Folgendes fest:

- Paketnamen im Format `ExamplePkg33`

> **Note**
>
> Verwenden Sie für Debian Server- und Ubuntu Server-Listen keine Elemente wie Architektur oder Versionen. Beispiel: Sie legen den Paketnamen `ExamplePkg33` fest, um alles Folgende in einer Patch-Liste einzubeziehen:
>
> - `ExamplePkg33.x86.1`
> - `ExamplePkg33.x86.2`
> - `ExamplePkg33.x64.1`
> - `ExamplePkg33.3.2.5-364.noarch`

SUSE Linux Enterprise Server (SLES)

Paket-Manager: Zypper

Genehmigte Patches und abgelehnte Patches: Sie können für genehmigte sowie abgelehnte Patch-Listen Folgendes festlegen:

- Vollständige Paketnamen in Formaten wie z. B.:
  - `SUSE-SLE-Example-Package-15-2023-123`
  - `example-pkg-2023.15.4-46.17.1.x86_64.rpm`
- Paketnamen mit einem einzigen Platzhalter wie z. B.:
  - `SUSE-SLE-Example-Package-15-2023-*`
  - `example-pkg-2023.15.4-46.17.1.*.rpm`

Paketnamen-Formate für macOS

Unterstützte Paketmanager: Softwareupdate, Installationsprogramm, Brew, Brew Cask

Genehmigte Patches und abgelehnte Patches: Geben Sie für genehmigte sowie abgelehnte Patch-Listen vollständige Paketnamen in folgenden formaten an:

- `XProtectPlistConfigData`
- `MRTConfigData`

Platzhalter werden in Listen genehmigter und abgelehnter Patches für macOS nicht unterstützt.

Paketnamen-Formate für Windows-Betriebssysteme

Geben Sie für Windows-Betriebssysteme Patches mithilfe von Microsoft Knowledge Base IDs und Microsoft Security Bulletin an IDs. Beispiel:

```
KB2032276,KB2124261,MS10-048
```

Patch-Gruppen

> **Note**
>
> Patch-Gruppen werden nicht in Patch-Vorgängen verwendet, die auf Patch-Richtlinien basieren. Weitere Informationen zur Arbeit mit Patch-Richtlinien finden Sie unter Patch-Richtlinienkonfigurationen in Quick Setup.
> Die Patchgruppenfunktion wird in der Konsole für Konto-Regionen-Paare nicht unterstützt, die vor der Veröffentlichung der Patch-Richtlinienunterstützung am 22. Dezember 2022 noch keine Patchgruppen verwendet haben. Die Patchgruppenfunktion ist weiterhin für Konto-Regionen-Paare verfügbar, die vor diesem Datum mit der Verwendung von Patchgruppen begonnen haben.

Mithilfe einer Patch-Gruppe können Sie verwaltete Knoten einer bestimmten Patch-Baseline in Patch Manager, ein Tool in AWS Systems Manager, zuordnen. Mit Patch-Gruppen können Sie sicherstellen, dass Sie geeignete Patches basierend auf den zugeordneten Patch-Baseline-Regeln für die richtigen Sätze von Knoten bereitstellen. Patch-Gruppen können außerdem dazu beitragen, die Bereitstellung von Patches zu vermeiden, bevor diese angemessen getestet sind. So können Sie Patch-Gruppen beispielsweise für unterschiedliche Umgebungen (Entwicklung, Test und Produktion) erstellen und jede Patch-Gruppe für eine geeignete Patch-Baseline registrieren.

Wenn Sie `AWS-RunPatchBaseline` oder andere SSM-Befehlsdokumente zum Patching ausführen, können Sie verwaltete Knoten über deren ID oder Tags anvisieren. Basierend auf dem Patch-Gruppenwert, den Sie dem verwalteten Knoten hinzugefügt haben, werten dann SSM Agent und Patch Manager aus, welche Patch-Baseline verwendet werden soll.

Verwendung von Tags zur Definition von Patchgruppen

Sie können eine Patchgruppe erstellen, indem Sie Tags verwenden, die auf Ihre Instances der Amazon Elastic Compute Cloud (Amazon EC2) und Nicht-EC2-Knoten in Ihrer Hybrid- und Multi-

Cloud-Umgebung angewendet werden. Beachten Sie die folgenden Details zum Verwenden von Tags für Patchgruppen:

- Eine Patchgruppe muss entweder mithilfe des Tag-Schlüssels `Patch Group` oder `PatchGroup` definiert werden, die auf Ihre verwalteten Knoten angewendet werden. Bei der Registrierung einer Patchgruppe für eine Patch-Baseline werden alle identischen Werte, die für diese beiden Schlüssel angegeben wurden, als Teil derselben Gruppe interpretiert. Nehmen wir zum Beispiel an, Sie haben fünf Knoten mit dem ersten der folgenden Schlüssel-Wert-Paare und fünf mit dem zweiten getaggt:
  - `key=PatchGroup,value=DEV`
  - `key=Patch Group,value=DEV`

  Der Patch Manager-Befehl zum Erstellen einer Baseline fasst diese 10 verwalteten Knoten auf der Grundlage des Werts `DEV` zu einer einzigen Gruppe zusammen. Das AWS CLI-Äquivalent für den Befehl zum Erstellen einer Patch-Baseline für Patch-Gruppen lautet wie folgt:

  ```
  aws ssm register-patch-baseline-for-patch-group \
      --baseline-id pb-0c10e65780EXAMPLE \
      --patch-group DEV
  ```

  Die Kombination von Werten aus verschiedenen Schlüsseln zu einem einzigen Ziel ist einzigartig für diesen Patch Manager-Befehl zum Erstellen einer neuen Patchgruppe und wird von anderen API-Aktionen nicht unterstützt. Wenn Sie beispielsweise send-command-Aktionen mit `PatchGroup`- und `Patch Group`-Schlüsseln und mit denselben Werten ausführen, zielen Sie auf zwei völlig unterschiedliche Gruppen von Knoten ab:

  ```
  aws ssm send-command \
      --document-name AWS-RunPatchBaseline \
      --targets "Key=tag:PatchGroup,Values=DEV"
  ```

  ```
  aws ssm send-command \
      --document-name AWS-RunPatchBaseline \
      --targets "Key=tag:Patch Group,Values=DEV"
  ```

- Es gibt Beschränkungen für das Tag-basierte Targeting. Jedes Ziele-Array für `SendCommand` kann maximal fünf Schlüssel-Wert-Paare haben.

- Es wird empfohlen, nur eine dieser Konventionen für Tag-Schlüssel zu wählen, entweder `PatchGroup` (ohne Leerzeichen) oder `Patch Group` (mit Leerzeichen). Wenn Sie Tags in EC2-Instance-Metadaten zugelassen haben, müssen Sie jedoch `PatchGroup` verwenden.
- Bei dem Schlüssel wird die Groß-/Kleinschreibung berücksichtigt. Sie können einen beliebigen Wert angeben, um die Ressourcen in dieser Gruppe zu identifizieren und darauf auszurichten, z. B. „Webserver“ oder „US-EAST-PROD“, aber der Schlüssel muss `Patch Group` oder `PatchGroup` sein.

Wenn Sie eine Patch-Gruppe erstellt und verwaltete Knoten mit Tags markiert haben, können Sie die Patch-Gruppe für eine Patch-Baseline anmelden. Indem Sie die Patch-Gruppe für eine Patch-Baseline registrieren, stellen Sie sicher, dass die Knoten innerhalb der Patch-Gruppe die in der zugehörigen Patch-Baseline definierten Regeln befolgen.

Weitere Informationen zum Erstellen von Patch-Gruppen und Zuordnen von Patch-Gruppen zu einer Patch-Baseline finden Sie unter Erstellen und Verwalten von Patch-Gruppen und Einer Patch-Baseline eine Patch-Gruppe hinzufügen.

Ein Beispiel für das Erstellen einer Patch-Baseline und von Patch-Gruppen über die AWS Command Line Interface (AWS CLI) finden Sie unter Tutorial: Patchen Sie eine Serverumgebung mit dem AWS CLI. Weitere Informationen über Amazon-EC2-Tags finden Sie unter Ihre Amazon-EC2-Ressourcen markieren im Amazon-EC2-Benutzerhandbuch.

Funktionsweise

Wenn das System eine Patch-Baseline auf einen verwalteten Knoten anwendet, überprüft SSM Agent, ob für den Knoten ein Patch-Gruppenwert definiert wurde. Wenn der Knoten einer Patch-Gruppe zugewiesen wurde, ermittelt Patch Manager anschließend, welche Patch-Baseline für diese Gruppe registriert wurde. Wenn für die Gruppe eine Patch-Baseline gefunden wird, weist Patch Manager SSM Agent an, die zugehörige Patch-Baseline zu verwenden. Wenn ein Knoten nicht für eine Patch-Gruppe konfiguriert wurde, weist Patch Manager SSM Agent automatisch an, die aktuell konfigurierte Standard-Patch-Baseline zu verwenden.

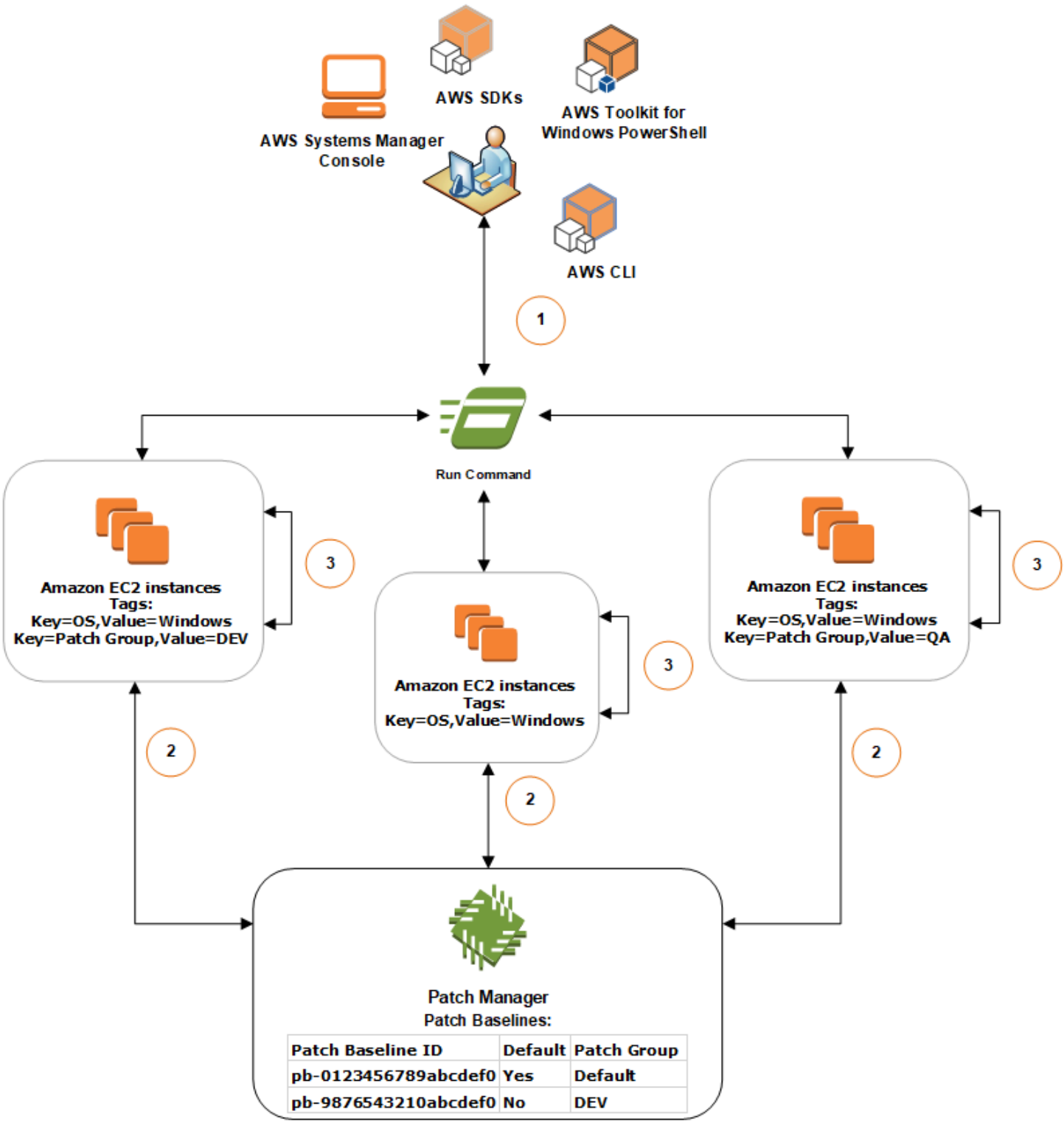
> **Important**
>
> Ein verwalteter Knoten kann sich nur in einer Patch-Gruppe befinden.
> Eine Patch-Gruppe kann nur für eine Patch-Baseline für jeden Betriebssystemtyp registriert werden.

Sie können das `Patch Group`-Tag (mit einem Leerzeichen) nicht auf eine Amazon-EC2-Instance anwenden, wenn die Option Allow tags in instance metadata (Tags in Instance-Metadaten zulassen) auf der Instance aktiviert ist. Durch das Zulassen von Tags in Instance-Metadaten wird verhindert, dass Tag-Schlüsselnamen Leerzeichen enthalten. Wenn Sie Tags in EC2-Instance-Metadaten zugelassen haben, müssen Sie den Tag-Schlüssel `PatchGroup` (ohne Leerzeichen) verwenden.

Abbildung 1: Allgemeines Beispiel für den Prozessablauf beim Patch-Vorgang

In der folgenden Abbildung sehen Sie ein allgemeines Beispiel der Prozesse, die Systems Manager beim Senden einer Run Command-Aufgabe an Ihre Serverflotte sendet, um mit Patch Manager Patches einzuspielen. Diese Prozesse bestimmen, welche Patch-Baselines für Patch-Vorgänge verwendet werden sollen. (Ein ähnlicher Prozess wird verwendet, wenn ein Wartungsfenster konfiguriert wurde, um einen Befehl zum Patchen mithilfe von Patch Manager zu senden.)

Der vollständige Vorgang wird unter der Abbildung erklärt.



In diesem Beispiel haben wir drei Gruppen von EC2-Instances für Windows Server mit den folgenden Tags:

EC2-Instances-Gruppe	Tags
Gruppe 1	`key=OS,value=Windows` `key=PatchGroup,value=DEV`
Gruppe 2	`key=OS,value=Windows`
Gruppe 3	`key=OS,value=Windows` `key=PatchGroup,value=QA`

In diesem Beispiel haben wir außerdem diese beiden Windows Server-Patch-Baselines:

Patch-Baseline-ID	Standard	Zugehörige Patch-Gruppe
`pb-0123456789abcdef0`	Ja	`Default`
`pb-9876543210abcdef0`	Nein	`DEV`

Der allgemeine Ablauf zum Scannen bzw. Installieren von Patches mit Run Command, einem Tool in AWS Systems Manager, und Patch Manager sieht wie folgt aus:

1. Einen Patchbefehl senden: Verwenden Sie die Systems Manager-Konsole, das SDK, die AWS Command Line Interface (AWS CLI) oder AWS Tools for Windows PowerShell zum Senden einer Run Command-Aufgabe mithilfe des Dokuments `AWS-RunPatchBaseline`. Die Abbildung zeigt eine Run Command-Aufgabe zum Patchen verwalteter Instances mit dem Tag `key=OS,value=Windows` als Ziel.
2. Patch-Baseline bestimmen: SSM Agent überprüft die auf die EC2-Instance angewendeten Patch-Gruppen-Tags und sendet eine Anfrage an Patch Manager für die entsprechende Patch-Baseline.
   - Passender Patch-Gruppenwert einer Patch-Baseline zugeordnet:
     1. SSM Agent, das auf EC2-Instance in Gruppe 1 installiert ist, empfängt den in Schritt 1 gesendeten Befehl, mit dem Patchvorgang zu beginnen. SSM Agent überprüft, ob die EC2-Instances über den Patch-Gruppen-Tag-Wert `DEV` verfügen und sendet eine Anfrage an Patch Manager für die zugehörige Patch-Baseline.

2. Patch Manager überprüft, ob der Patch-Baseline `pb-9876543210abcdef0` die Patch-Gruppe `DEV` zugeordnet ist, und sendet eine Benachrichtigung an SSM Agent.
3. SSM Agent ruft basierend auf den in Patch Manager konfigurierten Genehmigungsregeln und Ausnahmen einen Snapshot der Patch-Baseline von `pb-9876543210abcdef0` ab und fährt mit dem nächsten Schritt fort.

- Instance verfügt nicht über ein Patch-Gruppen-Tag:

1. SSM Agent, das auf EC2-Instance in Gruppe 2 installiert ist, empfängt den in Schritt 1 gesendeten Befehl, mit dem Patchvorgang zu beginnen. SSM Agent überprüft, ob die EC2-Instances nicht über das Patch-Gruppen-Tag `Patch Group` oder `PatchGroup` verfügen. SSM Agent sendet daraufhin eine Anfrage an Patch Manager für die Standard-Windows-Patch-Baseline.
2. Patch Manager überprüft, ob die Standard-Patch-Baseline für Windows Server `pb-0123456789abcdef0` ist, und benachrichtigt SSM Agent.
3. SSM Agent ruft basierend auf den in der Standard-Patch-Baseline Patch Manager konfigurierten Genehmigungsregeln und Ausnahmen einen Snapshot der Patch-Baseline von `pb-0123456789abcdef0` ab und fährt mit dem nächsten Schritt fort.

- Es gibt keinen passenden, einer Patch-Baseline zugeordneten Patch-Gruppenwert:

1. SSM Agent, das auf EC2-Instance in Gruppe 3 installiert ist, empfängt den in Schritt 1 gesendeten Befehl, mit dem Patchvorgang zu beginnen. SSM Agent überprüft, ob die EC2-Instances über den Patch-Gruppen-Tag-Wert `QA` verfügen und sendet eine Anfrage an Patch Manager für die zugehörige Patch-Baseline.
2. Patch Manager findet keine Patch-Baseline mit der zugeordneten Patch-Gruppe `QA`.
3. Patch Manager benachrichtigt SSM Agent, die Standard-Patch-Baseline für Windows, `pb-0123456789abcdef0`, zu verwenden.
4. SSM Agent ruft basierend auf den in der Standard-Patch-Baseline Patch Manager konfigurierten Genehmigungsregeln und Ausnahmen einen Snapshot der Patch-Baseline von `pb-0123456789abcdef0` ab und fährt mit dem nächsten Schritt fort.

3. Auf Patches scannen oder Patches installieren: Nachdem die anzuwendende Patch-Baseline bestimmt wurde, beginnt SSM Agent basierend auf dem in Schritt 1 festgelegten Vorgangswert entweder damit, nach Patches zu scannen oder diese zu installieren. Nach welchen Patches gescannt bzw. welche Patches installiert werden, wird durch die Genehmigungsregeln und Patch-Ausnahmen bestimmt, die im von Patch Manager bereitgestellten Patch-Baseline-Snapshot definiert sind.

Weitere Informationen

- Statuswerte der Patch-Compliance

Informationen zum Patchen von Anwendungen, die von Microsoft unter Windows Server veröffentlicht wurden

Verwenden Sie die Informationen in diesem Thema, um die Vorbereitung auf Patchanwendungen auf Windows Server mit Patch Manager, einem Tool in AWS Systems Manager, zu erleichtern.

Patching von Microsoft-Anwendungen

Patching-Support für Anwendungen auf von Windows Server verwalteten Knoten ist auf Anwendungen beschränkt, die von Microsoft veröffentlicht werden.

> **Note**
>
> In einigen Fällen veröffentlicht Microsoft Patches für Anwendungen, die kein aktualisiertes Datum und keine aktualisierte Uhrzeit angeben. In diesen Fällen wird ein aktualisiertes Datum und eine Uhrzeit von `01/01/1970` standardmäßig angegeben.

Patch-Baselines, um von Microsoft veröffentlichte Anwendungen zu patchen

Für Windows Server werden drei vordefinierte Patch-Baselines bereitgestellt. Die Patch-Baselines `AWS-DefaultPatchBaseline` und `AWS-WindowsPredefinedPatchBaseline-OS` unterstützen nur Betriebssystemupdates auf dem Windows-Betriebssystem selbst. `AWS-DefaultPatchBaseline` wird als Standard-Patch-Baseline für von Windows Server verwalteten Knoten verwendet, es sei denn, Sie geben eine andere Patch-Baseline an. Die Konfigurationseinstellungen in diesen beiden Patch-Baselines sind identisch. Die neuere der beiden, `AWS-WindowsPredefinedPatchBaseline-OS`, wurde erstellt, um sie von der dritten vordefinierten Patch-Baseline für Windows Server zu unterscheiden. Diese Patch-Baseline, `AWS-WindowsPredefinedPatchBaseline-OS-Applications`, kann verwendet werden, um Patches sowohl auf das Windows Server-Betriebssystem als auch auf unterstützte Anwendungen, die von Microsoft veröffentlicht wurden, anzuwenden.

Sie können zum Aktualisieren von Anwendungen, die von Microsoft veröffentlicht wurden, auf Windows Server-Computern auch benutzerdefinierte Patch-Baselines erstellen.

Support für das Patchen von Anwendungen, die von Microsoft auf lokalen Servern, Edge-Geräten und anderen Nicht-Knoten veröffentlicht wurden VMs EC2

Um von Microsoft veröffentlichte Anwendungen auf virtuellen Maschinen (VMs) und anderen nicht EC2 verwalteten Knoten zu patchen, müssen Sie die Stufe Advanced-Instances aktivieren. Die Nutzung des Advanced-Instances-Kontingents ist kostenpflichtig. Für Patch-Anwendungen, die von Microsoft auf Amazon Elastic Compute Cloud (Amazon EC2) -Instances veröffentlicht wurden, fallen jedoch keine zusätzlichen Gebühren an. Weitere Informationen finden Sie unter Konfigurieren von Instance-Kontingenten.

Windows-Update-Option für „andere Microsoft-Produkte“

Damit Patch Manager von Microsoft auf Ihren von Windows Server verwalteten Knoten veröffentlichte Anwendungen patchen kann, muss die Windows-Update-Option Ich möchte Updates für andere Microsoft-Produkte erhalten, wenn ich Windows aktualisiere auf dem verwalteten Knoten aktiviert sein.

Informationen zum Zulassen dieser Option für einen einzelnen verwalteten Knoten finden Sie unter Aktualisieren von Office mit Microsoft Update auf der Microsoft-Support-Website.

Bei einer Flotte von verwalteten Knoten auf Windows Server 2016 und höher können Sie die Einstellung mithilfe eines Group Policy Object (GPO, Gruppenrichtlinienobjekt) aktivieren. Navigieren Sie im Gruppenrichtlinien-Verwaltungseditor zu Computer-Konfiguration, Administrative Vorlagen, Windows-Komponenten, Windows-Updates und wählen Sie Installieren von Updates für andere Microsoft-Produkte aus. Wir empfehlen außerdem, das GPO mit zusätzlichen Parametern zu konfigurieren, die ungeplante automatische Updates und Neustarts außerhalb von Patch Manager verhindern. Weitere Informationen finden Sie unter Konfigurieren automatischer Updates in einer Umgebung ohne Active Directory auf der Website für technische Dokumentation von Microsoft.

Bei einer Flotte von verwalteten Knoten, die auf Windows Server 2012 oder 2012 R2 ausgeführt werden, können Sie die Option mithilfe eines Skripts aktivieren, wie unter Aktivieren und Deaktivieren von Microsoft Update in Windows 7 über Skript auf der Microsoft-Docs-Blog-Website beschrieben. Sie können z. B. Folgendes tun:

1. Speichern Sie das Skript aus dem Blogbeitrag in einer Datei.
2. Laden Sie die Datei in einen Amazon Simple Storage Service (Amazon S3)-Bucket oder an einem anderen zugänglichen Speicherort hoch.

3. Verwenden SieRun Command, ein Tool in AWS Systems Manager, um das Skript mithilfe des Systems Manager Manager-Dokuments (SSM-Dokument) `AWS-RunPowerShellScript` mit einem Befehl ähnlich dem folgenden auf Ihren verwalteten Knoten auszuführen.

```
Invoke-WebRequest `
    -Uri "https://s3.aws-api-domain/amzn-s3-demo-bucket/script.vbs" `
    -Outfile "C:\script.vbs" cscript c:\script.vbs
```

Mindestparameteranforderungen

Um von Microsoft veröffentlichte Anwendungen in Ihre benutzerdefinierte Patch-Baseline aufzunehmen, müssen Sie mindestens das Produkt angeben, das Sie patchen möchten. Der folgende Befehl AWS Command Line Interface (AWS CLI) veranschaulicht die Mindestanforderungen für das Patchen eines Produkts, z. B. Microsoft Office 2016.

Linux & macOS

```
aws ssm create-patch-baseline \
    --name "My-Windows-App-Baseline" \
    --approval-rules
 "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=PRODUCT,Values='Office 2016'},
{Key=PATCH_SET,Values='APPLICATION'}]},ApproveAfterDays=5}]"
```

Windows Server

```
aws ssm create-patch-baseline ^
    --name "My-Windows-App-Baseline" ^
    --approval-rules
 "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=PRODUCT,Values='Office 2016'},
{Key=PATCH_SET,Values='APPLICATION'}]},ApproveAfterDays=5}]"
```

Wenn Sie die Produktfamilie der Microsoft-Anwendung angeben, müssen alle Produkte der ausgewählten Produktfamilie unterstützt werden. Um beispielsweise das Produkt „Active Directory Rights Management Services Client 2.0“ zu patchen, müssen Sie dessen Produktfamilie als „Active Directory“ und nicht beispielsweise als „Office“ oder „SQL Server“ angeben. Der folgende AWS CLI Befehl demonstriert eine übereinstimmende Kombination von Produktfamilie und Produkt.

Linux & macOS

```
aws ssm create-patch-baseline \
    --name "My-Windows-App-Baseline" \
    --approval-rules
 "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=PRODUCT_FAMILY,Values='Active
 Directory'},{Key=PRODUCT,Values='Active Directory Rights Management Services Client
 2.0'},{Key=PATCH_SET,Values='APPLICATION'}]},ApproveAfterDays=5}]"
```

Windows Server

```
aws ssm create-patch-baseline ^
    --name "My-Windows-App-Baseline" ^
    --approval-rules
 "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=PRODUCT_FAMILY,Values='Active
 Directory'},{Key=PRODUCT,Values='Active Directory Rights Management Services Client
 2.0'},{Key=PATCH_SET,Values='APPLICATION'}]},ApproveAfterDays=5}]"
```

> **Note**
>
> Wenn Sie eine Fehlermeldung über eine nicht übereinstimmende Produkt- und Familienkopplung erhalten, finden Sie unter Problem: Nicht übereinstimmende Produktpaare family/product Tipps zur Lösung des Problems.

Verwenden von Kernel Live Patching auf von Amazon Linux 2 verwalteten Knoten

Kernel Live Patching für Amazon Linux 2 ermöglicht es Ihnen, Patches für Sicherheitsschwachstellen und kritische Fehler auf einen laufenden Linux-Kernel anzuwenden, ohne Neustarts oder Unterbrechungen der laufenden Anwendungen. Sie profitieren damit von einer verbesserten Service- und Anwendungsverfügbarkeit, gleichzeitig bleibt Ihre Infrastruktur sicher und auf dem neuesten Stand. Kernel Live Patching wird auf Amazon-EC2-Instances und AWS IoT Greengrass-Core-Geräten unterstützt sowie auf virtuellen On-Premises-Maschinen, die auf Amazon Linux 2 ausgeführt werden.

Allgemeine Informationen zu Kernel Live Patching finden Sie unter Kernel Live Patching auf AL2 im Amazon-Linux–2-Benutzerhandbuch.

Nachdem Sie Kernel Live Patching auf einem von Amazon Linux 2 verwalteten Knoten aktiviert haben, können Sie Patch Manager, ein Tool in AWS Systems Manager, verwenden, um Kernel-Live-Patches auf den verwalteten Knoten anzuwenden. Die Verwendung des Patch Manager ist eine Alternative zur Verwendung vorhandener Yum-Workflows auf dem Knoten, um die Updates anzuwenden.

Bevor Sie beginnen

Um mithilfe des Patch Manager Kernel-Live-Patches auf Ihre von Amazon Linux 2 verwalteten Knoten anzuwenden, stellen Sie sicher, dass Ihre Knoten auf der richtigen Architektur und Kernel-Version basieren. Weitere Informationen finden Sie unter Unterstützte Konfigurationen und Voraussetzungen im Amazon-EC2-Benutzerhandbuch.

Themen

- Kernel Live Patching mit Patch Manager verwenden
- So funktioniert Patch Manager mit Kernel Live Patching
- Aktivieren von Kernel Live Patching mit Run Command
- Anwenden von Kernel-Live-Patches unter Verwendung von Run Command
- Deaktivieren von Kernel Live Patching mit Run Command

Kernel Live Patching mit Patch Manager verwenden

Aktualisieren der Kernel-Version

Sie müssen einen verwalteten Knoten nicht neu starten, nachdem Sie ein Kernel-Live-Patch-Update angewendet haben. AWS stellt jedoch Kernel-Live-Patches für eine Kernel-Version von Amazon Linux 2 für bis zu drei Monate nach der Veröffentlichung bereit. Nach Ablauf der dreimonatigen Frist müssen Sie auf eine spätere Kernel-Version aktualisieren, um weiterhin Kernel-Live-Patches zu erhalten. Wir empfehlen Ihnen, mithilfe eines Wartungsfensters mindestens einmal alle drei Monate einen Neustart Ihres Knoten zu planen, um das Update der Kernel-Version zu veranlassen.

Deinstallieren von Kernel-Live-Patches

Kernel-Live-Patches können nicht mit dem Patch Manager deinstalliert werden. Stattdessen können Sie Kernel Live Patching deaktivieren, wodurch die RPM-Pakete für die angewendeten Kernel-Live-Patches entfernt werden. Weitere Informationen finden Sie unter Deaktivieren von Kernel Live Patching mit Run Command.

Kernel-Compliance

In einigen Fällen kann der Kernel durch die Installation aller CVE-Fixes von Live-Patches für die aktuelle Kernel-Version die Compliance-Ebene erreichen, die auch eine neuere Kernel-Version hätte. Wenn dies geschieht, wird die neuere Version als `Installed` und der verwaltete Knoten als `Compliant` gemeldet. Für die neuere Kernel-Version wird jedoch keine Installationszeit gemeldet.

Ein Kernel-Live-Patch, mehrere CVEs

Wenn sich ein Kernel-Live-Patch auf mehrere CVEs bezieht und diese CVEs verschiedene Klassifizierungs- und Schweregradwerte aufweisen, wird für den Patch nur die höchste Klassifizierung und der höchste Schweregrad der CVEs gemeldet.

Im weiteren Teil dieses Abschnitts wird erläutert, wie Patch Manager zum Anwenden von Kernel-Live-Patches auf verwaltete Knoten verwendet wird, die diese Anforderungen erfüllen.

So funktioniert Patch Manager mit Kernel Live Patching

AWS veröffentlicht zwei Arten von Kernel-Live-Patches für Amazon Linux 2: Sicherheitsupdates und Fehlerbehebungen. Um diese Patch-Typen anzuwenden, verwenden Sie ein Patch-Baseline-Dokument, das nur auf die in der folgenden Tabelle aufgeführten Klassifizierungen und Schweregrade ausgerichtet ist.

Klassifizierung	Schweregrad
`Security`	`Critical`, `Important`
`Bugfix`	`All`

Sie können eine benutzerdefinierte Patch-Baseline erstellen, die nur auf diese Patches ausgerichtet ist, oder die vordefinierte Patch-Baseline `AWS-AmazonLinux2DefaultPatchBaseline` verwenden. Mit anderen Worten, Sie können `AWS-AmazonLinux2DefaultPatchBaseline` mit von Amazon Linux 2 verwalteten Knoten verwenden, auf denen Kernel Live Patching aktiviert ist, und es werden Kernel-Live-Updates angewendet.

> **Note**
>
> Die `AWS-AmazonLinux2DefaultPatchBaseline`-Konfiguration hat eine Wartezeit von sieben Tagen nach Veröffentlichung oder letzten Aktualisierung eines Patches, bevor er automatisch installiert wird. Wenn Sie nicht sieben Tage warten möchten, bis Kernel-Live-Patches automatisch genehmigt werden, können Sie eine benutzerdefinierte Patch-Baseline erstellen und verwenden. In der Patch-Baseline können Sie keine Wartezeit für automatische Genehmigung oder einen kürzeren oder längeren Zeitraum angeben. Weitere Informationen finden Sie unter Arbeiten mit benutzerdefinierten Patch-Baselines.

Wir empfehlen die folgende Strategie zum Patchen Ihrer verwalteten Knoten mit Kernel-Live-Updates:

1. Aktivieren Sie Kernel Live Patching auf Ihren von Amazon Linux 2 verwalteten Knoten.
2. Verwenden Sie Run Command, ein Tool in AWS Systems Manager, um einen `Scan`-Vorgang auf Ihren verwalteten Knoten auszuführen, und verwenden Sie dabei die vordefinierte `AWS-AmazonLinux2DefaultPatchBaseline` oder eine benutzerdefinierte Patch-Baseline, die auch nur `Security`-Updates anvisiert, deren Schweregrade als `Critical` oder `Important` klassifiziert sind, und die den `Bugfix`-Schweregrad `All` haben.
3. Verwenden Sie Compliance, ein Tool in AWS Systems Manager, um zu überprüfen, ob für einen der gescannten verwalteten Knoten die Nichtkonformität zum Patchen gemeldet wird. Wenn dies der Fall ist, zeigen Sie die Compliance-Details für den Knoten an, um festzustellen, ob Kernel-Live-Patches im verwalteten Knoten fehlen.
4. Um fehlende Kernel-Live-Patches zu installieren, verwenden Sie Run Command mit derselben Patch-Baseline, die Sie zuvor angegeben haben, führen Sie dieses Mal jedoch eine `Install`-Operation anstelle einer `Scan`-Operation aus.

   Da Kernel-Live-Patches installiert werden, ohne dass ein Neustart erforderlich ist, können Sie die Neustartoption `NoReboot` für diese Operation auswählen.

   > **Note**
   >
   > Sie können den verwalteten Knoten dennoch neu starten, wenn dies für andere auf dem Knoten installierte Patch-Typen erforderlich ist oder wenn Sie auf einen neueren Kernel aktualisieren möchten. Wählen Sie in diesen Fällen stattdessen die Neustartoption `RebootIfNeeded` aus.

5. Kehren Sie zu Compliance zurück, um zu überprüfen, ob die Kernel-Live-Patches installiert wurden.

Aktivieren von Kernel Live Patching mit Run Command

Um Kernel Live Patching zu aktivieren, können Sie entweder `yum`-Befehle auf Ihren verwalteten Knoten ausführen oder Run Command und ein benutzerdefiniertes Systems-Manager-Dokument (SSM-Dokument) verwenden, das Sie erstellen.

Weitere Informationen zum Aktivieren von Kernel Live Patching durch Ausführen von `yum`-Befehlen direkt auf dem verwalteten Knoten finden Sie unter Aktivieren von Kernel Live Patching im Benutzerhandbuch zu Amazon EC2.

> **Note**
>
> Wenn Sie Kernel-Live-Patching aktivieren und der bereits auf dem verwalteten Knoten ausgeführte Kernel eine frühere Version als `kernel-4.14.165-131.185.amzn2.x86_64` (die unterstützte Mindestversion) ist, installiert der Prozess die neueste verfügbare Kernel-Version und startet den verwalteten Knoten neu. Wenn der Knoten bereits `kernel-4.14.165-131.185.amzn2.x86_64` oder höher ausführt, installiert der Prozess keine neuere Version und startet den Knoten nicht neu.

So aktivieren Sie Kernel Live Patching mit Run Command (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Befehl ausführen aus.
4. Wählen Sie in der Liste Command document (Befehlsdokument) das benutzerdefinierte SSM-Dokument `AWS-ConfigureKernelLivePatching` aus.
5. Geben Sie im Abschnitt Command parameters (Befehlsparameter) an, ob verwaltete Knoten als Teil dieser Operation neu gestartet werden sollen.
6. Weitere Informationen zur Verwendung der übrigen Steuerelemente auf dieser Seite finden Sie unter Ausführen von Befehlen über die Konsole.
7. Wählen Sie Ausführen aus.

Aktivieren von Kernel Live Patching (AWS CLI)

- Führen Sie den folgenden Befehl auf Ihrem lokalen Computer aus.

  Linux & macOS

  ```
  aws ssm send-command \
      --document-name "AWS-ConfigureKernelLivePatching" \
      --parameters "EnableOrDisable=Enable" \
      --targets "Key=instanceids,Values=instance-id"
  ```

  Windows Server

  ```
  aws ssm send-command ^
      --document-name "AWS-ConfigureKernelLivePatching" ^
      --parameters "EnableOrDisable=Enable" ^
      --targets "Key=instanceids,Values=instance-id"
  ```

  Ersetzen Sie *instance-id* durch die ID des von Amazon Linux 2 verwalteten Knoten, auf dem Sie das Feature aktivieren möchten, beispielsweise i-02573cafcfEXAMPLE. Um das Feature auf mehreren verwalteten Knoten zu aktivieren, können Sie eines der folgenden Formate verwenden.

  - `--targets "Key=instanceids,Values=instance-id1,instance-id2"`
  - `--targets "Key=tag:tag-key,Values=tag-value"`

  Informationen über andere Optionen, die Sie in dem Befehl verwenden können, finden Sie im Abschnitt send-command in der AWS CLI-Befehlsreferenz.

Anwenden von Kernel-Live-Patches unter Verwendung von Run Command

Um Kernel-Live-Patches anzuwenden, können Sie entweder `yum`-Befehle auf Ihren verwalteten Knoten ausführen oder Run Command und das SSM-Dokument `AWS-RunPatchBaseline` verwenden.

Weitere Informationen zum Anwenden von Kernel-Live-Patches durch Ausführung von `yum`-Befehlen direkt auf dem verwalteten Knoten finden Sie unter Anwenden von Kernel-Live-Patches im Benutzerhandbuch zu Amazon EC2.

So wenden Sie Kernel-Live-Patches unter Verwendung von Run Command an (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Befehl ausführen aus.
4. Wählen Sie in der Liste Command document (Befehlsdokument) das SSM-Dokument `AWS-RunPatchBaseline` aus.
5. Führen Sie im Abschnitt Command parameters (Befehlsparameter) einen der folgenden Schritte aus:

   - Wenn Sie prüfen, ob neue Kernel-Live-Patches verfügbar sind, wählen Sie für Operation die Option `Scan` aus. Wenn Ihre verwalteten Knoten nach dieser Operation nicht neu gestartet werden sollen, wählen Sie für Reboot Option (Neustartoption) `NoReboot` aus. Nach Abschluss der Operation können Sie in Compliance prüfen, ob neue Patches vorhanden sind und wie der Compliance-Status lautet.
   - Wenn Sie die Patch-Compliance bereits überprüft haben und bereit sind, verfügbare Kernel-Live-Patches anzuwenden, wählen Sie für Operation die Option `Install` aus. Wenn Ihre verwalteten Knoten nach dieser Operation nicht neu gestartet werden sollen, wählen Sie für Reboot Option (Neustartoption) `NoReboot` aus.
6. Weitere Informationen zur Verwendung der übrigen Steuerelemente auf dieser Seite finden Sie unter Ausführen von Befehlen über die Konsole.
7. Wählen Sie Ausführen aus.

So wenden Sie Kernel-Live-Patches unter Verwendung von Run Command an (AWS CLI)

1. Führen Sie den folgenden Befehl von Ihrem lokalen Computer aus, um eine `Scan`-Operation auszuführen, bevor Sie Ihre Ergebnisse in Compliance überprüfen.

   Linux & macOS

   ```
   aws ssm send-command \
       --document-name "AWS-RunPatchBaseline" \
       --targets "Key=InstanceIds,Values=instance-id" \
       --parameters '{"Operation":["Scan"],"RebootOption":["RebootIfNeeded"]}'
   ```

Windows Server

```
aws ssm send-command ^
    --document-name "AWS-RunPatchBaseline" ^
    --targets "Key=InstanceIds,Values=instance-id" ^
    --parameters {\"Operation\":[\"Scan\"],\"RebootOption\":[\"RebootIfNeeded
\"]}
```

Informationen über andere Optionen, die Sie in dem Befehl verwenden können, finden Sie im Abschnitt send-command in der AWS CLI-Befehlsreferenz.

2. Führen Sie den folgenden Befehl von Ihrem lokalen Computer aus, um eine `Install`-Operation auszuführen, nachdem Sie die Ergebnisse in Compliance überprüft haben.

   Linux & macOS

   ```
   aws ssm send-command \
       --document-name "AWS-RunPatchBaseline" \
       --targets "Key=InstanceIds,Values=instance-id" \
       --parameters '{"Operation":["Install"],"RebootOption":["NoReboot"]}'
   ```

   Windows Server

   ```
   aws ssm send-command ^
       --document-name "AWS-RunPatchBaseline" ^
       --targets "Key=InstanceIds,Values=instance-id" ^
       --parameters {\"Operation\":[\"Install\"],\"RebootOption\":[\"NoReboot\"]}
   ```

Ersetzen Sie in den beiden vorangegangenen Befehlen *instance-id* durch die ID des von Amazon Linux 2 verwalteten Knoten, auf dem Sie Kernel-Live-Patches anwenden möchten, beispielsweise i-02573cafcfEXAMPLE. Um das Feature auf mehreren verwalteten Knoten zu aktivieren, können Sie eines der folgenden Formate verwenden.

- `--targets "Key=instanceids,Values=`*instance-id1*`,`*instance-id2*`"`
- `--targets "Key=tag:`*tag-key*`,Values=`*tag-value*`"`

Informationen über andere Optionen, die Sie in diesen Befehlen verwenden können, finden Sie im Abschnitt send-command in der AWS CLI-Befehlsreferenz.

Deaktivieren von Kernel Live Patching mit Run Command

Um Kernel Live Patching zu deaktivieren, können Sie entweder `yum`-Befehle auf Ihren verwalteten Knoten ausführen oder Run Command und das benutzerdefinierte SSM-Dokument `AWS-ConfigureKernelLivePatching`.

> **Note**
>
> Wenn Sie das Kernel-Live-Patching nicht mehr verwenden möchten, können Sie es jederzeit deaktivieren. In den meisten Fällen ist das Deaktivieren des Features nicht erforderlich.

Weitere Informationen zum Deaktivieren von Kernel Live Patching durch Ausführen von `yum`-Befehlen direkt auf dem verwalteten Knoten finden Sie unter Aktivieren von Kernel Live Patching im Benutzerhandbuch zu Amazon EC2.

> **Note**
>
> Wenn Sie Kernel Live Patching deaktivieren, deinstalliert der Prozess das Kernel Live Patching-Plugin und startet dann den verwalteten Knoten neu.

Deaktivieren von Kernel Live Patching mit Run Command (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Befehl ausführen aus.
4. Wählen Sie in der Liste Command document (Befehlsdokument) das SSM-Dokument `AWS-ConfigureKernelLivePatching` aus.
5. Geben Sie im Abschnitt Befehlsparameter Werte für erforderliche Parameter an.
6. Weitere Informationen zur Verwendung der übrigen Steuerelemente auf dieser Seite finden Sie unter Ausführen von Befehlen über die Konsole.
7. Wählen Sie Ausführen aus.

Deaktivieren von Kernel Live Patching (AWS CLI)

- Verwenden Sie einen Befehl ähnlich dem folgenden:

  Linux & macOS

  ```
  aws ssm send-command \
      --document-name "AWS-ConfigureKernelLivePatching" \
      --targets "Key=instanceIds,Values=instance-id" \
      --parameters "EnableOrDisable=Disable"
  ```

  Windows Server

  ```
  aws ssm send-command ^
      --document-name "AWS-ConfigureKernelLivePatching" ^
      --targets "Key=instanceIds,Values=instance-id" ^
      --parameters "EnableOrDisable=Disable"
  ```

  Ersetzen Sie *instance-id* durch die ID des von Amazon Linux 2 verwalteten Knoten, auf dem Sie das Feature deaktivieren möchten, beispielsweise i-02573cafcfEXAMPLE. Um das Feature auf mehreren verwalteten Knoten zu deaktivieren, können Sie eines der folgenden Formate verwenden.

  - `--targets "Key=instanceids,Values=instance-id1,instance-id2"`
  - `--targets "Key=tag:tag-key,Values=tag-value"`

  Informationen über andere Optionen, die Sie in dem Befehl verwenden können, finden Sie im Abschnitt send-command in der AWS CLI-Befehlsreferenz.

Arbeiten mit Patch Manager-Ressourcen und Compliance mithilfe der Konsole

Führen Sie Folgendes aus, bevor Sie Patch Manager, ein Tool in AWS Systems Manager, verwenden. Diese Aufgaben werden später in diesem Abschnitt ausführlich erläutert.

1. Stellen Sie sicher, dass die von AWS vordefinierte Patch-Baseline für die einzelnen verwendeten Betriebssystemtypen Ihre Anforderungen erfüllt. Ist dies nicht der Fall, sollten Sie eine Patch-Baseline erstellen, in der Standard-Patches für diesen Typ von verwalteten Knoten definiert sind, und diese als Standard-Patch-Baseline festlegen.

2. Organisieren Sie verwaltete Knoten mithilfe von Amazon Elastic Compute Cloud (Amazon EC2)-Tags in Patch-Gruppen (optional, aber empfohlen).
3. Führen Sie eine der folgenden Aktionen aus:
   - (Empfohlen) Konfigurieren Sie eine Patch-Richtlinie in Quick Setup, einem Tool in Systems Manager, mit dem Sie fehlende Patches nach einem Zeitplan für eine gesamte Organisation, eine Teilmenge von Organisationseinheiten oder ein einzelnes AWS-Konto installieren können. Weitere Informationen finden Sie unter Das Patchen für Instances in einer Organisation mithilfe einer Patch-Richtlinie für Quick Setup konfigurieren.
   - Erstellen Sie ein Wartungsfenster, das das Systems Manager-Dokument (SSM-Dokument) `AWS-RunPatchBaseline` in einem Run Command-Aufgabentyp verwendet. Weitere Informationen finden Sie unter Tutorial: Erstellen Sie ein Wartungsfenster zum Patchen über die Konsole.
   - Führen Sie `AWS-RunPatchBaseline` manuell in einer Run Command-Operation aus. Weitere Informationen finden Sie unter Ausführen von Befehlen über die Konsole.
   - Patchen Sie Knoten bei Bedarf manuell mit der Funktion Patch now (Jetzt patchen). Weitere Informationen finden Sie unter On-Demand-Patchen von verwalteten Knoten.
4. Überwachen Sie das Patching, um die Compliance zu überprüfen und Fehler zu untersuchen.

Themen

- Erstellen einer Patch-Richtlinie
- Patch-Dashboard-Zusammenfassungen anzeigen
- Arbeiten mit Patch-Compliance-Berichten
- On-Demand-Patchen von verwalteten Knoten
- Arbeiten mit Patch-Baselines
- Anzeigen verfügbarer Patches
- Erstellen und Verwalten von Patch-Gruppen
- Integration von Patch Manager in AWS Security Hub CSPM

Erstellen einer Patch-Richtlinie

Eine Patch-Richtlinie ist eine Konfiguration, die Sie mit Quick Setup, einem Tool in AWS Systems Manager, einrichten. Patch-Richtlinien bieten eine umfassendere und zentralisiertere Kontrolle über Ihre Patching-Vorgänge, als dies mit anderen Methoden zum Konfigurieren von Patches möglich ist.

Eine Patch-Richtlinie definiert den Zeitplan und die Baseline, die beim automatischen Patchen Ihrer Knoten und Anwendungen verwendet werden sollen.

Weitere Informationen finden Sie unter den folgenden Themen:

- Patch-Richtlinienkonfigurationen in Quick Setup
- Das Patchen für Instances in einer Organisation mithilfe einer Patch-Richtlinie für Quick Setup konfigurieren

Patch-Dashboard-Zusammenfassungen anzeigen

Die Dashboard-Registerkarte in Patch Manager bietet eine Übersicht über die Konsole, die Sie verwenden können, um Ihre Patch-Vorgänge in einer konsolidierten Ansicht zu überwachen. Patch Manager ist ein Tool in AWS Systems Manager. Auf der Registerkarte Dashboard können Sie folgenden Tabellen anzeigen:

- Ein Snapshot, wie viele verwaltete Knoten mit Patching-Regeln konform bzw. nicht konform sind.
- Ein Snapshot des Alters der Patch-Compliance-Ergebnisse für Ihre verwalteten Knoten.
- Eine verknüpfte Anzahl davon, wie viele nicht konforme verwaltete Knoten für jeden der häufigsten Gründe für Nicht-Compliance vorhanden sind.
- Eine verknüpfte Liste der letzten Patching-Vorgänge.
- Eine verknüpfte Liste der wiederkehrenden Patching-Vorgänge, die eingerichtet wurden.

So zeigen Sie Patch-Dashboard-Übersichten

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Dashboard aus.
4. Scrollen Sie zu dem Abschnitt mit zusammenfassenden Daten, den Sie anzeigen möchten:
    - Amazon-EC2-Instance-Management
    - Zusammenfassung der Compliance
    - Anzahl der Nichteinhaltungen der Compliance
    - Compliance-Berichte
    - Nicht auf Patch-Richtlinien basierende Vorgänge

- Wiederkehrende Aufgaben, die nicht auf Patch-Richtlinien basieren

Arbeiten mit Patch-Compliance-Berichten

Die Informationen in den folgenden Themen helfen Ihnen beim Erstellen von und Arbeiten mit Patch-Compliance-Berichten in Patch Manager, einem Tool in AWS Systems Manager.

Die Informationen in den folgenden Themen gelten unabhängig davon, welche Methode oder Art der Konfiguration Sie für Ihre Patching-Vorgänge verwenden:

- Eine in Quick Setup konfigurierte Patch-Richtlinie
- Eine in Quick Setup konfigurierte Host-Management-Option
- Ein Wartungsfenster zum Ausführen eines Patch-`Scan` oder einer `Install`-Aufgabe
- Ein On-Demand-Jetzt patchen-Vorgang

> **Important**
>
> Patch-Compliance-Berichte sind point-in-time Schnappschüsse, die nur bei erfolgreichen Patch-Vorgängen generiert werden. Jeder Bericht enthält eine Erfassungszeit, aus der hervorgeht, wann der Compliance-Status berechnet wurde.
> Wenn Sie mehrere Arten von Vorgängen einsetzen, um Ihre Instances auf Patch-Compliance zu überprüfen, beachten Sie, dass jeder Scan die Patch-Compliance-Daten der vorherigen Scans überschreibt. Dies kann zu unerwarteten Ergebnissen in Ihren Patch-Compliance-Daten führen. Weitere Informationen finden Sie unter Identifizieren der Ausführung, die Patch-Compliance-Daten erstellt hat.
> Um zu überprüfen, welche Patch-Baseline verwendet wurde, um die neuesten Compliance-Informationen zu generieren, navigieren Sie zur Registerkarte Compliance-Berichte in Patch Manager, suchen Sie die Zeile für den verwalteten Knoten, zu dem Sie Informationen wünschen, und wählen Sie dann die Baseline-ID in der Spalte Verwendete Baseline-ID aus.

Themen

- Anzeigen der Patch-Compliance-Ergebnisse
- Generieren von Patch-Compliance-Berichten im .csv-Format
- Behebung nicht konformer verwalteter Knoten mit Patch Manager
- Identifizieren der Ausführung, die Patch-Compliance-Daten erstellt hat

Anzeigen der Patch-Compliance-Ergebnisse

Gehen Sie wie folgt vor, um Patch-Compliance-Informationen zu Ihren verwalteten Knoten anzuzeigen.

Dieses Verfahren gilt für Patchvorgänge, die das `AWS-RunPatchBaseline`-Dokument verwenden. Weitere Informationen zum Anzeigen von Patch-Compliance-Informationen für Patch-Operationen, die das `AWS-RunPatchBaselineAssociation`-Dokument verwenden, finden Sie unter Identifizieren von nicht konformen verwalteten Knoten.

> **Note**
>
> Die Patchscanvorgänge für Quick Setup und die Explorer Verwendung des `AWS-RunPatchBaselineAssociation` Dokuments. Quick Setupund Explorer sind beide Tools in AWS Systems Manager.

Identifizieren der Patch-Lösung für ein bestimmtes CVE-Problem (Linux)

Für viele Linux-basierte Betriebssysteme geben die Patch-Compliance-Ergebnisse an, welche CVE (Common Vulnerabilities and Exposure) Bulletin-Probleme durch welche Patches behoben werden. Mithilfe dieser Informationen können Sie feststellen, wie dringend Sie einen fehlenden oder fehlgeschlagenen Patch installieren müssen.

CVE-Details sind für unterstützte Versionen der folgenden Betriebssystemtypen enthalten:

- AlmaLinux
- Amazon Linux 2
- Amazon Linux 2023
- Oracle Linux
- Red Hat Enterprise Linux (RHEL)
- Rocky Linux

> **Note**
>
> Standardmäßig stellt CentOS Stream keine CVE-Informationen zu Updates bereit. Sie können diese Unterstützung jedoch zulassen, indem Sie Repositorys von Drittanbietern wie

das EPEL-Repository (EPEL), das von Fedora veröffentlicht wurde, verwenden. Weitere Informationen finden Sie unter EPEL im Fedora-Wiki.
Derzeit werden CVE-ID-Werte nur für Patches mit dem Status `Missing` oder `Failed` gemeldet.

Sie können CVE auch IDs zu Ihren Listen mit genehmigten oder abgelehnten Patches in Ihren Patch-Baselines hinzufügen, je nach Situation und Ihren Patch-Zielen.

Weitere Informationen zum Arbeiten mit genehmigten und abgelehnten Patch-Listen finden Sie in den folgenden Themen:

- Arbeiten mit benutzerdefinierten Patch-Baselines
- Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen
- Funktionsweise von Patch-Baseline-Regeln auf Linux-basierten Systemen
- Wie Patches installiert werden

> **Note**
>
> In einigen Fällen veröffentlicht Microsoft Patches für Anwendungen, die kein aktualisiertes Datum und keine aktualisierte Uhrzeit angeben. In diesen Fällen wird ein aktualisiertes Datum und eine Uhrzeit von `01/01/1970` standardmäßig angegeben.

Anzeigen der Patch-Compliance-Ergebnisse

Verwenden Sie die folgenden Verfahren, um Patch-Compliance-Ergebnisse in der AWS Systems Manager -Konsole anzuzeigen.

> **Note**
>
> Weitere Informationen zum Erstellen von Patch-Compliance-Berichten, die in einen Amazon Simple Storage Service (Amazon S3)-Bucket heruntergeladen werden, finden Sie unter Generieren von Patch-Compliance-Berichten im .csv-Format.

Anzeigen der Patch-Compliance-Ergebnisse

1. Führen Sie eine der folgenden Aufgaben aus.

   Option 1 (empfohlen) – Navigieren Sie von Patch Manager, einem Tool in AWS Systems Manager:

   - Wählen Sie im Navigationsbereich Patch Manager aus.
   - Wählen Sie die Registerkarte Compliance reporting (Compliance-Berichte).
   - Wählen Sie im Bereich Details zum Knoten-Patching die Knoten-ID des verwalteten Knotens aus, für den Sie die Ergebnisse der Patch-Compliance überprüfen möchten.
   - Wählen Sie im Bereich Details in der Eigenschaftenliste die Option Patches aus.

   Option 2 – Navigieren Sie ab Compliance, einem Tool in AWS Systems Manager:

   - Wählen Sie im linken Navigationsbereich Compliance.
   - Für Zusammenfassung von Compliance-Ressourcen wählen Sie in der Spalte für die Typen von Patch-Ressourcen, die Sie überprüfen möchten, z. B. Nicht regelkonforme Ressourcen, eine Zahl aus.
   - In der Ressourcen-Liste unten wählen Sie die ID des verwalteten Knoten aus, für den Sie die Patch-Compliance-Ergebnisse prüfen möchten.
   - Wählen Sie im Bereich Details in der Eigenschaftenliste die Option Patches aus.

   Option 3 – Navigieren Sie von Fleet Manager, einem Tool in AWS Systems Manager.

   - Wählen Sie im Navigationsbereich Fleet Manager aus.
   - Wählen Sie im Bereich Verwaltete instances die ID des verwalteten Knotens aus, für den Sie die Ergebnisse der Patch-Compliance überprüfen möchten.
   - Wählen Sie im Bereich Details in der Eigenschaftenliste die Option Patches aus.

2. (Optional) Wählen Sie im Suchfeld
   ( )
   einen der verfügbaren Filter aus.

   Wählen Sie zum Beispiel für Red Hat Enterprise Linux (RHEL) aus den folgenden Optionen aus:

   - Name

- Klassifizierung
- Status
- Schweregrad

Wählen Sie für Windows Server eine der folgenden Optionen aus:

- KB
- Klassifizierung
- Status
- Schweregrad

3. Wählen Sie einen der verfügbaren Werte für den ausgewählten Filtertyp aus. Wenn Sie beispielsweise „Status“ ausgewählt haben, wählen Sie jetzt einen Konformitätsstatus wie „Fehlgeschlagen“ oder „InstalledPendingRebootFehlend“.

   > **Note**
   >
   > Derzeit werden CVE-ID-Werte nur für Patches mit dem Status `Missing` oder `Failed` gemeldet.

4. Abhängig vom Compliance-Zustand des verwalteten Knoten können Sie auswählen, welche Maßnahmen zum Beheben von nicht konformen Knoten ergriffen werden sollen.

   Sie können beispielsweise wählen, dass Ihre nicht konformen verwalteten Knoten sofort gepatcht werden sollen. Informationen zum On-Demand-Patching Ihrer verwalteten Knoten finden Sie unter On-Demand-Patchen von verwalteten Knoten.

   Weitere Informationen zu Patch-Compliance-Status finden Sie unter Statuswerte der Patch-Compliance.

Generieren von Patch-Compliance-Berichten im .csv-Format

Sie können die AWS Systems Manager Konsole verwenden, um Patch-Compliance-Berichte zu erstellen, die als CSV-Datei in einem Amazon Simple Storage Service (Amazon S3) -Bucket Ihrer Wahl gespeichert werden. Sie können einen einzelnen On-Demand-Bericht erstellen oder einen Zeitplan für die automatische Generierung der Berichte angeben.

Berichte können für einen einzelnen verwalteten Knoten oder für alle verwalteten Knoten in Ihrem ausgewählten AWS-Konto und AWS-Region generiert werden. Für einen einzelnen Knoten enthält ein Bericht umfassende Details, einschließlich der Patches, die IDs sich auf die Nichtkonformität eines Knotens beziehen. Für einen Bericht über alle verwaltete Knoten werden nur zusammenfassende Informationen und die Anzahl der Patches von nicht konformen Knoten bereitgestellt.

Nachdem ein Bericht generiert wurde, können Sie ein Tool wie Amazon Quick Suite verwenden, um die Daten zu importieren und zu analysieren. Quick Suite ist ein Business Intelligence (BI) -Service, mit dem Sie Informationen in einer interaktiven visuellen Umgebung untersuchen und interpretieren können. Weitere Informationen finden Sie im Amazon Quick Suite-Benutzerhandbuch.

> **Note**
>
> Wenn Sie eine benutzerdefinierte Patch-Baseline erstellen, können Sie für Patches, die von dieser Patch-Baseline genehmigt wurden, einen Schweregrad für die Konformität angeben, beispielsweise `Critical` oder `High`. Wenn der Patch-Status eines genehmigten Patches als `Missing` gemeldet wird, dann ist der insgesamt gemeldete Konformitätsschweregrad der Patch-Baseline der von Ihnen angegebene Schweregrad.

Sie können auch ein Thema zum Amazon Simple Notification Service (Amazon SNS) angeben, um Benachrichtigungen zu senden, wenn ein Bericht erstellt wird.

Servicerollen zum Generieren von Patch-Compliance-Berichten

Wenn Sie zum ersten Mal einen Bericht erstellen, erstellt Systems Manager eine angenommene Automatisierungsrolle mit dem Namen `AWS-SystemsManager-PatchSummaryExportRole`, die für den Exportprozess zu S3 verwendet wird.

> **Note**
>
> Wenn Sie Compliance-Daten in einen verschlüsselten S3-Bucket exportieren, müssen Sie die zugehörige AWS KMS Schlüsselrichtlinie aktualisieren, um die erforderlichen Berechtigungen für bereitzustellen`AWS-SystemsManager-PatchSummaryExportRole`. Fügen Sie beispielsweise der AWS KMS Richtlinie Ihres S3-Buckets eine ähnliche Berechtigung hinzu:
>
> ```
> {
>     "Effect": "Allow",
> ```

```
    "Action": [
        "kms:GenerateDataKey"
    ],
    "Resource": "role-arn"
}
```

*role-arn*Ersetzen Sie es durch den Amazon-Ressourcennamen (ARN) der in Ihrem Konto erstellten Datei im Format`arn:aws:iam::111222333444:role/service-role/AWS-SystemsManager-PatchSummaryExportRole`.
Weitere Informationen finden Sie unter Schlüsselrichtlinien in AWS KMS im Entwicklerhandbuch für AWS Key Management Service .

Wenn Sie zum ersten Mal einen Bericht nach einem Zeitplan generieren, erstellt Systems Manager eine weitere Servicerolle mit dem Namen `AWS-EventBridge-Start-SSMAutomationRole` zusammen mit der Servicerolle `AWS-SystemsManager-PatchSummaryExportRole` (falls nicht bereits erstellt), die für den Exportvorgang verwendet werden soll. `AWS-EventBridge-Start-SSMAutomationRole`ermöglicht Amazon EventBridge , eine Automatisierung mit dem Runbook AWS ExportPatchReportToS3 zu starten.

Es wird empfohlen, diese Richtlinien und Rollen zu ändern. Dies kann dazu führen, dass die Erstellung von Patch-Compliance-Berichten fehlschlägt. Weitere Informationen finden Sie unter Problembehandlung bei der Erstellung von Patch-Compliance-Berichten.

Themen

- Was ist in einem generierten Patch-Compliance-Bericht enthalten?
- Generieren von Patch-Compliance-Berichten für einen einzelnen verwalteten Knoten
- Generieren von Patch-Compliance-Berichten für alle verwaltete Knoten
- Berichtsverlauf für Patch-Compliance anzeigen
- Zeitpläne für Patch-Compliance-Berichte anzeigen
- Problembehandlung bei der Erstellung von Patch-Compliance-Berichten

Was ist in einem generierten Patch-Compliance-Bericht enthalten?

Dieses Thema enthält Informationen zu den Inhaltstypen, die in den Patch-Compliance-Berichten enthalten sind, die generiert und in einen angegebenen S3-Bucket heruntergeladen werden.

Berichtsformat für einen einzelnen verwalteten Knoten

Ein für einen einzelnen verwalteten Knoten generierter Bericht liefert sowohl zusammenfassende als auch detaillierte Informationen.

Herunterladen eines Beispielberichts (einzelner Knoten)

Zu den zusammenfassenden Informationen für einen einzelnen verwalteten Knoten gehört Folgendes:

- Index
- Instance-ID
- Instance name
- Instance IP
- Plattformname
- Version der Plattform
- SSM Agent-Version
- Patch-Baseline
- Patch-Gruppe
- Compliance status (Compliance-Status)
- Compliance-Schweregrad
- Anzahl nicht konformer Patches mit kritischem Schweregrad
- Anzahl nicht konformer Patches mit hohem Schweregrad
- Anzahl nicht konformer Patches mit der Schweregrad Mittel
- Anzahl nicht konformer Patches mit niedrigem Schweregrad
- Anzahl nicht konformer Patches mit informativen Schweregrad
- Anzahl nicht konformer Patches mit nicht spezifiziertem Schweregrad

Zu den detaillierten Informationen für einen verwalteten einzelnen Knoten gehört Folgendes:

- Index
- Instance-ID
- Instance name

- Patch-Name
- ID/Patch KB-ID
- Patch-Status
- Zeitpunkt des letzten Berichts
- Compliance-Stufe
- Patch-Schweregrad
- Patch-Klassifizierung
- CVE-ID
- Patch-Baseline
- Logs-URL
- Instance IP
- Plattformname
- Version der Plattform
- SSM Agent-Version

> **Note**
>
> Wenn Sie eine benutzerdefinierte Patch-Baseline erstellen, können Sie für Patches, die von dieser Patch-Baseline genehmigt wurden, einen Schweregrad für die Konformität angeben, beispielsweise `Critical` oder `High`. Wenn der Patch-Status eines genehmigten Patches als `Missing` gemeldet wird, dann ist der insgesamt gemeldete Konformitätsschweregrad der Patch-Baseline der von Ihnen angegebene Schweregrad.

Berichtsformat für alle verwaltete Knoten

Ein für alle verwaltete Knoten generierter Bericht enthält nur zusammenfassende Informationen.

Herunterladen eines Beispielberichts (alle verwaltete Knoten)

Zu den zusammenfassenden Informationen für alle verwaltete Knoten gehört Folgendes:

- Index
- Instance-ID

- Instance name
- Instance IP
- Plattformname
- Version der Plattform
- SSM Agent-Version
- Patch-Baseline
- Patch-Gruppe
- Compliance status (Compliance-Status)
- Compliance-Schweregrad
- Anzahl nicht konformer Patches mit kritischem Schweregrad
- Anzahl nicht konformer Patches mit hohem Schweregrad
- Anzahl nicht konformer Patches mit der Schweregrad Mittel
- Anzahl nicht konformer Patches mit niedrigem Schweregrad
- Anzahl nicht konformer Patches mit informativen Schweregrad
- Anzahl nicht konformer Patches mit nicht spezifiziertem Schweregrad

Generieren von Patch-Compliance-Berichten für einen einzelnen verwalteten Knoten

Gehen Sie wie folgt vor, um einen Patch-Zusammenfassungs-Bericht für einen einzelnen verwalteten Knoten in Ihrem AWS-Konto zu generieren. Der Bericht für einen einzelnen verwalteten Knoten enthält Details zu jedem Patch, der nicht konform ist, einschließlich Patch-Namen und IDs.

So generieren Sie Patch-Compliance-Berichte für einen einzelnen verwalteten Knoten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Compliance reporting (Compliance-Berichte) aus.
4. Wählen Sie die Schaltfläche für die Zeile des verwalteten Knoten aus, für den Sie einen Bericht erstellen möchten, und wählen Sie dann View detail (Detail anzeigen) aus.
5. Wählen Sie Abschnitt mit der Patch-Zusammenfassung Exportieren nach S3 aus.
6. Für Berichtname geben Sie einen Namen ein, damit Sie den Bericht später leichter identifizieren können.

7. Für Meldehäufigkeit wählen Sie eine der folgenden Optionen aus:

   - On Demand – Erstellen Sie einen einmaligen Bericht. Fahren Sie mit Schritt 9 fort.
   - Nach einem Plan – Geben Sie einen wiederkehrenden Zeitplan für die automatische Erstellung von Berichten an. Fahren Sie fort mit Schritt 8.
8. Für den Typ „Nach einem Plan“ geben Sie entweder einen Kursausdruck an, z. B. alle 3 Tage, oder geben Sie einen Cron-Ausdruck an, um die Berichtshäufigkeit festzulegen.

   Informationen zu Cron-Ausdrücken finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.
9. Für Bucket-Name wählen Sie den Namen eines S3-Buckets aus, in dem die CSV-Berichtsdateien gespeichert werden sollen.

   > **Important**
   >
   > Wenn Sie in einem System arbeiten AWS-Region , das nach dem 20. März 2019 gestartet wurde, müssen Sie einen S3-Bucket in derselben Region auswählen. Nach diesem Datum gestartete Regionen wurden standardmäßig deaktiviert. Weitere Informationen und eine Liste dieser Regionen finden Sie unter Aktivieren einer Region in der Allgemeine Amazon Web Services-Referenz.
10. (Optional) Um Benachrichtigungen zu senden, wenn der Bericht erstellt wird, erweitern Sie den Abschnitt SNS-Thema und wählen Sie dann aus SNS-Thema Amazon-Ressourcenname (ARN) ein vorhandenes Amazon-SNS-Thema aus.
11. Wählen Sie Absenden aus.

Informationen zum Anzeigen eines Verlaufs von generierten Berichten finden Sie unter Berichtsverlauf für Patch-Compliance anzeigen.

Informationen zum Anzeigen von Details zu von Ihnen erstellten Berichtszeitplänen finden Sie unter Zeitpläne für Patch-Compliance-Berichte anzeigen.

Generieren von Patch-Compliance-Berichten für alle verwaltete Knoten

Gehen Sie wie folgt vor, um einen Patch-Zusammenfassungs-Bericht für alle verwaltete Knoten in Ihrem AWS-Konto zu generieren. Der Bericht für alle verwalteten Knoten zeigt an, welche Knoten nicht konform sind und wie viele Patches nicht konform sind. Es gibt keine Namen oder andere Bezeichner der Patches. Für diese zusätzlichen Details können Sie einen Patch-Compliance-Bericht

für einen einzelnen verwalteten Knoten erstellen. Informationen finden Sie unter Generieren von Patch-Compliance-Berichten für einen einzelnen verwalteten Knoten weiter vorne in diesem Thema.

Generieren von Patch-Compliance-Berichten für alle verwaltete Knoten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Compliance reporting (Compliance-Berichte) aus.
4. Klicken Sie auf Export to S3 (Exportieren nach S3). (Wählen Sie nicht zuerst eine Knoten-ID aus.)
5. Für Berichtname geben Sie einen Namen ein, damit Sie den Bericht später leichter identifizieren können.
6. Für Meldehäufigkeit wählen Sie eine der folgenden Optionen aus:
   - On Demand – Erstellen Sie einen einmaligen Bericht. Fahren Sie mit Schritt 8 fort.
   - Nach einem Plan – Geben Sie einen wiederkehrenden Zeitplan für die automatische Erstellung von Berichten an. Fahren Sie fort mit Schritt 7.
7. Für den Typ „Nach einem Plan“ geben Sie entweder einen Kursausdruck an, z. B. alle 3 Tage, oder geben Sie einen Cron-Ausdruck an, um die Berichtshäufigkeit festzulegen.

   Informationen zu Cron-Ausdrücken finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.
8. Für Bucket-Name wählen Sie den Namen eines S3-Buckets aus, in dem die CSV-Berichtsdateien gespeichert werden sollen.

   > **Important**
   >
   > Wenn Sie in einem System arbeiten AWS-Region , das nach dem 20. März 2019 gestartet wurde, müssen Sie einen S3-Bucket in derselben Region auswählen. Nach diesem Datum gestartete Regionen wurden standardmäßig deaktiviert. Weitere Informationen und eine Liste dieser Regionen finden Sie unter Aktivieren einer Region in der Allgemeine Amazon Web Services-Referenz.
9. (Optional) Um Benachrichtigungen zu senden, wenn der Bericht erstellt wird, erweitern Sie den Abschnitt SNS-Thema und wählen Sie dann aus SNS-Thema Amazon-Ressourcenname (ARN) ein vorhandenes Amazon-SNS-Thema aus.

10. Wählen Sie Absenden aus.

Informationen zum Anzeigen eines Verlaufs von generierten Berichten finden Sie unter Berichtsverlauf für Patch-Compliance anzeigen.

Informationen zum Anzeigen von Details zu von Ihnen erstellten Berichtszeitplänen finden Sie unter Zeitpläne für Patch-Compliance-Berichte anzeigen.

Berichtsverlauf für Patch-Compliance anzeigen

Mithilfe der Informationen in diesem Thema können Sie sich Details zu den Patch-Compliance-Berichten anzeigen lassen, die in Ihrem erstellt wurden AWS-Konto.

Anzeigen des Berichtsverlaufs für Patch-Compliance

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Compliance reporting (Compliance-Berichte) aus.
4. Klicken Sie auf Alle S3-Exporte anzeigen und danach auf die Registerkarte Exportieren des Verlaufs.

Zeitpläne für Patch-Compliance-Berichte anzeigen

Mithilfe der Informationen in diesem Thema können Sie sich Details zu den in Ihrem erstellten Zeitplan für die Erstellung von Berichten zur Patch-Konformität anzeigen lassen AWS-Konto.

Anzeigen des Berichtsverlaufs für Patch-Compliance

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Compliance reporting (Compliance-Berichte) aus.
4. Wählen Sie View all S3 exports (Alle S3-Exporte anzeigen) und danach die Registerkarte Report schedule rules (Regeln für die Berichtsplanung) aus.

Problembehandlung bei der Erstellung von Patch-Compliance-Berichten

Im Folgenden finden Sie Informationen zur Behandlung von Problemen mit Patch-Compliance-Berichten in Patch Manager, einem Tool in AWS Systems Manager.

Themen

- Eine Nachricht meldet, dass die AWS-SystemsManager-PatchManagerExportRolePolicy-Richtlinie beschädigt ist
- Nach dem Löschen von Patch-Compliance-Richtlinien oder -Rollen werden geplante Berichte nicht erfolgreich generiert

Eine Nachricht meldet, dass die **`AWS-SystemsManager-PatchManagerExportRolePolicy`**-Richtlinie beschädigt ist

Problem: Sie erhalten eine Fehlermeldung ähnlich der folgenden, die angibt, dass `AWS-SystemsManager-PatchManagerExportRolePolicy` beschädigt ist:

```
An error occurred while updating the AWS-SystemsManager-PatchManagerExportRolePolicy
policy. If you have edited the policy, you might need to delete the policy, and any
role that uses it, then try again. Systems Manager recreates the roles and policies
you have deleted.
```

- Lösung: Verwenden Sie die Patch Manager Konsole oder löschen AWS CLI Sie die betroffenen Rollen und Richtlinien, bevor Sie einen neuen Patch-Compliance-Bericht erstellen.

  So löschen Sie die beschädigte Richtlinie über die Konsole

  1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
  2. Führen Sie eine der folgenden Aktionen aus:

     On-Demand-Berichte – Wenn das Problem beim Generieren eines einmaligen On-Demand-Berichts aufgetreten ist, wählen Sie in der linken Navigation Richtlinien aus und suchen Sie nach `AWS-SystemsManager-PatchManagerExportRolePolicy`. Löschen Sie dann die Richtlinie. Wählen Sie anschließend Rollen aus, suchen Sie nach `AWS-SystemsManager-PatchSummaryExportRole` und löschen Sie sie.

     Geplante Berichte – Wenn der Fehler während der Erstellung eines geplanten Berichts aufgetreten ist, wählen Sie in der linken Navigation Richtlinien aus, suchen Sie nacheinander nach `AWS-EventBridge-Start-SSMAutomationRolePolicy` und `AWS-`

SystemsManager-PatchManagerExportRolePolicy und löschen Sie jede Richtlinie. Wählen Sie anschließend Rollen aus, schen Sie nacheinander nach AWS-EventBridge-Start-SSMAutomationRole und AWS-SystemsManager-PatchSummaryExportRole und löschen Sie jede Rolle.

Um die beschädigte Richtlinie mit dem zu löschen AWS CLI

Ersetzen Sie das *placeholder values* durch Ihre Konto-ID.

- Wenn das Problem bei der Erstellung eines einmaligen On-Demand-Berichts aufgetreten ist, führen Sie die folgenden Befehle aus:

```
aws iam delete-policy --policy-arn arn:aws:iam::account-id:policy/AWS-
SystemsManager-PatchManagerExportRolePolicy
```

```
aws iam delete-role --role-name AWS-SystemsManager-PatchSummaryExportRole
```

Wenn das Problem bei der Erstellung eines Zeitplanberichts auftritt, führen Sie die folgenden Befehle aus:

```
aws iam delete-policy --policy-arn arn:aws:iam::account-id:policy/AWS-
EventBridge-Start-SSMAutomationRolePolicy
```

```
aws iam delete-policy --policy-arn arn:aws:iam::account-id:policy/AWS-
SystemsManager-PatchManagerExportRolePolicy
```

```
aws iam delete-role --role-name AWS-EventBridge-Start-SSMAutomationRole
```

```
aws iam delete-role --role-name AWS-SystemsManager-PatchSummaryExportRole
```

Nachdem Sie eines der beiden Verfahren abgeschlossen haben, folgen Sie den Schritten, um einen neuen Patch-Compliance-Bericht zu erstellen oder zu planen.

Nach dem Löschen von Patch-Compliance-Richtlinien oder -Rollen werden geplante Berichte nicht erfolgreich generiert

Problem: Wenn Sie zum ersten Mal einen Bericht erstellen, erstellt Systems Manager eine Servicerolle und eine Richtlinie für den Exportprozess (`AWS-SystemsManager-PatchSummaryExportRole` und `AWS-SystemsManager-PatchManagerExportRolePolicy`). Wenn Sie zum ersten Mal einen geplanten Bericht erstellen, erstellt Systems Manager eine weitere Servicerolle und eine Richtlinie (`AWS-EventBridge-Start-SSMAutomationRole` und `AWS-EventBridge-Start-SSMAutomationRolePolicy`). Diese ermöglichen es Amazon, eine Automatisierung mithilfe des Runbooks AWS ExportPatchReportToS3 zu EventBridge starten.

Wenn Sie eine dieser Richtlinien oder Rollen löschen, gehen die Verbindungen zwischen Ihrem Zeitplan und Ihrem angegebenen S3-Bucket und Amazon SNS-Thema möglicherweise verloren.

- Lösung: Um dieses Problem zu umgehen, empfehlen wir, den vorherigen Zeitplan zu löschen und einen neuen Zeitplan zu erstellen, um den zu ersetzen, bei dem Probleme aufgetreten sind.

Behebung nicht konformer verwalteter Knoten mit Patch Manager

Die Themen in diesem Abschnitt enthalten eine Übersicht darüber, wie verwaltete Knoten, die die Patch-Compliance nicht erfüllen, identifiziert werden können und wie sie konform gemacht werden.

Themen

- Identifizieren von nicht konformen verwalteten Knoten
- Statuswerte der Patch-Compliance
- Patchen nicht konformer verwalteter Knoten

Identifizieren von nicht konformen verwalteten Knoten

Out-of-compliance verwaltete Knoten werden identifiziert, wenn eines von zwei AWS Systems Manager Dokumenten (SSM-Dokumente) ausgeführt wird. Diese SSM-Dokumente verweisen auf die entsprechende Patch-Baseline für jeden verwalteten Knoten in Patch Manager, einem Tool in AWS Systems Manager. Anschließend werten sie den Patch-Zustand des verwalteten Knoten aus und stellen Ihnen dann Compliance-Ergebnisse zur Verfügung.

Es gibt zwei SSM-Dokumente, die verwendet werden, um nicht konforme verwaltete Knoten zu identifizieren oder zu aktualisieren: `AWS-RunPatchBaseline` und `AWS-RunPatchBaselineAssociation`. Jedes wird von verschiedenen Prozessen verwendet, und ihre

Compliance-Ergebnisse sind über verschiedene Kanäle verfügbar. In der folgenden Tabelle werden die Unterschiede zwischen diesen Dokumenten aufgeführt.

> **Note**
>
> Daten zur Patch-Konformität von Patch Manager können an AWS Security Hub CSPM gesendet werden. Security Hub CSPM bietet Ihnen einen umfassenden Überblick über Ihre Sicherheitswarnungen mit hoher Priorität und den Compliance-Status. Er überwacht auch den Patching-Status Ihrer Flotte. Weitere Informationen finden Sie unter Integration von Patch Manager in AWS Security Hub CSPM.

	AWS-RunPatchBaseline	**AWS-RunPatchBaselineAssociation**
Prozesse, die das Dokument verwenden	On-Demand-Patchen – Sie können verwaltete Knoten bei Bedarf scannen oder patchen, indem Sie die Option Patch now (Jetzt patchen) verwenden. Weitere Informationen finden Sie unter On-Demand-Patchen von verwalteten Knoten. Quick Setup-Patch-Richtlinien von Systems Manager – Sie können eine Patching-Konfiguration in Quick Setup, einem Tool in AWS Systems Manager, erstellen, die fehlende Patches in separaten Zeitplänen für eine gesamte Organisation, eine Teilmenge von Organisationseinheiten oder ein einzelnes AWS-Konto scannen oder installie	Quick Setup-Host-Verwaltung für Systems Manager – Sie können eine Host-Management-Konfigurationsoption in Quick Setup aktivieren, um Ihre verwalteten Instances täglich auf Patch-Compliance zu scannen. Weitere Informationen finden Sie unter Amazon-EC2-Host-Verwaltung mit Quick Setup einrichten. Systems Manager Explorer—Wenn Sie es zulassen Explorer, scannt ein Tool Ihre verwalteten Instanzen regelmäßig auf Patch-Konformität und meldet die Ergebnisse im Explorer Dashboard. AWS Systems Manager

	AWS-RunPatchBaseline	AWS-RunPatchBaselineAssociation
	ren kann. Weitere Informationen finden Sie unter Das Patchen für Instances in einer Organisation mithilfe einer Patch-Richtlinie für Quick Setup konfigurieren. Einen Befehl ausführen – Sie können `AWS-RunPatchBaseline` manuell in einem Vorgang in Run Command, einem Tool in AWS Systems Manager, ausführen. Weitere Informationen finden Sie unter Ausführen von Befehlen über die Konsole. Wartungsfenster – Sie können ein Wartungsfenster erstellen, das das SSM-Dokument `AWS-RunPatchBaseline` in einem Run Command-Aufgabentyp verwendet. Weitere Informationen finden Sie unter Tutorial: Erstellen Sie ein Wartungsfenster zum Patchen über die Konsole.	

	AWS-RunPatchBaseline	**AWS-RunPatchBaselineAssociation**
Format der Patch-Scan-Ergebnisdaten	Nach der Ausführung von `AWS-RunPatchBaseline` sendet Patch Manager ein `AWS:PatchSummary` -Objekt an Inventory, ein Tool in AWS Systems Manager. Dieser Bericht wird nur nach erfolgreichen Patch-Vorgängen generiert und enthält eine Erfassungszeit, anhand derer angegeben wird, wann der Compliance-Status berechnet wurde.	Nach der Ausführung von `AWS-RunPatchBaselineAssociation` sendet Patch Manager ein `AWS:ComplianceItem` -Objekt an Systems Manager Inventory.

	AWS-RunPatchBaseline	AWS-RunPatchBaselineAssociation
So zeigen Sie aktuelle Compliance-Berichte in der Konsole an	Sie können Patch-Compliance-Informationen für Prozesse anzeigen, die `AWS-RunPatchBaseline` in Systems Manager Compliance und Arbeiten mit verwalteten Knoten verwenden. Weitere Informationen finden Sie unter Anzeigen der Patch-Compliance-Ergebnisse.	Wenn Sie Quick Setup verwenden, um Ihre verwalteten Instances auf Patch-Compliance zu überprüfen, finden Sie den Compliance-Bericht unter Systems Manager Fleet Manager. Wählen Sie in der Fleet Manager-Konsole die Knoten-ID Ihres verwalteten Knotens aus. Wählen Sie im Menü Allgemein die Option Konfigurationskonformität aus. Wenn Sie Explorer verwenden, um Ihre verwalteten Instances auf Patch-Compliance zu überprüfen, finden Sie den Compliance-Bericht sowohl unter Explorer als auch unter Systems Manager OpsCenter.

	AWS-RunPatchBaseline	AWS-RunPatchBaselineAssociation
AWS CLI Befehle zum Anzeigen der Patch-Konformitätsergebnisse	Für Prozesse, die dies verwendenAWS-RunPatchBaseline , können Sie die folgenden AWS CLI Befehle verwenden, um zusammenfassende Informationen zu Patches auf einem verwalteten Knoten anzuzeigen. • describe-instance-patch-states • describe-instance-patch-states-for-patch-group • describe-patch-group-state	Für Prozesse, die dies verwendenAWS-RunPatchBaselineAssociation , können Sie den folgenden AWS CLI Befehl verwenden, um zusammenfassende Informationen zu Patches auf einer Instanz anzuzeigen. • list-compliance-items
Patch-Operationen	Für Prozesse, die AWS-RunPatchBaseline verwenden, geben Sie an, ob der Vorgang nur eine Scan-Operation oder eine Scan and install-Operation ausführen soll. Wenn Ihr Ziel darin besteht, nicht konforme verwaltete Knoten zu identifizieren und nicht zu beheben, führen Sie nur eine Scan-Operation durch.	Quick Setup- und Explorer-Prozesse, die AWS-RunPatchBaselineAssociation verwenden, führen nur eine Scan-Operation aus.
Weitere Informationen	SSM-Befehlsdokument zum Patchen: AWS-RunPatchBaseline	SSM-Befehlsdokument zum Patchen: AWS-RunPatchBaselineAssociation

Informationen zu den verschiedenen Patch-Compliance-Status, die möglicherweise gemeldet werden, finden Sie unter Statuswerte der Patch-Compliance

Informationen zur Behebung verwalteter Knoten, die die Patch-Compliance nicht erfüllen, finden Sie unter Patchen nicht konformer verwalteter Knoten.

Statuswerte der Patch-Compliance

Zu den Informationen über Patches für einen verwalteten Knoten gehört ein Bericht über den Zustand oder den Status jedes einzelnen Patches.

> **Tip**
>
> Wenn Sie einem verwalteten Knoten einen bestimmten Patch-Compliance-Status zuweisen möchten, können Sie den Befehl put-compliance-items AWS Command Line Interface (AWS CLI) oder die PutComplianceItemsAPI-Operation verwenden. Das Zuweisen eines Compliance-Zustands wird in der Konsole nicht unterstützt.

Verwenden Sie die Informationen in den folgenden Tabellen, um zu ermitteln, warum ein verwalteter Knoten möglicherweise nicht die Patch-Compliance erfüllt.

Patch-Compliance-Werte für Debian Server und Ubuntu Server

Für Debian Server und Ubuntu Server werden die Regeln für die Paketklassifizierung in verschiedene Compliance-Zustände in der folgenden Tabelle beschrieben.

> **Note**
>
> Beachten Sie bei der Auswertung der Statuswerte `INSTALLED`, `INSTALLED_OTHER` und `MISSING` Folgendes: Wenn Sie beim Erstellen oder Aktualisieren einer Patch-Baseline das Kontrollkästchen Funktionsupdates einschließen nicht aktivieren, sind Patch-Kandidaten-Versionen auf Patches in den folgenden Repositorys beschränkt:
>
> - Ubuntu Server 16.04 LTS: `xenial-security`
> - Ubuntu Server 18.04 LTS: `bionic-security`
> - Ubuntu Server 20.04 LTS: `focal-security`
> - Ubuntu Server 22.04 LTS (`jammy-security`)
> - Ubuntu Server 24.04 LTS (`noble-security`)

- Ubuntu Server 25.04 (`plucky-security`)
- `debian-security` (Debian Server

Wenn Sie die Option Nicht sicherheitsrelevante Aktualisierungen einschließen auswählen, werden auch Patches aus anderen Repositorys berücksichtigt.

Patch-Status	Description	Compliance status (Compliance-Status)
`INSTALLED`	Der Patch wird in der Patch-Baseline aufgeführt und ist auf dem verwalteten Knoten installiert. Er könnte entweder manuell von einer Person oder automatisch von Patch Manager installiert worden sein, wenn das `AWS-RunPatchBaseline` -Dokument auf dem verwalteten Knoten ausgeführt wurde.	Konform
`INSTALLED_OTHER`	Der Patch ist nicht in der Baseline enthalten oder wird von der Baseline nicht genehmigt, ist aber auf dem verwalteten Knoten installiert. Der Patch wurde möglicherweise manuell installiert, das Paket könnte eine erforderliche Abhängigkeit von einem anderen genehmigten Patch sein, oder der Patch war möglicherweise Teil eines InstallOverrideList Vorgangs. Wenn Sie `Block` nicht als die	Konform

Patch-Status	Description	Compliance status (Compliance-Status)
	Zurückgewiesene Patches-Aktion angeben, schließt `INSTALLED_OTHER` auch installiert, aber abgelehnte Patches ein.	

Patch-Status	Description	Compliance status (Compliance-Status)
INSTALLED_PENDING_REBOOT	INSTALLED_PENDING_REBOOT kann eines von zwei Dingen bedeuten: • Der Install-Vorgang von Patch Manager hat den Patch zwar auf den verwalteten Knoten angewendet, aber der Knoten wurde seit dem Anwenden des Patches nicht mehr neu gestartet. Dies bedeutet in der Regel, dass für den Parameter RebootOption die Option NoReboot ausgewählt wurde, als das AWS-RunPatchBaseline -Dokument zuletzt auf dem verwalteten Knoten ausgeführt wurde. Weitere Informationen finden Sie unter Parameter name: RebootOption . • Seit dem letzten Neustart des verwalteten Knotens wurde ein Patch außerhalb von Patch Manager installiert. In keinem Fall bedeutet dies, dass ein Patch mit diesem Status einen Neustart erfordert	Nicht konform

Patch-Status	Description	Compliance status (Compliance-Status)
	, sondern nur, dass der Knoten seit der Installation des Patches nicht neu gestartet wurde.	
INSTALLED_REJECTED	Der Patch wird auf dem verwalteten Knoten installiert, jedoch in einer Liste der Rejected patches (Abgelehnte Patches) angegeben. Dies bedeutet normalerweise, dass der Patch installiert wurde, bevor er einer Liste der abgelehnten Patches hinzugefügt wurde.	Nicht konform
MISSING	Pakete, die über die Baseline gefiltert und noch nicht installiert sind.	Nicht konform
FAILED	Pakete, die während des Patch-Vorgangs nicht installiert werden konnten.	Nicht konform

Patch-Compliance-Werte für andere Betriebssysteme

Für alle Betriebssysteme außer Debian Server und Ubuntu Server werden die Regeln für die Paketklassifizierung in verschiedene Compliance-Zustände in der folgenden Tabelle beschrieben.

Patch-Status	Description	Compliancewert
INSTALLED	Der Patch wird in der Patch-Baseline aufgeführt und ist auf dem verwalteten Knoten installiert. Er könnte entweder	Konform

Patch-Status	Description	Compliancewert
	manuell von einer Person oder automatisch von Patch Manager installiert worden sein, als das `AWS-RunPatchBaseline` -Dokument auf dem Knoten ausgeführt wurde.	

Patch-Status	Description	Compliancewert
`INSTALLED_OTHER` [1]	Der Patch befindet sich nicht an der Baseline, ist aber auf dem verwalteten Knoten installiert. Hierfür gibt es zwei mögliche Gründe: 1. Der Patch wurde möglicher weise manuell installiert. 2. Nur Linux: Das Paket wurde möglicherweise als erforderliche Abhängigk eit von einem anderen, genehmigten Patch installie rt. Wenn Sie `Allow as dependency` als Abgelehnte Patches angeben, erhalten Patches, die als Abhängigkeiten installiert wurden, den Berichtsstatus `INSTALLED _OTHER` . Windows Server unterstüt zt das Konzept der Patch-Abhängigkeiten nicht. Informationen dazu, wie Patch Manager mit Patches in der Liste Abgelehnt e Patches auf Windows Server umgeht, finden Sie unter Optionen für die Liste Abgelehnte Patches in benutzerdefinierten Patch-Baselines.	Konform

Patch-Status	Description	Compliancewert
INSTALLED_REJECTED	Der Patch wird auf dem verwalteten Knoten installiert, jedoch in einer Liste der abgelehnten Patches angegeben. Dies bedeutet normalerweise, dass der Patch installiert wurde, bevor er einer Liste der abgelehnten Patches hinzugefügt wurde.	Nicht konform

Patch-Status	Description	Compliancewert
INSTALLED_PENDING_REBOOT	INSTALLED_PENDING_REBOOT kann eines von zwei Dingen bedeuten: • Der Install-Vorgang von Patch Manager hat den Patch zwar auf den verwalteten Knoten angewendet, aber der Knoten wurde seit dem Anwenden des Patches nicht mehr neu gestartet. Dies bedeutet in der Regel, dass für den Parameter RebootOption die Option NoReboot ausgewählt wurde, als das AWS-RunPatchBaseline -Dokument zuletzt auf dem verwalteten Knoten ausgeführt wurde. Weitere Informationen finden Sie unter Parameter name: RebootOption . • Seit dem letzten Neustart des verwalteten Knotens wurde ein Patch außerhalb von Patch Manager installiert. In keinem Fall bedeutet dies, dass ein Patch mit diesem Status einen Neustart erfordert, sondern nur, dass der	Nicht konform

Patch-Status	Description	Compliancewert
	Knoten seit der Installation des Patches nicht neu gestartet wurde.	
MISSING	Der Patch wurde in der Baseline genehmigt, aber nicht auf dem verwalteten Knoten installiert. Wenn Sie die `AWS-RunPatchBaseline` -Dokumentaufgabe zum Scannen (nicht Installieren) konfigurieren, meldet das System diesen Status bei Patches, die beim Scan gefunden wurden, aber noch nicht installiert sind.	Nicht konform
FAILED	Der Patch wurde an der Baseline genehmigt, konnte aber nicht installiert werden. Zum Beheben dieses Problems überprüfen Sie die Befehlsausgabe auf Informationen, die Ihnen helfen, das Problem zu verstehen.	Nicht konform

Patch-Status	Description	Compliancewert
NOT_APPLICABLE [1]	Dieser Compliance-Status wird nur in Windows Server-Betriebssystemen gemeldet. Der Patch wurde in der Baseline genehmigt, der Service oder dem Feature, die den Patch verwendet, wurde aber auf dem verwaltet en Knoten nicht installiert. Beispielsweise würde ein Patch für einen Webserver-Service wie Internet Informati on Services (IIS) `NOT_APPLI CABLE` anzeigen, wenn er in der Baseline genehmigt wurde, der Webservice jedoch nicht auf dem verwalteten Knoten installiert ist. Ein Patch kann auch als `NOT_APPLI CABLE` markiert sein, wenn er durch ein nachfolgendes Update ersetzt wurde. Dies bedeutet, dass das spätere Update installiert ist und das `NOT_APPLICABLE` -Update nicht mehr benötigt wird.	Nicht zutreffend

Patch-Status	Description	Compliancewert
`AVAILABLE_SECURITY_UPDATES`	Dieser Compliance-Status wird nur in Windows Server-Betriebssystemen gemeldet. Ein verfügbarer Sicherheitsupdate-Patch, der nicht von der Patch-Baseline genehmigt wurde, kann den Konformitätswert `Compliant` oder `Non-Compliant` haben, wie in einer benutzerdefinierten Patch-Baseline definiert. Wenn Sie eine Patch-Baseline erstellen oder aktualisieren, wählen Sie den Status, den Sie Sicherheitspatches zuweisen möchten, die zwar verfügbar, aber nicht genehmigt sind, weil sie die in der Patch-Baseline angegebenen Installationskriterien nicht erfüllen. Beispielsweise können Sicherheitspatches, die Sie möglicherweise installieren möchten, übersprungen werden, wenn Sie eine lange Wartezeit für die Installation nach der Veröffentlichung eines Patches angegeben haben. Wenn während der von Ihnen angegebenen Wartezeit ein Update für den Patch veröffentlicht wird, beginnt	Konform oder nicht konform, je nachdem, welche Option für verfügbare Sicherheitsupdates ausgewählt wurde. **Note** Wenn Sie die Konsole verwenden, um eine Patch-Baseline zu erstellen oder zu aktualisieren, geben Sie diese Option im Feld Compliance-Status für verfügbare Sicherheitsupdates an. Mit dem update-patch-baselineBefehl AWS CLI to run the create-patch-baselineor geben Sie diese Option im `available-security-updates-compliance-status` Parameter an.

Patch-Status	Description	Compliancewert
	die Wartezeit für die Installation des Patches von vorne. Wenn die Wartezeit zu lang ist, können mehrere Versionen des Patches veröffentlicht, aber nie installiert werden. Bei der Anzahl der Patch-Zusammenfassungen gilt: Wenn ein Patch als `Available SecurityUpdate` gemeldet wird, ist er immer in `AvailableSecurityUpdateCount` enthalten. Wenn die Baseline so konfiguriert ist, dass diese Patches als `NonCompliant` gemeldet werden, ist sie auch in `SecurityNonComplianceCount` enthalten. Wenn die Baseline so konfiguriert ist, dass diese Patches als `Compliant` gemeldet werden, sind sie nicht in `SecurityNonComplianceCount` enthalten. Diese Patches werden immer mit einem nicht spezifizierten Schweregrad gemeldet und sind niemals in `CriticalNonCompliantCount` enthalten.	

[1] Für Patches mit dem Status `INSTALLED_OTHER` und `NOT_APPLICABLE`, lässt Patch Manager einige Daten aus Abfrageergebnissen aus, die auf dem Befehl describe-instance-patches basieren, z. B. die Werte für `Classification` und `Severity`. Dies geschieht, um zu verhindern, dass das Datenlimit für einzelne Knoten in Inventory, einem Tool in AWS Systems Manager, überschritten wird. Um alle Patch-Details anzuzeigen, können Sie den Befehl describe-available-patches nutzen.

Patchen nicht konformer verwalteter Knoten

Viele der gleichen AWS Systems Manager-Tools und -Prozesse, mit denen Sie verwaltete Knoten auf Patch-Compliance überprüfen können, können auch verwendet werden, damit Knoten die derzeit für sie geltenden Patch-Regeln erfüllen. Damit verwaltete Knoten die Patch-Compliance erfüllen, muss Patch Manager, ein Tool in AWS Systems Manager, einen `Scan and install`-Vorgang ausführen. (Wenn es Ihr Ziel ist, nicht konforme verwaltete Knoten nur zu identifizieren und sie nicht zu beheben, führen Sie stattdessen eine `Scan`-Operation aus. Weitere Informationen finden Sie unter Identifizieren von nicht konformen verwalteten Knoten.)

Installieren von Patches mit Systems Manager

Sie können aus mehreren Tools wählen, um eine `Scan and install`-Operation auszuführen:

- (Empfohlen) Konfigurieren Sie eine Patch-Richtlinie in Quick Setup, einem Tool in Systems Manager, mit dem Sie fehlende Patches nach einem Zeitplan für eine gesamte Organisation, eine Teilmenge von Organisationseinheiten oder ein einzelnes AWS-Konto installieren können. Weitere Informationen finden Sie unter Das Patchen für Instances in einer Organisation mithilfe einer Patch-Richtlinie für Quick Setup konfigurieren.
- Erstellen Sie ein Wartungsfenster, das das Systems Manager-Dokument (SSM-Dokument) `AWS-RunPatchBaseline` in einem Run Command-Aufgabentyp verwendet. Weitere Informationen finden Sie unter Tutorial: Erstellen Sie ein Wartungsfenster zum Patchen über die Konsole.
- Führen Sie `AWS-RunPatchBaseline` manuell in einer Run Command-Operation aus. Weitere Informationen finden Sie unter Ausführen von Befehlen über die Konsole.

- Installieren Sie Patches bei Bedarf mithilfe der Option Patch now (Jetzt patchen). Weitere Informationen finden Sie unter On-Demand-Patchen von verwalteten Knoten.

Identifizieren der Ausführung, die Patch-Compliance-Daten erstellt hat

Die Patch-Compliance-Daten stellen eine point-in-time Momentaufnahme des letzten erfolgreichen Patch-Vorgangs dar. Jeder Konformitätsbericht enthält sowohl eine Ausführungs-ID als auch eine

Erfassungszeit, anhand derer Sie feststellen können, welcher Vorgang die Konformitätsdaten erstellt hat und wann sie generiert wurden.

Wenn Sie mehrere Arten von Vorgängen zum Scannen Ihrer Instances auf Patch-Compliance haben, überschreibt jeder Scan die Patch-Compliance-Daten vorheriger Scans. Dies kann zu unerwarteten Ergebnissen in Ihren Patch-Compliance-Daten führen.

Nehmen wir an, Sie erstellen eine Patch-Richtlinie, die jeden Tag um 02:00 Uhr Ortszeit auf Patch-Compliance scannt. Diese Patch-Richtlinie verwendet eine Patch-Baseline, die Patches als Ziel haben, deren Schweregrad mit `Critical`, `Important` und `Moderate` gekennzeichnet ist. Diese Patch-Baseline gibt auch einige speziell abgelehnte Patches an.

Nehmen Sie außerdem an, dass Sie bereits ein Wartungsfenster eingerichtet haben, um jeden Tag um 04:00 Uhr Ortszeit denselben Satz verwalteter Knoten zu scannen, die Sie nicht löschen oder deaktivieren. Die Aufgabe dieses Wartungsfensters verwendet eine andere Patch-Baseline, eine, die nur Patches mit dem Schweregrad `Critical` als Ziel hat und keine bestimmten Patches ausschließt.

Wenn dieser zweite Scan vom Wartungsfenster durchgeführt wird, werden die Patch-Compliance-Daten des ersten Scans gelöscht und durch die Patch-Compliance des zweiten Scans ersetzt.

Daher empfehlen wir dringend, nur eine automatisierte Methode zum Scannen und Installieren in Ihren Patching-Vorgängen zu verwenden. Wenn Sie Patch-Richtlinien einrichten, sollten Sie andere Methoden zum Scannen auf Patch-Compliance löschen oder deaktivieren. Weitere Informationen finden Sie unter den folgenden Themen:

- So entfernen Sie eine Patching-Aufgabe aus einem Wartungsfenster – Aktualisieren oder Abmelden von Wartungsfenster-Aufgaben mithilfe der Konsole
- So löschen Sie eine State Manager-Zuordnung – Löschen von Zuordnungen.

Um tägliche Patch-Compliance-Scans in einer Host-Verwaltungskonfiguration zu deaktivieren, gehen Sie in Quick Setup wie folgt vor:

1. Wählen Sie im Navigationsbereich Quick Setup aus.
2. Wählen Sie die zu aktualisierende Host-Management-Konfiguration aus.
3. Wählen Sie Actions, Edit configuration (Aktionen, Konfiguration bearbeiten) aus.
4. Deaktivieren Sie das Kontrollkästchen Scan instances for missing patches daily (Instances täglich auf fehlende Patches scannen).

5. Wählen Sie Aktualisieren aus.

> **Note**
>
> Die Verwendung von Patch now (Jetzt patchen) zum Überprüfen eines verwalteten Knotens auf Compliance führt auch zu einem Überschreiben von Patch-Compliance-Daten.

On-Demand-Patchen von verwalteten Knoten

Verwenden Sie die Option Jetzt patchen in Patch Manager, einem Tool in AWS Systems Manager, um Patching-Vorgänge auf Abruf über die Systems-Manager-Konsole auszuführen. Dies bedeutet, dass Sie keinen Zeitplan erstellen müssen, um den Compliance-Status Ihrer verwalteten Knoten zu aktualisieren oder Patches auf nicht kompatiblen Knoten zu installieren. Sie müssen die Systems Manager Manager-Konsole auch nicht zwischen Patch Manager undMaintenance Windows, einem Tool in AWS Systems Manager, umschalten, um ein geplantes Patch-Fenster einzurichten oder zu ändern.

Patch now (Jetzt patchen) ist besonders nützlich, wenn Sie so schnell wie möglich Zero-Day-Updates anwenden oder andere kritische Patches auf Ihren verwalteten Knoten installieren müssen.

> **Note**
>
> Patching auf Abruf wird jeweils für ein AWS-Konto einzelnes AWS-Region Paar unterstützt. Es kann nicht mit Patching-Vorgängen verwendet werden, die auf Patch-Richtlinien basieren. Wir empfehlen die Verwendung von Patch-Richtlinien, um sicherzustellen, dass alle Ihre verwalteten Knoten die Compliance einhalten. Weitere Informationen zur Arbeit mit Patch-Richtlinien finden Sie unter Patch-Richtlinienkonfigurationen in Quick Setup.

Themen

- So funktioniert „Patch now“ (Jetzt patchen)
- Ausführen von „Patch now“ (Jetzt patchen)

So funktioniert „Patch now“ (Jetzt patchen)

Um Patch now (Jetzt patchen) auszuführen, müssen Sie nur zwei erforderliche Einstellungen angeben:

- Ob nur nach fehlenden Patches gescannt werden soll oder ob Patches auf Ihren verwalteten Knoten gescannt und installiert werden sollen
- Auf welchen verwalteten Knoten die Operation ausgeführt werden soll

Wenn die Operation Patch now (Jetzt patchen) läuft, bestimmt sie, welche Patch-Baseline auf die gleiche Weise verwendet werden soll, wie eine für andere Patching-Operationen ausgewählt wird. Wenn ein verwalteter Knoten einer Patch-Gruppe zugeordnet ist, wird die für diese Gruppe angegebene Patch-Baseline verwendet. Wenn der verwaltete Knoten nicht einer Patch-Gruppe zugeordnet ist, verwendet die Operation die Patch-Baseline, die derzeit als Standard für den Betriebssystemtyp des verwalteten Knotens festgelegt ist. Dabei kann es sich um eine vordefinierte Baseline oder um die benutzerdefinierte Baseline handeln, die Sie als Standard festgelegt haben. Weitere Informationen zur Patch-Baseline-Auswahl finden Sie unter Patch-Gruppen.

Zu den Optionen, die Sie für Patch now (Jetzt patchen) angeben können, gehört, ob verwaltete Knoten nach dem Patchen neu gestartet werden sollen, indem Sie einen Amazon Simple Storage Service (Amazon S3)-Bucket zum Speichern von Protokolldaten für den Patchvorgang angeben und Systems-Manager-Dokumente (SSM-Dokumente) als Lebenszyklus-Hooks während des Patchings ausführen.

Parallelitäts- und Fehlerschwellenwerte für „Patch now“ (Jetzt patchen)

Für Patch now-Operationen (Jetzt patchen) werden Optionen für Parallelitäts- und Fehlerschwellen von Patch Manager gehandhabt. Sie müssen weder angeben, wie viele verwaltete Knoten gleichzeitig gepatcht werden sollen, noch wie viele Fehler zulässig sind, bevor die Operation fehlschlägt. Patch Manager wendet die Einstellungen für Parallelitäts- und Fehlerschwellenwert an, die in den folgenden Tabellen beschrieben werden, wenn Sie On-Demand-Patches anwenden.

> **Important**
>
> Die folgenden Schwellenwerte gelten für nur für `Scan and install`-Operationen. Für `Scan`-Operationen versucht Patch Manager bis zu 1.000 Knoten gleichzeitig zu scannen und weiter zu scannen, bis bis zu 1.000 Fehler aufgetreten sind.

Gleichzeitigkeit: Installationsvorgänge

Die Gesamtanzahl von verwalteten Knoten in der Operation Patch now (Jetzt patchen)	Anzahl der gleichzeitig gescannten oder gepatchten verwalteten Knoten
Weniger als 25	1
25–100	5 %
101 bis 1.000	8%
Mehr als 1.000	10 %

Fehlerschwellenwert: Installationsvorgänge

Die Gesamtanzahl von verwalteten Knoten in der Operation Patch now (Jetzt patchen)	Anzahl der zulässigen Fehler, bevor der Vorgang fehlschlägt
Weniger als 25	1
25–100	5
101 bis 1.000	10
Mehr als 1.000	10

Verwenden von „Patch now“ (Jetzt patchen)-Lebenszyklus-Hooks

Patch now (Jetzt patchen) bietet Ihnen die Möglichkeit, SSM Command-Dokumente als Lebenszyklus-Hooks während eines `Install`-Patch-Vorgang auszuführen. Sie können diese Hooks für Aufgaben wie das Herunterfahren von Anwendungen vor dem Patchen oder Ausführen von Zustandsprüfungen für Ihre Anwendungen nach dem Patchen oder nach einem Neustart verwenden.

Weitere Informationen über das Verwenden von Lebenszyklus-Hooks finden Sie unter SSM-Befehlsdokument zum Patchen: `AWS-RunPatchBaselineWithHooks`.

In der folgenden Tabelle werden die Lebenszyklus-Hooks aufgeführt, die für jede der drei Patch now-Neustartoptionen (Jetzt patchen) aufgelistet sowie Beispielanwendungen für jeden Hook.

Lebenszyklus-Hooks und Beispielanwendungen

Neustartoption	Hook: Vor Installation	Hook: Nach Installation	Hook: Beim Verlassen	Hook: Nach geplantem Neustart
Bei Bedarf neu starten	Führen Sie ein SSM-Dokument aus, bevor das Patchen beginnt. Anwendungsbeispiel: Fahren Sie Anwendungen sicher herunter, bevor der Patchvorgang beginnt.	Führen Sie ein SSM-Dokument am Ende der Patching-Operation und vor dem Neustart des verwalteten Knoten aus. Anwendungsbeispiel: Führen Sie Vorgänge wie die Installation von Drittanbieter-Anwendungen vor einem potenziellen Neustart aus.	Führen Sie ein SSM-Dokument aus, nachdem die Patching-Operation abgeschlossen und die Instances neu gestartet wurden. Anwendungsbeispiel: Stellen Sie sicher, dass Anwendungen nach dem Patchen wie erwartet ausgeführt werden.	Nicht verfügbar
Meine Instances nicht neu starten	Wie oben.	Führen Sie ein SSM-Dokument am Ende des Patching-Vorgangs aus. Anwendungsbeispiel: Stellen Sie sicher, dass Anwendungen nach dem	Nicht verfügbar	Nicht verfügbar

Neustartoption	Hook: Vor Installation	Hook: Nach Installation	Hook: Beim Verlassen	Hook: Nach geplantem Neustart
		Patchen wie erwartet ausgeführt werden.		
Neustartzeit planen	Wie oben.	Dasselbe wie für Meine Instances nicht neu starten.	Nicht verfügbar	Führen Sie sofort nach Abschluss eines geplanten Neustarts ein SSM-Dokument aus. Anwendungsbeispiel: Stellen Sie sicher, dass Anwendungen nach dem Neustart wie erwartet ausgeführt werden.

Ausführen von „Patch now“ (Jetzt patchen)

Mit dem folgenden Verfahren können Sie On-Demand-Patches auf Ihre verwalteten Knoten anwenden.

So führen Sie „Patch now“ (Jetzt patchen) aus

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie Patch now (Jetzt patchen) aus.

4. Für Patch-Operation wählen Sie eine der folgenden Optionen aus:

   - Scan (Scannen): Patch Manager findet die Patches, die in Ihren verwalteten Knoten fehlen, installiert sie aber nicht. Sie können die Ergebnisse im Compliance-Dashboard oder in anderen Tools, die Sie zum Anzeigen der Patch-Compliance verwenden, anzeigen.
   - Scan and install (Scannen und installieren): Patch Manager findet die Patches, die in Ihren verwalteten Knoten fehlen, und installiert sie.

5. Führen Sie diesen Schritt nur aus, wenn Sie im vorherigen Schritt Scan und Installation ausgewählt haben. Wählen Sie bei Neustartoption eine der folgenden Optionen aus:

   - Reboot if needed (Bei Bedarf neu starten): Nach der Installation startet Patch Manager verwaltete Knoten nur dann neu, wenn dies zum Abschluss einer Patch-Installation erforderlich ist.
   - Don‘t reboot my instances (Meine Instances nicht neu starten): Nach der Installation startet Patch Manager verwaltete Knoten nicht neu. Sie können verwaltete Knoten manuell neu starten, wenn Sie Neustarts außerhalb von Patch Manager auswählen oder verwalten.
   - Schedule a reboot time (Neustartzeit planen): Geben Sie das Datum, die Uhrzeit und die UTC-Zeitzone an, wenn Patch Manager Ihre verwalteten Knoten neu starten sollen. Nachdem Sie den Patch now (Jetzt patchen)-Vorgang ausgeführt haben, wird der geplante Neustart als Zuordnung in State Manager mit dem Namen `AWS-PatchRebootAssociation` gelistet.

   > **Important**
   >
   > Wenn Sie den Hauptpatchvorgang abbrechen, nachdem er gestartet wurde, State Manager wird die `AWS-PatchRebootAssociation` Zuordnung NICHT automatisch abgebrochen. Um unerwartete Neustarts zu verhindern, müssen Sie manuell `AWS-PatchRebootAssociation` löschen, State Manager falls der geplante Neustart nicht mehr stattfinden soll. Andernfalls kann es zu ungeplanten Systemneustarts kommen, die sich auf die Produktionsauslastung auswirken können. Sie finden diese Zuordnung in der Systems Manager Manager-Konsole unter State Manager> Verknüpfungen.

6. Wählen Sie unter Instances to patch (Zu patchende Instances) eine der folgenden Optionen aus:

   - Alle Instanzen patchen: Patch Manager Führt den angegebenen Vorgang auf allen verwalteten Knoten AWS-Konto in Ihrer aktuellen Version aus AWS-Region.

- Patch only the target instances I specify (Nur die von mir angegebenen Ziel-Instances patchen): Sie geben im nächsten Schritt an, welche verwalteten Knoten anvisiert werden sollen.

7. Führen Sie diesen Schritt nur aus, wenn Sie im vorherigen Schritt Nur die von mir angegebenen Ziel-Instances patchen ausgewählt haben. Identifizieren Sie für den Abschnitt Target selection (Zielauswahl) die Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Knoten manuell auswählen oder eine Ressourcengruppe angeben.

> **Note**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.
> Wenn Sie sich für eine Ressourcengruppe entscheiden, beachten Sie, dass Ressourcengruppen, die auf einem AWS CloudFormation Stack basieren, trotzdem mit dem `aws:cloudformation:stack-id` Standard-Tag gekennzeichnet werden müssen. Wenn es entfernt wurde, kann Patch Manager möglicherweise nicht ermitteln, welche verwalteten Knoten zur Ressourcengruppe gehören.

8. (Optional) Für Patch-Protokollspeicher wählen Sie, wenn Sie Protokolle aus diesem Patchvorgang erstellen und speichern möchten, den S3-Bucket zum Speichern der Protokolle aus.

> **Note**
>
> Bei den S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, handelt es sich um die Berechtigungen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle (hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, und nicht um die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Die für Systems Manager erforderlichen Instance-Berechtigungen konfigurieren oder Die für Systems Manager erforderliche IAM-Servicerolle in Hybrid- und Multi-Cloud-Umgebungen erstellen. Wenn sich der angegebene S3-Bucket in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das dem verwalteten Knoten zugeordnete Instanzprofil oder die IAM-Dienstrolle über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

9. (Optional) Wenn Sie SSM-Dokumente als Lebenszyklus-Hooks an bestimmten Punkten des Patching-Vorgangs ausführen möchten, gehen Sie wie folgt vor:

   - Klicken Sie auf Verwenden von Lebenszyklus-Hooks.
   - Wählen Sie für jeden verfügbaren Hook das SSM-Dokument aus, das am angegebenen Punkt des Vorgangs ausgeführt werden soll:
     - Vor Installation
     - Nach Installation
     - Beim Verlassen
     - Nach geplantem Neustart

   > **Note**
   >
   > Das Standarddokument, `AWS-Noop`, führt keine Vorgänge aus.

10. Wählen Sie Patch now (Jetzt patchen) aus.

    Die Seite Association execution summary (Zusammenfassung der Zuordnungsausführung) wird geöffnet. (Jetzt patchen verwendet jetzt Zuordnungen in State Manager, einem Tool in AWS Systems Manager, für seine Vorgänge.) Im Bereich Operation summary (Operationsübersicht) können Sie den Scan- oder Patch-Status auf den von Ihnen angegebenen verwalteten Knoten überwachen.

Arbeiten mit Patch-Baselines

Eine Patch-Baseline in Patch Manager, einem Tool in AWS Systems Manager, definiert, welche Patches für die Installation in Ihren verwalteten Knoten genehmigt sind. Sie können für Patches einzeln angeben, ob sie genehmigt oder abgelehnt werden. Sie können auch automatische Genehmigungsregeln erstellen, um festzulegen, dass bestimmte Arten von Updates (z. B. wichtige Updates), automatisch genehmigt werden. Die Liste mit den Ablehnungen setzt sowohl die Regeln als auch die Liste mit Genehmigungen außer Kraft. Wenn Sie ausschließlich eine Liste mit genehmigten Patches verwenden möchten, um spezifische Pakete zu installieren, entfernen Sie erst alle automatischen Genehmigungsregeln. Wenn Sie explizit einen Patch als abgelehnt kennzeichnen, wird er nicht genehmigt oder installiert, selbst wenn er mit allen Kriterien in einer automatischen Genehmigungsregel übereinstimmt. Außerdem wird ein Patch nur dann auf einem verwalteten Knoten installiert, wenn er für die Software auf dem Knoten geeignet ist, auch wenn der Patch anderweitig für den verwalteten Knoten genehmigt wurde.

Themen

- So zeigen Sie von AWS vordefinierte Patch-Baselines an
- Arbeiten mit benutzerdefinierten Patch-Baselines
- Festlegen einer vorhandenen Patch-Baseline als Standard

Weitere Informationen

- Patch-Baselines

So zeigen Sie von AWS vordefinierte Patch-Baselines an

Patch Manager, ein Tool in AWS Systems Manager, enthält eine vordefinierte Patch-Baseline für jedes Betriebssystem, das von unterstützt wird. Patch Manager Sie können diese Patch-Baselines verwenden (Sie können sie jedoch nicht anpassen) oder Sie können eine eigene Patch-Baseline erstellen. Nachfolgend wird beschrieben, wie Sie eine vordefinierte Patch-Baseline anzeigen, um sie auf Ihre Anforderungen hin zu überprüfen. Weitere Informationen zu Patch-Baselines finden Sie unter Vordefinierte und benutzerdefinierte Patch-Baselines.

So zeigen Sie AWS vordefinierte Patch-Baselines an

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie in der Liste der Patch-Baselines die Baseline-ID einer der vordefinierten Patch-Baselines aus.

   –oder–

   Wenn Sie in der aktuellen AWS-Region zum ersten Mal auf Patch Manager zugreifen, wählen Sie Mit einer Übersicht beginnen, wählen Sie die Registerkarte Patch-Baselines und wählen Sie dann die Baseline-ID einer der vordefinierten Patch-Baselines.

   > **Note**
   >
   > Für Windows Server werden drei vordefinierte Patch-Baselines bereitgestellt. Die Patch-Baselines `AWS-DefaultPatchBaseline` und `AWS-WindowsPredefinedPatchBaseline-OS` unterstützen nur Betriebssystemupdates auf dem Windows-Betriebssystem selbst. `AWS-DefaultPatchBaseline` wird als

Standard-Patch-Baseline für von Windows Server verwalteten Knoten verwendet, es sei denn, Sie geben eine andere Patch-Baseline an. Die Konfigurationseinstellungen in diesen beiden Patch-Baselines sind identisch. Die neuere der beiden, `AWS-WindowsPredefinedPatchBaseline-OS`, wurde erstellt, um sie von der dritten vordefinierten Patch-Baseline für Windows Server zu unterscheiden. Diese Patch-Baseline, `AWS-WindowsPredefinedPatchBaseline-OS-Applications`, kann verwendet werden, um Patches sowohl auf das Windows Server-Betriebssystem als auch auf unterstützte Anwendungen, die von Microsoft veröffentlicht wurden, anzuwenden.
Weitere Informationen finden Sie unter Festlegen einer vorhandenen Patch-Baseline als Standard.

4. Im Abschnitt Genehmigungsregeln überprüfen Sie die Konfiguration der Patch-Baseline-Konfiguration.
5. Wenn die Konfiguration für Ihre verwalteten Knoten geeignet ist, können Sie direkt mit dem Verfahren Erstellen und Verwalten von Patch-Gruppen fortfahren.

   –oder–

   Fahren Sie zum Erstellen einer eigenen Standard-Patch-Baseline mit dem Thema Arbeiten mit benutzerdefinierten Patch-Baselines fort.

Arbeiten mit benutzerdefinierten Patch-Baselines

Patch Manager, ein Tool in AWS Systems Manager, enthält eine vordefinierte Patch-Baseline für jedes der von Patch Manager unterstützten Betriebssysteme. Sie können diese Patch-Baselines verwenden (Sie können sie jedoch nicht anpassen) oder Sie können eine eigene Patch-Baseline erstellen.

In den folgenden Verfahren wird beschrieben, wie Sie eigene benutzerdefinierte Patch-Baselines erstellen, aktualisieren und löschen. Weitere Informationen zu Patch-Baselines finden Sie unter Vordefinierte und benutzerdefinierte Patch-Baselines.

Themen

- So erstellen Sie eine benutzerdefinierte Patch-Baseline für Linux
- Erstellen einer benutzerdefinierten Patch-Baseline für macOS
- Erstellen einer benutzerdefinierten Patch-Baseline für Windows Server

- Aktualisieren oder Löschen einer benutzerdefinierten Patch-Baseline

So erstellen Sie eine benutzerdefinierte Patch-Baseline für Linux

Verwenden Sie das folgende Verfahren, um eine benutzerdefinierte Patch-Baseline für Linux-verwaltete Knoten in Patch Manager, einem Tool in AWS Systems Manager, zu erstellen.

Informationen zum Erstellen einer Patch-Baseline für macOS-verwaltete Knoten finden Sie unter Erstellen einer benutzerdefinierten Patch-Baseline für macOS. Informationen zum Erstellen einer Patch-Baseline für Windows-verwaltete Knoten finden Sie unter Erstellen einer benutzerdefinierten Patch-Baseline für Windows Server.

So erstellen Sie eine benutzerdefinierte Patch-Baseline für Linux-verwaltete Knoten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Patch-Baselines und dann Patch-Baseline erstellen aus.

   –oder–

   Wenn Sie in der aktuellen AWS-Region zum ersten Mal auf Patch Manager zugreifen, wählen Sie Mit einer Übersicht beginnen, wechseln Sie zur Registerkarte Patch-Baselines und wählen Sie dann Patch-Baseline erstellen.
4. Geben Sie im Feld Name einen Namen für die neue Patch-Baseline ein, z. B. `MyRHELPatchBaseline`.
5. (Optional) Geben Sie im Feld Beschreibung eine Beschreibung für diese Patch-Baseline ein.
6. Wählen Sie unter Operating system (Betriebssystem) ein Betriebssystem aus, z. B. `Red Hat Enterprise Linux`.
7. Wenn Sie diese Patch-Baseline sofort nach der Erstellung als Standard für das ausgewählte Betriebssystem verwenden möchten, aktivieren Sie das Kontrollkästchen neben Diese Patch-Baseline als Standard-Patch-Baseline für ***`operating system name`*** Instances festlegen.

   > **Note**
   >
   > Diese Option ist nur verfügbar, wenn Sie vor der Veröffentlichung der Patch-Richtlinien am 22. Dezember 2022 zum ersten Mal auf Patch Manager zugegriffen haben.

Weitere Informationen zum Festlegen einer vorhandenen Patch-Baseline als Standard finden Sie unter Festlegen einer vorhandenen Patch-Baseline als Standard.

8. Erstellen Sie im Abschnitt Genehmigungsregeln für Betriebssysteme unter Verwendung der Felder ein oder mehrere automatische Genehmigungsregeln.

   - Produkte: Die Version der Betriebssysteme, auf die sich die Genehmigungsregel bezieht, z. B. `RedhatEnterpriseLinux7.4`. Die Standardauswahl ist `All`.
   - Classification (Klassifizierung): Der Typ der Patches, auf die sich die Genehmigungsregel bezieht, z. B. `Security` oder `Enhancement`. Die Standardauswahl ist `All`.

   Tip

   Sie können eine Patch-Baseline konfigurieren, um zu steuern, ob Nebenversions-Upgrades für Linux installiert werden, z. B. RHEL 7.8. Nebenversionsupgrades können vom Patch Manager automatisch installiert werden, wenn das Update im entsprechenden Repository verfügbar ist.
   Im Fall von Linux-Betriebssystemen werden Nebenversionsupgrades nicht konsistent klassifiziert. Sie können als Fehlerbehebungen oder Sicherheitsupdates klassifiziert (oder nicht klassifiziert) werden, selbst innerhalb derselben Kernel-Version. Im Folgenden werden einige Optionen aufgelistet, mit denen Sie steuern können, ob sie von einer Patch-Baseline installiert werden.

   - Option 1: Die umfassendste Genehmigungsregel, die sicherzustellt, dass Nebenversionsupgrades installiert werden, wenn verfügbar, besteht in der Angabe von Classification (Klassifizierung) als `All` (*) und der Auswahl der Option Include nonsecurity updates (Auch andere Updates als Sicherheitsupdates einschließen).
   - Option 2: Um die Installation von Patches für eine Betriebssystemversion sicherzustellen, können Sie ein Platzhalterzeichen (*) verwenden, um das Kernel-Format im Abschnitt Patch exceptions (Patch-Ausnahmen) der Baseline anzugeben. Zum Beispiel ist das Kernel-Format für RHEL 7.* `kernel-3.10.0-*.el7.x86_64`.

     Geben Sie `kernel-3.10.0-*.el7.x86_64` in der Liste Approved patches (Genehmigte Patches) in Ihrer Patch-Baseline ein, um die Anwendung aller Patches einschließlich Nebenversionsupgrades auf Ihren von RHEL 7.* verwalteten Knoten sicherzustellen. (Wenn Sie den genauen Paketnamen eines Nebenversionspatches kennen, können Sie diesen stattdessen eingeben.)

- Option 3: Mithilfe des InstallOverrideListParameters im `AWS-RunPatchBaseline` Dokument haben Sie die größtmögliche Kontrolle darüber, welche Patches auf Ihre verwalteten Knoten angewendet werden, einschließlich kleinerer Versions-Upgrades. Weitere Informationen finden Sie unter SSM-Befehlsdokument zum Patchen: `AWS-RunPatchBaseline`.

- Schweregrad: Der Schweregradwert von Patches, auf den die Regel anzuwenden ist, z. B. `Critical`. Die Standardauswahl ist `All`.
- Automatische Genehmigung: Die Methode zum Auswählen von Patches für die automatische Genehmigung.

> **Note**
>
> Da es nicht möglich ist, die Veröffentlichungsdaten von Updatepaketen für Ubuntu Server zuverlässig zu bestimmen, werden die Optionen für die automatische Genehmigung für dieses Betriebssystem nicht unterstützt.

  - Approve patches after a specified number of days (Patches nach einer bestimmten Anzahl von Tagen genehmigen): Die Anzahl der Tage, die der Patch Manager warten muss, nachdem ein Patch veröffentlicht oder zuletzt aktualisiert wurde, bevor ein Patch automatisch genehmigt wird. Sie können jede Ganzzahl von Null (0) bis 360 eingeben. Für die meisten Szenarien empfehlen wir, nicht länger als 100 Tage zu warten.
  - Patches genehmigen, die bis zu einem bestimmten Datum veröffentlicht wurden: Das Datum der Patch-Veröffentlichung, an dem der Patch Manager automatisch alle Patches anwendet, die bis zu diesem Datum veröffentlicht oder aktualisiert wurden. Wenn Sie beispielsweise den 7. Juli 2023 angeben, werden Patches, die am oder nach dem 8. Juli 2023 veröffentlicht oder zuletzt aktualisiert wurden, nicht automatisch installiert.
- (Optional) Konformitätsbericht : Der Schweregrad, den Sie Patches zuweisen möchten, die von der Baseline genehmigt wurden (z. B. `Critical` oder `High`).

> **Note**
>
> Wenn Sie eine Konformitätsberichtsstufe angeben und der Patch-Status eines genehmigten Patches als `Missing` gemeldet wird, dann entspricht der insgesamt

gemeldete Konformitätsschweregrad der Patch-Baseline dem von Ihnen angegebenen Schweregrad.

- Include non-security updates (Nicht sicherheitsrelevante Updates einbeziehen): Aktivieren Sie das Kontrollkästchen zum Installieren von nicht sicherheitsrelevanten Linux-Betriebssystem-Patches, die im Quell-Repository verfügbar gemacht wurden, zusätzlich zu den sicherheitsrelevanten Patches.

Weitere Informationen zum Arbeiten mit Genehmigungsregeln in einer benutzerdefinierten Patch-Baseline finden Sie unter Benutzerdefinierte Baselines.

9. Wenn Sie zusätzlich zu den Patches, die Ihren Genehmigungsregeln entsprechen, alle Patches ausdrücklich genehmigen möchten, gehen Sie im Abschnitt Patch-Ausnahmen wie folgt vor:

   - Geben Sie im Feld Genehmigte Patches eine durch Komma getrennte Liste der Patches ein, die Sie genehmigen möchten.

     Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen.
   - (Optional) Weisen Sie in der Liste Compliance-Stufe genehmigter Patches den Patches in der Liste eine Compliance-Stufe zu.
   - Wenn genehmigte Patches, die Sie angeben, nicht sicherheitsbezogen sind, wählen Sie das Kästchen Genehmigte Patches umfassen nicht sicherheitsrelevante Updates aus, damit diese Patches ebenfalls auf Ihrem Linux-Betriebssystem installiert werden.

10. Wenn Sie Patches ablehnen möchten, die ansonsten Ihren Genehmigungsregeln entsprechen, gehen Sie im Abschnitt Patch-Ausnahmen wie folgt vor:

    - Geben Sie im Feld Abgelehnte Patches eine durch Komma getrennte Liste der Patches ein, die Sie ablehnen möchten.

      Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen.
    - Wählen Sie in der Liste Aktion für abgelehnte Patches die Aktion aus, die Patch Manager für Patches in der Liste Abgelehnte Patches ausführen soll.
      - Als Abhängigkeit zulassen: Ein Paket in der Liste Abgelehnte Patches wird nur installiert, wenn es sich um eine Abhängigkeit eines anderen Pakets handelt. Es gilt als konform

mit der Patch-Baseline und sein Status wird als gemeldet InstalledOther. Dies ist die Standardaktion, wenn keine Option ausgewählt ist.

- Blockieren: Pakete in der Liste Abgelehnte Patches und Pakete, die diese als Abhängigkeiten enthalten, werden unter keinen Umständen von Patch Manager installiert. Wenn ein Paket installiert wurde, bevor es zur Liste der abgelehnten Patches hinzugefügt wurde, oder danach außerhalb von Patch Manager installiert wird, gilt es als nicht konform mit der Patch-Baseline und sein Status wird als InstalledRejected gemeldet.

> **Note**
>
> Patch Manager sucht rekursiv nach Patch-Abhängigkeiten.

11. (Optional) Wenn Sie alternative Patch-Repositorys für verschiedene Versionen eines Betriebssystems angeben möchten, z. B. AmazonLinux2016.03 und AmazonLinux2017.09, gehen Sie für jedes Produkt im Abschnitt Patch-Quellen wie folgt vor:

    - Geben Sie in Name (Name) einen Namen ein, um Sie bei der Identifizierung der Quellkonfiguration zu unterstützen.
    - Wählen Sie unter Product (Produkt) die Version der Betriebssysteme aus, für die das Patch-Quell-Repository bestimmt ist, z. B. `RedhatEnterpriseLinux7.4`.
    - Geben Sie unter Konfiguration den Wert der zu verwendenden Repository-Konfiguration im entsprechenden Format ein:

      Example for yum repositories

      ```
      [main]
      name=MyCustomRepository
      baseurl=https://my-custom-repository
      enabled=1
      ```

      > **Tip**
      >
      > Informationen zu anderen Optionen für Ihre Yum-Repository-Konfiguration finden Sie unter dnf.conf (5).

Examples for Ubuntu Server and Debian Server

```
deb http://security.ubuntu.com/ubuntu jammy main
```

```
deb https://site.example.com/debian distribution component1
component2 component3
```

Repository-Informationen für Ubuntu Server-Repositorys müssen in einer einzigen Zeile angegeben werden. Weitere Beispiele und Informationen finden Sie unter jammy (5) sources.list.5.gz auf der Website Ubuntu Server Manuals und unter sources.list-Format im Debian-Wiki.

Wählen Sie Add another source aus, um ein Quell-Repository für jede zusätzliche Betriebssystemversion anzugeben, bis maximal 20.

Weitere Informationen über alternative Quell-Patch-Repositorys finden Sie unter So geben Sie ein alternatives Patch-Quell-Repository an (Linux).

12. (Optional) Wenden Sie unter Tags verwalten ein oder mehrere name/value Tag-Schlüsselpaare auf die Patch-Baseline an.

    Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Sie können beispielsweise eine Patch-Baseline kennzeichnen, um den Schweregrad der angegebenen Patches, die Betriebssystemfamilie, auf die sie sich bezieht, und den Umgebungstyp zu identifizieren. In diesem Fall könnten Sie Tags angeben, die den folgenden name/value Schlüsselpaaren ähneln:

    - `Key=PatchSeverity,Value=Critical`
    - `Key=OS,Value=RHEL`
    - `Key=Environment,Value=Production`

13. Wählen Sie die Option Patch-Baseline erstellen.

Erstellen einer benutzerdefinierten Patch-Baseline für macOS

Verwenden Sie das folgende Verfahren, um eine benutzerdefinierte Patch-Baseline für macOS-verwaltete Knoten in Patch Manager zu erstellen, einem Tool in AWS Systems Manager.

Informationen zum Erstellen einer Patch-Baseline für Windows Server-verwaltete Knoten finden Sie unter Erstellen einer benutzerdefinierten Patch-Baseline für Windows Server. Informationen zum Erstellen einer Patch-Baseline für Linux-verwaltete Knoten finden Sie unter So erstellen Sie eine benutzerdefinierte Patch-Baseline für Linux.

> **Note**
>
> macOSwird nicht in allen unterstützt AWS-Regionen. Weitere Informationen zum EC2 Amazon-Support für macOS finden Sie unter Amazon EC2 Mac-Instances im EC2 Amazon-Benutzerhandbuch.

So erstellen Sie eine benutzerdefinierte Patch-Baseline für macOS-verwaltete Knoten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Patch-Baselines und dann Patch-Baseline erstellen aus.

   –oder–

   Wenn Sie in der aktuellen AWS-Region zum ersten Mal auf Patch Manager zugreifen, wählen Sie Mit einer Übersicht beginnen, wechseln Sie zur Registerkarte Patch-Baselines und wählen Sie dann Patch-Baseline erstellen.
4. Geben Sie im Feld Name einen Namen für die neue Patch-Baseline ein, z. B. `MymacOSPatchBaseline`.
5. (Optional) Geben Sie im Feld Beschreibung eine Beschreibung für diese Patch-Baseline ein.
6. Wählen Sie unter Betriebssystem die Option macOS aus.
7. Wenn Sie die Patch-Baseline direkt nach dem Erstellen als Standard für macOS verwenden möchten, aktivieren Sie das Kontrollkästchen Set this patch baseline as the default patch baseline for macOS instances (Diese Patch-Baseline als Standard-Patch-Baseline für macOs-Instances festlegen).

   > **Note**
   >
   > Diese Option ist nur verfügbar, wenn Sie vor der Veröffentlichung der Patch-Richtlinien am 22. Dezember 2022 zum ersten Mal auf Patch Manager zugegriffen haben.

Weitere Informationen zum Festlegen einer vorhandenen Patch-Baseline als Standard finden Sie unter Festlegen einer vorhandenen Patch-Baseline als Standard.

8. Erstellen Sie im Abschnitt Genehmigungsregeln für Betriebssysteme unter Verwendung der Felder ein oder mehrere automatische Genehmigungsregeln.

   - Produkte: Die Version der Betriebssysteme, auf die sich die Genehmigungsregel bezieht, z. B. `BigSur11.3.1` oder `Ventura13.7`. Die Standardauswahl ist `All`.
   - Klassifizierung: Der oder die Paketmanager, auf den/die während des Patchvorgangs Pakete angewendet werden sollen. Sie können aus den folgenden Optionen auswählen:
     - softwareupdate
     - installer
     - brew
     - brew cask

     Die Standardauswahl ist `All`.
   - (Optional) Konformitätsbericht : Der Schweregrad, den Sie Patches zuweisen möchten, die von der Baseline genehmigt wurden (z. B. `Critical` oder `High`).

     > **Note**
     >
     > Wenn Sie eine Konformitätsberichtsstufe angeben und der Patch-Status eines genehmigten Patches als `Missing` gemeldet wird, dann entspricht der insgesamt gemeldete Konformitätsschweregrad der Patch-Baseline dem von Ihnen angegebenen Schweregrad.

   Weitere Informationen zum Arbeiten mit Genehmigungsregeln in einer benutzerdefinierten Patch-Baseline finden Sie unter Benutzerdefinierte Baselines.

9. Wenn Sie zusätzlich zu den Patches, die Ihren Genehmigungsregeln entsprechen, alle Patches ausdrücklich genehmigen möchten, gehen Sie im Abschnitt Patch-Ausnahmen wie folgt vor:

   - Geben Sie im Feld Genehmigte Patches eine durch Komma getrennte Liste der Patches ein, die Sie genehmigen möchten.

     Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen.

- (Optional) Weisen Sie in der Liste Compliance-Stufe genehmigter Patches den Patches in der Liste eine Compliance-Stufe zu.

10. Wenn Sie Patches ablehnen möchten, die ansonsten Ihren Genehmigungsregeln entsprechen, gehen Sie im Abschnitt Patch-Ausnahmen wie folgt vor:

    - Geben Sie im Feld Abgelehnte Patches eine durch Komma getrennte Liste der Patches ein, die Sie ablehnen möchten.

      Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen.

    - Wählen Sie in der Liste Aktion für abgelehnte Patches die Aktion aus, die Patch Manager für Patches in der Liste Abgelehnte Patches ausführen soll.
      - Als Abhängigkeit zulassen: Ein Paket in der Liste Abgelehnte Patches wird nur installiert, wenn es sich um eine Abhängigkeit eines anderen Pakets handelt. Es gilt als konform mit der Patch-Baseline und sein Status wird als gemeldet InstalledOther. Dies ist die Standardaktion, wenn keine Option ausgewählt ist.
      - Blockieren: Pakete in der Liste Abgelehnte Patches und Pakete, die diese als Abhängigkeiten enthalten, werden unter keinen Umständen von Patch Manager installiert. Wenn ein Paket installiert wurde, bevor es zur Liste der abgelehnten Patches hinzugefügt wurde, oder danach außerhalb von Patch Manager installiert wird, gilt es als nicht konform mit der Patch-Baseline und sein Status wird als InstalledRejected gemeldet.

11. (Optional) Wenden Sie unter Tags verwalten ein oder mehrere name/value Tag-Schlüsselpaare auf die Patch-Baseline an.

    Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Sie können beispielsweise eine Patch-Baseline kennzeichnen, um den Schweregrad der angegebenen Patches, den Paketmanager, auf den sie sich bezieht, und den Umgebungstyp zu identifizieren. In diesem Fall könnten Sie Tags angeben, die den folgenden name/value Schlüsselpaaren ähneln:

    - `Key=PatchSeverity,Value=Critical`
    - `Key=PackageManager,Value=softwareupdate`
    - `Key=Environment,Value=Production`

12. Wählen Sie die Option Patch-Baseline erstellen.

Erstellen einer benutzerdefinierten Patch-Baseline für Windows Server

Verwenden Sie das folgende Verfahren, um eine benutzerdefinierte Patch-Baseline für Windows-verwaltete Knoten in Patch Manager zu erstellen, einem Tool in AWS Systems Manager.

Informationen zum Erstellen einer Patch-Baseline für Linux-verwaltete Knoten finden Sie unter So erstellen Sie eine benutzerdefinierte Patch-Baseline für Linux. Informationen zum Erstellen einer Patch-Baseline für macOS-verwaltete Knoten finden Sie unter Erstellen einer benutzerdefinierten Patch-Baseline für macOS.

Ein Beispiel für das Erstellen einer Patch-Baseline, die auf die Installation von Windows Service Packs eingeschränkt ist, finden Sie unter So erstellen Sie eine Patch-Baseline für die Installation von Windows Service Packs mithilfe der Konsole.

So erstellen Sie eine benutzerdefinierte Patch-Baseline (Windows)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Patch-Baselines und dann Patch-Baseline erstellen aus.

   –oder–

   Wenn Sie in der aktuellen AWS-Region zum ersten Mal auf Patch Manager zugreifen, wählen Sie Mit einer Übersicht beginnen, wechseln Sie zur Registerkarte Patch-Baselines und wählen Sie dann Patch-Baseline erstellen.
4. Geben Sie im Feld Name einen Namen für die neue Patch-Baseline ein, z. B. `MyWindowsPatchBaseline`.
5. (Optional) Geben Sie im Feld Beschreibung eine Beschreibung für diese Patch-Baseline ein.
6. Wählen Sie unter Betriebssystem die Option `Windows` aus.
7. Wählen Sie unter Compliance-Status für verfügbare Sicherheitsupdates den Status aus, den Sie Sicherheitspatches zuweisen möchten, die zwar verfügbar, aber nicht genehmigt sind, weil sie die in der Patch-Baseline angegebenen Installationskriterien (Nicht konform oder Konform) nicht erfüllen.

   Beispielszenario: Sicherheitspatches, die Sie möglicherweise installieren möchten, können übersprungen werden, wenn Sie eine lange Wartezeit für die Installation nach der Veröffentlichung eines Patches angegeben haben. Wenn während der von Ihnen angegebenen

Wartezeit ein Update für den Patch veröffentlicht wird, beginnt die Wartezeit für die Installation des Patches von vorne. Wenn die Wartezeit zu lang ist, können mehrere Versionen des Patches veröffentlicht, aber nie installiert werden.

8. Wenn Sie diese Patch-Baseline direkt nach dem Erstellen als Standard für Windows verwenden möchten, wählen Sie Set this patch baseline as the default patch baseline for Windows Server instances (Diese Patch-Baseline als Standard-Patch-Baseline für Windows Server-Instances festlegen) aus.

> **Note**
>
> Diese Option ist nur verfügbar, wenn Sie vor der Veröffentlichung der Patch-Richtlinien am 22. Dezember 2022 zum ersten Mal auf Patch Manager zugegriffen haben. Weitere Informationen zum Festlegen einer vorhandenen Patch-Baseline als Standard finden Sie unter Festlegen einer vorhandenen Patch-Baseline als Standard.

9. Erstellen Sie im Abschnitt Approval Rules for operating-systems (Genehmigungsregeln für Betriebssysteme) unter Verwendung der Felder ein oder mehrere automatische Genehmigungsregeln.

   - Produkte: Die Version der Betriebssysteme, auf die sich die Genehmigungsregel bezieht, z. B. `WindowsServer2012`. Die Standardauswahl ist `All`.
   - Classification (Klassifizierung): Der Typ der Patches, auf die sich die Genehmigungsregel bezieht, z. B. `CriticalUpdates`, `Drivers` und `Tools`. Die Standardauswahl ist `All`.

     > **Tip**
     >
     > Sie können Windows Service Pack-Installationen in die Genehmigungsregeln einschließen, indem Sie die `ServicePacks` einschließen oder `All` in der Liste Classification (Klassifizierung) auswählen. Ein Beispiel finden Sie unter So erstellen Sie eine Patch-Baseline für die Installation von Windows Service Packs mithilfe der Konsole.

   - Schweregrad: Der Schweregradwert von Patches, auf den die Regel anzuwenden ist, z. B. `Critical`. Die Standardauswahl ist `All`.
   - Automatische Genehmigung: Die Methode zum Auswählen von Patches für die automatische Genehmigung.

- Patches nach einer bestimmten Anzahl von Tagen genehmigen: Die Anzahl der Tage, die der Patch Manager warten muss, nachdem ein Patch veröffentlicht oder aktualisiert wurde, bevor ein Patch automatisch genehmigt wird. Sie können jede Ganzzahl von Null (0) bis 360 eingeben. Für die meisten Szenarien empfehlen wir, nicht länger als 100 Tage zu warten.
- Patches genehmigen, die bis zu einem bestimmten Datum veröffentlicht wurden: Das Datum der Patch-Veröffentlichung, an dem der Patch Manager automatisch alle Patches anwendet, die bis zu diesem Datum veröffentlicht oder aktualisiert wurden. Wenn Sie beispielsweise den 07. Juli 2023 angeben, werden Patches, die am oder nach dem 08. Juli 2023 veröffentlicht oder zuletzt aktualisiert wurden, nicht automatisch installiert.

- (Optional) Compliance-Bericht : Der Schweregrad, den Sie Patches zuweisen möchten, die von der Baseline genehmigt wurden (z. B. `High`).

> **Note**
>
> Wenn Sie eine Konformitätsberichtsstufe angeben und der Patch-Status eines genehmigten Patches als `Missing` gemeldet wird, dann entspricht der insgesamt gemeldete Konformitätsschweregrad der Patch-Baseline dem von Ihnen angegebenen Schweregrad.

10. (Optional) Erstellen Sie im Abschnitt Approval Rules for applications (Genehmigungsregeln für Anwendungen) unter Verwendung der Felder ein oder mehrere automatische Genehmigungsregeln.

> **Note**
>
> Anstatt Genehmigungsregeln anzugeben, können Sie Listen genehmigter und abgelehnter Patches als Patch-Ausnahmen angeben. Siehe Schritte 10 und 11.

- Product family (Produktfamilie): Die allgemeine Microsoft-Produktfamilie, für die Sie eine Regel festlegen möchten, z. B. `Office` oder `Exchange Server`.
- Produkte: Die Version der Anwendung, auf die sich die Genehmigungsregel bezieht, z. B. `Office 2016` oder `Active Directory Rights Management Services Client 2.0 2016`. Die Standardauswahl ist `All`.
- Classification (Klassifikation): Der Typ der Patches, auf die sich die Genehmigungsregel bezieht, z. B. `CriticalUpdates`. Die Standardauswahl ist `All`.

- Severity (Schweregrad): Der Schweregradwert von Patches, auf den die Regel anzuwenden ist, z. B. `Critical`. Die Standardauswahl ist `All`.
- Automatische Genehmigung: Die Methode zum Auswählen von Patches für die automatische Genehmigung.
  - Patches nach einer bestimmten Anzahl von Tagen genehmigen: Die Anzahl der Tage, die der Patch Manager warten muss, nachdem ein Patch veröffentlicht oder aktualisiert wurde, bevor ein Patch automatisch genehmigt wird. Sie können jede Ganzzahl von Null (0) bis 360 eingeben. Für die meisten Szenarien empfehlen wir, nicht länger als 100 Tage zu warten.
  - Patches genehmigen, die bis zu einem bestimmten Datum veröffentlicht wurden: Das Datum der Patch-Veröffentlichung, an dem der Patch Manager automatisch alle Patches anwendet, die bis zu diesem Datum veröffentlicht oder aktualisiert wurden. Wenn Sie beispielsweise den 7. Juli 2023 angeben, werden Patches, die am oder nach dem 8. Juli 2023 veröffentlicht oder zuletzt aktualisiert wurden, nicht automatisch installiert.
- (Optional) Konformitätsbericht : Der Schweregrad, den Sie Patches zuweisen möchten, die von der Baseline genehmigt wurden (z. B. `Critical` oder `High`).

  > **Note**
  >
  > Wenn Sie eine Konformitätsberichtsstufe angeben und der Patch-Status eines genehmigten Patches als `Missing` gemeldet wird, dann entspricht der insgesamt gemeldete Konformitätsschweregrad der Patch-Baseline dem von Ihnen angegebenen Schweregrad.

11. (Optional) Wenn Sie Patches explizit genehmigen möchten, anstatt Patches gemäß Genehmigungsregeln auszuwählen, gehen Sie im Abschnitt Patch-Ausnahmen folgendermaßen vor:

    - Geben Sie im Feld Genehmigte Patches eine durch Komma getrennte Liste der Patches ein, die Sie genehmigen möchten.

      Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen.
    - (Optional) Weisen Sie in der Liste Compliance-Stufe genehmigter Patches den Patches in der Liste eine Compliance-Stufe zu.
12. Wenn Sie Patches ablehnen möchten, die ansonsten Ihren Genehmigungsregeln entsprechen, gehen Sie im Abschnitt Patch-Ausnahmen wie folgt vor:

- Geben Sie im Feld Abgelehnte Patches eine durch Komma getrennte Liste der Patches ein, die Sie ablehnen möchten.

  Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen.

- Wählen Sie in der Liste Aktion für abgelehnte Patches die Aktion aus, die Patch Manager für Patches in der Liste Abgelehnte Patches ausführen soll.
  - Als Abhängigkeit zulassen: unterstützt das Konzept der Paketabhängigkeiten Windows Server nicht. Wenn ein Paket in der Liste der abgelehnten Patches enthalten ist und bereits auf dem Knoten installiert ist, wird sein Status als `INSTALLED_OTHER` gemeldet. Jedes Paket, das noch nicht auf dem Knoten installiert ist, wird übersprungen.
  - Blockieren: Pakete in der Liste der abgelehnten Patches werden von Patch Manager unter keinen Umständen installiert. Wenn ein Paket installiert wurde, bevor es zur Liste der abgelehnten Patches hinzugefügt wurde, oder danach außerhalb von Patch Manager installiert wird, gilt es als nicht konform mit der Patch-Baseline und sein Status wird als `INSTALLED_REJECTED` gemeldet.

  Weitere Informationen zu Aktionen für abgelehnte Pakete finden Sie unter Optionen für die Liste abgelehnter Patches in benutzerdefinierten Patch-Baselines.

13. (Optional) Wählen Sie für Tags verwalten ein oder mehrere Tag-Schlüsselname/Wertpaare für die Patch-Baseline aus.

    Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Sie können beispielsweise eine Patch-Baseline kennzeichnen, um den Schweregrad der angegebenen Patches, die Betriebssystemfamilie, auf die sie sich bezieht, und den Umgebungstyp zu identifizieren. In diesem Fall können Sie etwa Tags mit den folgenden Schlüsselnamen/Wertpaaren angeben:

    - `Key=PatchSeverity,Value=Critical`
    - `Key=OS,Value=RHEL`
    - `Key=Environment,Value=Production`

14. Wählen Sie die Option Patch-Baseline erstellen.

Aktualisieren oder Löschen einer benutzerdefinierten Patch-Baseline

Sie können eine benutzerdefinierte Patch-Baseline, die Sie erstellt haben, in Patch Manager, einem Tool in AWS Systems Manager, aktualisieren oder löschen. Wenn Sie eine Patch-Baseline aktualisieren, können Sie deren Namen oder Beschreibung, die Genehmigungsregeln sowie die Ausnahmen für genehmigte und abgelehnte Patches ändern. Sie können auch die Tags aktualisieren, die auf die Patch-Baseline angewendet werden. Sie können den Betriebssystemtyp, für den eine Patch-Baseline erstellt wurde, nicht ändern, und Sie können keine Änderungen an einer vordefinierten Patch-Baseline vornehmen, die von bereitgestellt wird AWS.

Aktualisieren oder Löschen einer Patch-Baseline

Gehen Sie wie folgt vor, um eine Patch-Baseline zu aktualisieren oder zu löschen.

> **Important**
>
> Gehen Sie vorsichtig vor, wenn Sie eine benutzerdefinierte Patch-Baseline löschen, die möglicherweise von einer Patch-Richtlinienkonfiguration in Quick Setup verwendet wird. Wenn Sie eine Patch-Richtlinienkonfiguration in Quick Setup verwenden, werden Aktualisierungen, die Sie an benutzerdefinierten Patch-Baselines vornehmen, einmal pro Stunde mit Quick Setup synchronisiert.
> Wenn eine benutzerdefinierte Patch-Baseline gelöscht wird, auf die in einer Patch-Richtlinie verwiesen wurde, wird auf der Seite mit den Quick Setup-Konfigurationsdetails ein Banner für Ihre Patch-Richtlinie angezeigt. Das Banner informiert Sie darüber, dass die Patch-Richtlinie auf eine nicht mehr vorhandene Patch-Baseline verweist und nachfolgende Patching-Vorgänge fehlschlagen werden. Kehren Sie in diesem Fall zur Seite Quick Setup-Konfigurationen zurück, wählen Sie die Patch Manager-Konfiguration aus und wählen Sie Aktionen, Konfiguration bearbeiten. Der Name der gelöschten Patch-Baseline wird hervorgehoben, und Sie müssen eine neue Patch-Baseline für das betroffene Betriebssystem auswählen.

So aktualisieren oder löschen Sie eine Patch-Baseline

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.

3. Wählen Sie die Patch-Baseline aus, die Sie aktualisieren oder löschen möchten, und führen Sie dann einen der folgenden Schritte aus:

   - Um die Patch-Baseline aus Ihrem zu entfernen AWS-Konto, wählen Sie Löschen. Sie werden aufgefordert, Ihre Aktionen zu bestätigen.
   - Wenn Sie den Namen oder die Beschreibung, die Genehmigungsregeln oder Patch-Ausnahmen der Patch-Baseline ändern möchten, wählen Sie Edit (Bearbeiten) aus. Nehmen Sie auf der Seite Edit patch baseline (Patch-Baseline bearbeiten) die gewünschten Änderungen vor und klicken Sie dann auf Save changes (Änderungen speichern).
   - Wenn Sie auf die Patch-Baseline angewendete Tags hinzufügen, ändern oder löschen möchten, klicken Sie auf die Registerkarte Tags (Tags) und dann auf Edit tags (Tags bearbeiten). Nehmen Sie auf der Seite Edit patch baseline tags (Patch-Baseline-Tags bearbeiten) die gewünschten Änderungen vor und klicken Sie dann auf Save changes (Änderungen speichern).

   Weitere Informationen zu den Konfigurationsoptionen, die Sie ausführen können, finden Sie unter Arbeiten mit benutzerdefinierten Patch-Baselines.

Festlegen einer vorhandenen Patch-Baseline als Standard

> **Important**
>
> Alle hier getroffenen Standardauswahlen für die Patch-Baseline gelten nicht für Patching-Vorgänge, die auf einer Patch-Richtlinie basieren. Patch-Richtlinien verwenden ihre eigenen Patch-Baseline-Spezifikationen. Weitere Informationen zu Patch-Richtlinien finden Sie unter Patch-Richtlinienkonfigurationen in Quick Setup.

Bereits beim Erstellen einer benutzerdefinierten Patch-Baseline in Patch Manager, einem Tool in AWS Systems Manager, können Sie die Baseline als Standard für den zugehörigen Betriebssystemtyp festlegen. Weitere Informationen finden Sie unter Arbeiten mit benutzerdefinierten Patch-Baselines.

Sie können auch eine vorhandene Patch-Baseline als Standard für einen Betriebssystemtyp festlegen.

> **Note**
>
> Welche Schritte Sie ausführen, hängt davon ab, ob Sie vor oder nach der Veröffentlichung der Patch-Richtlinien am 22. Dezember 2022 auf Patch Manager zugegriffen haben. Wenn Sie Patch Manager vor diesem Datum verwendet haben, können Sie das Konsolenverfahren verwenden. Andernfalls verwenden Sie das AWS CLI-Verfahren. Das Menü Aktionen, auf das im Konsolenverfahren verwiesen wird, wird in Regionen, in denen Patch Manager vor der Veröffentlichung der Patch-Richtlinien nicht verwendet wurde, nicht angezeigt.

So legen Sie eine Patch-Baseline als Standard fest

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Patch-Baselines aus.
4. Wählen Sie in der Liste der Patch-Baselines die Schaltfläche einer Patch-Baseline aus, die derzeit nicht als Standard für ein Betriebssystem festgelegt ist.

   Die Spalte Default baseline (Standard-Baseline) gibt an, welche Baselines derzeit als Standardwerte festgelegt sind.
5. Wählen Sie im Menü Actions (Aktionen) die Option Set default patch baseline (Standard-Patch-Baseline festlegen) aus.

   > **Important**
   >
   > Das Aktionsmenü ist nicht verfügbar, wenn Sie vor dem 22. Dezember 2022 nicht mit Patch Manager im aktuellen AWS-Konto und der Region gearbeitet haben. Weitere Informationen finden Sie in der Anmerkung weiter oben in diesem Thema.
6. Wählen Sie im Bestätigungsdialogfeld Set default (Als Standard festlegen) aus.

So legen Sie eine Patch-Baseline als Standard fest (AWS CLI)

1. Führen Sie den Befehl describe-patch-baselines aus, um eine Liste der verfügbaren Patch-Baselines und ihrer IDs und Amazon-Ressourcennamen (ARNs) anzuzeigen.

```
aws ssm describe-patch-baselines
```

2. Führen Sie den Befehl register-default-patch-baseline aus, um eine Baseline als Standard für das Betriebssystem festzulegen, mit dem sie verknüpft ist. Ersetzen Sie *baseline-id-or-ARN* durch die ID der zu verwendenden benutzerdefinierten Patch-Baseline oder vordefinierten Baseline.

   Linux & macOS

   ```
   aws ssm register-default-patch-baseline \
       --baseline-id baseline-id-or-ARN
   ```

   Im Folgenden finden Sie ein Beispiel für die Festlegung einer benutzerdefinierten Baseline als Standard.

   ```
   aws ssm register-default-patch-baseline \
       --baseline-id pb-abc123cf9bEXAMPLE
   ```

   Im Folgenden finden Sie ein Beispiel für die Einstellung einer vordefinierten Baseline, die standardmäßig von AWS verwaltet wird.

   ```
   aws ssm register-default-patch-baseline \
       --baseline-id arn:aws:ssm:us-east-2:733109147000:patchbaseline/
   pb-0574b43a65ea646e
   ```

   Windows Server

   ```
   aws ssm register-default-patch-baseline ^
       --baseline-id baseline-id-or-ARN
   ```

   Im Folgenden finden Sie ein Beispiel für die Festlegung einer benutzerdefinierten Baseline als Standard.

   ```
   aws ssm register-default-patch-baseline ^
       --baseline-id pb-abc123cf9bEXAMPLE
   ```

   Im Folgenden finden Sie ein Beispiel für die Einstellung einer vordefinierten Baseline, die standardmäßig von AWS verwaltet wird.

```
aws ssm register-default-patch-baseline ^
    --baseline-id arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-071da192df1226b63
```

Anzeigen verfügbarer Patches

Mit Patch Manager, einem Tool in AWS Systems Manager, können Sie alle verfügbaren Patches für ein bestimmtes Betriebssystem und optional eine bestimmte Betriebssystemversion anzeigen.

> **Tip**
>
> Um eine Liste verfügbarer Patches zu generieren und diese in einer Datei zu speichern, können Sie den Befehl describe-available-patches verwenden und Ihre bevorzugte Ausgabe angeben.

Anzeigen verfügbarer Patches

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wählen Sie die Registerkarte Patches aus.

   –oder–

   Wenn Sie in der aktuellen AWS-Region zum ersten Mal auf Patch Manager zugreifen, wählen Sie Mit einer Übersicht beginnen und dann die Registerkarte Patches aus.

   > **Note**
   >
   > Für Windows Server zeigt die Registerkarte Patches Updates an, die vom Windows Server Update Service (WSUS) verfügbar sind.

4. Für Betriebssystem wählen Sie das Betriebssystem aus, für das Sie verfügbare Patches anzeigen möchten, z. B. `Windows` oder `Amazon Linux`.
5. (Optional) Für Product (Produkt) wählen Sie eine Betriebssystemversion aus, z. B. `WindowsServer2019` oder `AmazonLinux2018.03`.

6. (Optional) Um Informationsspalten für Ihre Ergebnisse hinzuzufügen oder zu entfernen, wählen Sie die Konfigurationsschaltfläche ( ) oben rechts in der Liste Patches aus. (Standardmäßig zeigt die Registerkarte Patches nur Spalten für einige der verfügbaren Patch-Metadaten an.)

   Informationen zu den Arten von Metadaten, die Sie Ihrer Ansicht hinzufügen können, finden Sie unter Patch in der AWS Systems Manager-API-Referenz.

Erstellen und Verwalten von Patch-Gruppen

Wenn Sie in Ihrem Betrieb keine Patching-Richtlinien verwenden, können Sie Ihre Patching-Aufgaben organisieren, indem Sie verwaltete Knoten mithilfe von Tags zu Patch-Gruppen hinzufügen.

> **Note**
>
> Patch-Gruppen werden nicht in Patch-Vorgängen verwendet, die auf Patch-Richtlinien basieren. Weitere Informationen zur Arbeit mit Patch-Richtlinien finden Sie unter Patch-Richtlinienkonfigurationen in Quick Setup.
> Die Patchgruppenfunktion wird in der Konsole für Konto-Regionen-Paare nicht unterstützt, die vor der Veröffentlichung der Patch-Richtlinienunterstützung am 22. Dezember 2022 noch keine Patchgruppen verwendet haben. Die Patchgruppenfunktion ist weiterhin für Konto-Regionen-Paare verfügbar, die vor diesem Datum mit der Verwendung von Patchgruppen begonnen haben.

Um Tags bei Patching-Operationen zu verwenden, müssen Sie den Tag-Schlüssel `Patch Group` oder `PatchGroup` auf Ihre verwalteten Knoten anwenden. Sie müssen auch den Namen, den Sie der Patch-Gruppe geben möchten, als Wert des Tags angeben. Sie können einen beliebigen Tag-Wert angeben, aber der Tag-Schlüssel muss `Patch Group` oder `PatchGroup` lauten.

`PatchGroup`(ohne Leerzeichen) ist erforderlich, wenn Sie Tags in EC2 Instanzmetadaten zugelassen haben.

Nachdem Sie Ihre verwalteten Knoten mithilfe von Tags gruppiert haben, fügen Sie den Patch-Gruppenwert einer Patch-Baseline hinzu. Mit der Registrierung der Patch-Gruppe für eine Patch-Baseline können Sie sicherstellen, dass beim Einspielen von Patches die richtigen Patches installiert werden. Weitere Informationen zu Patch-Gruppen finden Sie unter Patch-Gruppen.

Führen Sie die Aufgaben in diesem Thema aus, um Ihre verwalteten Knoten für das Patching vorzubereiten, indem Sie Tags mit Ihren Knoten und der Patch-Baseline verwenden. Aufgabe 1 ist nur erforderlich, wenn Sie EC2 Amazon-Instances patchen. Aufgabe 2 ist nur erforderlich, wenn Sie EC2 Nicht-Instances in einer Hybrid- und Multi-Cloud-Umgebung patchen. Aufgabe 3 ist für alle verwalteten Knoten erforderlich.

> **Tip**
>
> Sie können verwalteten Knoten auch Tags hinzufügen, indem Sie den AWS CLI Befehl `add-tags-to-resource` oder den Systems Manager API-Vorgang ssm-agent-minimum-s `AddTagsToResource` 3-permissions-required verwenden.

Aufgaben

- Aufgabe 1: Hinzufügen von EC2 Instanzen zu einer Patchgruppe mithilfe von Tags
- Aufgabe 2: Hinzufügen von verwalteten Knoten zu einer Patch-Gruppe mithilfe von Tags
- Aufgabe 3: Hinzufügen einer Patch-Gruppe zu einer Patch-Baseline

Aufgabe 1: Hinzufügen von EC2 Instanzen zu einer Patchgruppe mithilfe von Tags

Sie können EC2 Instances mithilfe der Systems Manager Manager-Konsole oder der EC2 Amazon-Konsole Tags hinzufügen. Diese Aufgabe ist nur erforderlich, wenn Sie EC2 Amazon-Instances patchen.

> **Important**
>
> Sie können das `Patch Group` Tag (mit einem Leerzeichen) nicht auf eine EC2 Amazon-Instance anwenden, wenn die Option Tags in Instance-Metadaten zulassen für die Instance aktiviert ist. Durch das Zulassen von Tags in Instance-Metadaten wird verhindert, dass Tag-Schlüsselnamen Leerzeichen enthalten. Wenn Sie Tags in EC2 Instance-Metadaten zugelassen haben, müssen Sie den Tag-Schlüssel `PatchGroup` (ohne Leerzeichen) verwenden.

Option 1: Hinzufügen von EC2 Instanzen zu einer Patch-Gruppe (Systems Manager Manager-Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie in der Liste Verwaltete Knoten die ID einer verwalteten EC2 Instanz aus, die Sie für das Patchen konfigurieren möchten. Der Knoten IDs für EC2 Instanzen beginnt mit`i-`.

   > **Note**
   >
   > Wenn Sie die EC2 Amazon-Konsole und verwenden AWS CLI, ist es möglich, `Key = PatchGroup` Or-Tags auf Instances anzuwenden`Key = Patch Group`, die noch nicht für die Verwendung mit Systems Manager konfiguriert sind.
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

4. Wählen Sie die Registerkarte Tags und dann Bearbeiten aus.
5. Geben Sie in der linken Spalte **`Patch Group`** oder **`PatchGroup`** ein. Wenn Sie Tags in den EC2 Instance-Metadaten zugelassen haben, müssen Sie die Option `PatchGroup` (ohne Leerzeichen) verwenden.
6. Geben Sie in der rechten Spalte einen Tag-Wert ein, der als Name für die Patch-Gruppe dienen soll.
7. Wählen Sie Speichern.
8. Wiederholen Sie dieses Verfahren, um der gleichen Patchgruppe weitere EC2 Instanzen hinzuzufügen.

Option 2: Hinzufügen von EC2 Instances zu einer Patch-Gruppe (EC2 Amazon-Konsole)

1. Öffnen Sie die EC2 Amazon-Konsole und wählen Sie dann im Navigationsbereich Instances aus.
2. Wählen Sie in der Liste der Instances eine Instance aus, die Sie für das Einspielen von Patches konfigurieren möchten.
3. Wählen Sie im Menü Aktionen die Option Instance-Einstellungen, Tags verwalten aus.

4. Wählen Sie Neues Tag hinzufügen aus.
5. Geben Sie für Key (Schlüssel) **`Patch Group`** oder **`PatchGroup`** ein. Wenn Sie Tags in den EC2 Instance-Metadaten zugelassen haben, müssen Sie die Option `PatchGroup` (ohne Leerzeichen) verwenden.
6. Geben Sie für Wert einen Wert ein, der als Name für die Patch-Gruppe dienen soll.
7. Wählen Sie Speichern.
8. Wiederholen Sie dieses Verfahren, um andere Instances zur selben Patch-Gruppe hinzuzufügen.

Aufgabe 2: Hinzufügen von verwalteten Knoten zu einer Patch-Gruppe mithilfe von Tags

Folgen Sie den Schritten in diesem Thema, um Tags zu AWS IoT Greengrass Kerngeräten und nicht EC2 hybridaktivierten verwalteten Knoten (mi-*) hinzuzufügen. Diese Aufgabe ist nur erforderlich, wenn Sie EC2 Nicht-Instanzen in einer Hybrid- und Multi-Cloud-Umgebung patchen.

> **Note**
>
> Sie können mit der EC2 Amazon-Konsole keine Tags für nicht EC2 verwaltete Knoten hinzufügen.

So fügen Sie einer Patchgruppe nicht EC2 verwaltete Knoten hinzu (Systems Manager Manager-Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie in der Liste der verwalteten Knoten einen verwalteten Knoten, für den Sie das Patching konfigurieren möchten.

   > **Note**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

4. Wählen Sie die Registerkarte Tags und dann Bearbeiten aus.

5. Geben Sie in der linken Spalte **`Patch Group`** oder **`PatchGroup`** ein. Wenn Sie Tags in EC2 Instanz-Metadaten zugelassen haben, müssen Sie `PatchGroup` (ohne Leerzeichen) verwenden.
6. Geben Sie in der rechten Spalte einen Tag-Wert ein, der als Name für die Patch-Gruppe dienen soll.
7. Wählen Sie Speichern.
8. Wiederholen Sie dieses Verfahren, um andere verwaltete Knoten zur selben Patch-Gruppe hinzuzufügen.

Aufgabe 3: Hinzufügen einer Patch-Gruppe zu einer Patch-Baseline

Um Ihren verwalteten Knoten eine bestimmte Patch-Baseline zuzuordnen, müssen Sie den Patch-Gruppenwert der Patch-Baseline hinzufügen. Mit der Registrierung der Patch-Gruppe für eine Patch-Baseline können Sie sicherstellen, dass beim Einspielen von Patches die richtigen Patches installiert werden. Diese Aufgabe ist unabhängig davon erforderlich, ob Sie EC2 Instanzen, nicht EC2 verwaltete Knoten oder beides patchen.

Weitere Informationen zu Patch-Gruppen finden Sie unter Patch-Gruppen.

> **Note**
>
> Welche Schritte Sie ausführen, hängt davon ab, ob Sie vor oder nach der Veröffentlichung der Patch-Richtlinien am 22. Dezember 2022 zum ersten Mal auf Patch Manager zugegriffen haben.

So fügen Sie eine Patch-Gruppe einer Patch-Baseline hinzu (Systems-Manager-Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Patch Manager aus.
3. Wenn Sie auf Patch Manager zum ersten Mal in der aktuellen AWS-Region zugreifen und sich die Patch Manager-Startseite öffnet, wählen Sie Mit einer Übersicht beginnen.
4. Wählen Sie die Registerkarte Patch-Baselines und wählen Sie dann in der Liste Patch-Baselines den Namen der Patch-Baseline, die Sie für Ihre Patch-Gruppe konfigurieren möchten.

Wenn Sie erst nach der Veröffentlichung der Patch-Richtlinien auf Patch Manager zugegriffen haben, müssen Sie eine von Ihnen erstellte benutzerdefinierte Baseline wählen.

5. Wenn die Detailseite der Baseline-ID ein Menü Aktionen enthält, gehen Sie wie folgt vor:

   - Wählen Sie Actions (Aktionen) und dann Modify patch groups (Patch-Gruppen modifizieren) aus.
   - Geben Sie den Tag-Wert, den Sie Ihren verwalteten Knoten in Aufgabe 2: Hinzufügen von verwalteten Knoten zu einer Patch-Gruppe mithilfe von Tags hinzugefügt haben, und wählen Sie dann Hinzufügen.

   Wenn die Detailseite der Baseline-ID kein Menü Aktionen enthält, können Patch-Gruppen in der Konsole nicht konfiguriert werden. Sie können stattdessen eine der folgenden Aktionen ausführen:

   - (Empfohlen) Richten Sie eine Patch-Richtlinie inQuick Setup, ein Tool in AWS Systems Manager, ein, um eine Patch-Baseline einer oder mehreren EC2 Instanzen zuzuordnen.

     Weitere Informationen finden Sie unter Verwenden von Quick Setup-Patch-Richtlinien und Automatisieren des unternehmensweiten Patchen mithilfe einer Quick Setup-Patch-Richtlinie.
   - Verwenden Sie den register-patch-baseline-for-patch-groupBefehl in AWS Command Line Interface (AWS CLI), um eine Patchgruppe zu konfigurieren.

Integration von Patch Manager in AWS Security Hub CSPM

AWS Security Hub CSPMbietet Ihnen einen umfassenden Überblick über Ihren Sicherheitsstatus in. AWS Security Hub CSPM sammelt Sicherheitsdaten von allen AWS-Konten und unterstützten Produkten von Drittanbietern. AWS-Services Mit Security Hub CSPM können Sie Ihre Umgebung anhand von Branchenstandards und Best Practices überprüfen. Security Hub CSPM hilft Ihnen dabei, Ihre Sicherheitstrends zu analysieren und Sicherheitsprobleme mit der höchsten Priorität zu identifizieren.

Mithilfe der Integration zwischenPatch Manager, einem Tool in AWS Systems Manager und Security Hub CSPM können Sie Erkenntnisse über nicht konforme Knoten von Patch Manager an Security Hub CSPM senden. Ein Ergebnis ist der beobachtbare Datensatz einer Sicherheitsprüfung oder sicherheitsrelevanten Erkennung. Security Hub CSPM kann diese patch-bezogenen Ergebnisse dann in die Analyse Ihres Sicherheitsstatus einbeziehen.

Die Informationen in den folgenden Themen gelten unabhängig davon, welche Methode oder Art der Konfiguration Sie für Ihre Patching-Vorgänge verwenden:

- Eine in Quick Setup konfigurierte Patch-Richtlinie
- Eine in Quick Setup konfigurierte Host-Management-Option
- Ein Wartungsfenster zum Ausführen eines Patch-`Scan` oder einer `Install`-Aufgabe
- Ein On-Demand-Jetzt patchen-Vorgang

Inhalt

- Wie Patch Manager sendet man Ergebnisse an Security Hub CSPM
  - Arten von Erkenntnissen, die Patch Manager sendet
  - Latenz für das Senden von Erkenntnissen
  - Wiederholter Versuch, wenn Security Hub CSPM nicht verfügbar ist
  - Ergebnisse in Security Hub CSPM anzeigen
- Typische Erkenntnis von Patch Manager
- Aktivieren und Konfigurieren der Integration
- So beenden Sie das Senden von Ergebnissen

Wie Patch Manager sendet man Ergebnisse an Security Hub CSPM

In Security Hub CSPM werden Sicherheitsprobleme als Ergebnisse verfolgt. Einige Ergebnisse stammen aus Problemen, die von anderen AWS-Services oder von Drittanbietern entdeckt wurden. Security Hub CSPM verfügt außerdem über eine Reihe von Regeln, anhand derer Sicherheitsprobleme erkannt und Ergebnisse generiert werden.

Patch Managerist eines der Systems Manager Manager-Tools, das Ergebnisse an Security Hub CSPM sendet. Nachdem Sie einen Patchvorgang durchgeführt haben, indem Sie ein SSM-Dokument (`AWS-RunPatchBaseline`,, oder`AWS-RunPatchBaselineWithHooks`) ausführen`AWS-RunPatchBaselineAssociation`, werden die Patching-Informationen an Inventory oder Compliance, Tools in oder an beide gesendet. AWS Systems Manager Nachdem Inventory, Compliance oder beide die Daten erhalten haben, erhält Patch Manager eine Benachrichtigung. Dann wertet Patch Manager die Daten auf Genauigkeit, Formatierung und Compliance aus. Wenn alle Bedingungen erfüllt sind, werden die Daten an Security Hub CSPM Patch Manager weitergeleitet.

Security Hub CSPM bietet Tools zur Verwaltung von Ergebnissen aus all diesen Quellen. Sie können Listen mit Erkenntnissen anzeigen und filtern und Details zu einer Erkenntnis anzeigen. Weitere Informationen finden Sie unter Anzeigen der Erkenntnisse im AWS Security Hub -Benutzerhandbuch. Sie können auch den Status einer Untersuchung zu einer Erkenntnis nachverfolgen. Weitere Informationen finden Sie unter Ergreifen von Maßnahmen zu Erkenntnissen im AWS Security Hub -Benutzerhandbuch.

Alle Ergebnisse in Security Hub CSPM verwenden ein standardmäßiges JSON-Format, das AWS Security Finding Format (ASFF). Das ASFF enthält Details über die Ursache des Problems, die betroffenen Ressourcen und den aktuellen Status der Erkenntnis. Weitere Informationen finden Sie unter AWS -Security Finding-Format (ASFF) im AWS Security Hub -Benutzerhandbuch.

Arten von Erkenntnissen, die Patch Manager sendet

Patch Managersendet die Ergebnisse mithilfe des Security Finding Formats (ASFF) an AWS Security Hub CSPM. In ASFF gibt das `Types`-Feld die Art der Erkenntnis an. Die Ergebnisse von Patch Manager können den folgenden Wert für `Types` haben:

- Software- und Konfigurationsmanagement Checks/Patch

Patch Manager sendet ein Ergebnis pro nicht konformen verwalteten Knoten. Das Ergebnis wird mit dem Ressourcentyp gemeldet, `AwsEc2Instance`sodass die Ergebnisse mit anderen Security Hub CSPM-Integrationen, die Ressourcentypen melden, korreliert werden können. `AwsEc2Instance` Patch Managerleitet ein Ergebnis nur dann an Security Hub CSPM weiter, wenn bei dem Vorgang festgestellt wurde, dass der verwaltete Knoten nicht konform ist. Das Ergebnis enthält die Ergebnisse der Patch-Zusammenfassung.

> **Note**
>
> Nach der Meldung eines nicht konformen Knotens an Security Hub CSPM. Patch Managersendet kein Update an Security Hub CSPM, nachdem der Knoten konform gemacht wurde. Sie können die Ergebnisse in Security Hub CSPM manuell beheben, nachdem die erforderlichen Patches auf den verwalteten Knoten angewendet wurden.

Weitere Informationen zu Compliance-Definitionen finden Sie unter Statuswerte der Patch-Compliance. Weitere Informationen zu `PatchSummary` finden Sie PatchSummaryin der AWS Security Hub API-Referenz.

Latenz für das Senden von Erkenntnissen

Wenn ein neues Ergebnis Patch Manager erstellt wird, wird es normalerweise innerhalb weniger Sekunden bis 2 Stunden an Security Hub CSPM gesendet. Die Geschwindigkeit hängt vom Verkehr zu diesem Zeitpunkt in der AWS-Region verarbeiteten Verkehr ab.

Wiederholter Versuch, wenn Security Hub CSPM nicht verfügbar ist

Bei einem Dienstausfall wird eine AWS Lambda Funktion ausgeführt, mit der die Nachrichten wieder in die Hauptwarteschlange verschoben werden, nachdem der Dienst wieder ausgeführt wurde. Nachdem sich die Nachrichten in der Hauptwarteschlange befinden, erfolgt die Wiederholung automatisch.

Wenn Security Hub CSPM nicht verfügbar ist, Patch Manager versucht es erneut, die Ergebnisse zu senden, bis sie empfangen werden.

Ergebnisse in Security Hub CSPM anzeigen

In diesem Verfahren wird beschrieben, wie Sie in Security Hub CSPM Ergebnisse zu verwalteten Knoten in Ihrer Flotte einsehen können, die nicht mehr patch-konform sind.

Um die CSPM-Ergebnisse von Security Hub auf Patch-Konformität zu überprüfen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS Security Hub CSPM Konsole unter. https://console.aws.amazon.com/securityhub/
2. Wählen Sie im Navigationsbereich Findings aus.
3. Wählen Sie das Feld Filter hinzufügen ( ).
4. Wählen Sie im Menü unter Filter die Option Produktname aus.
5. Wählen Sie in dem sich öffnenden Dialogfeld im ersten Feld die Option ist und geben Sie dann **`Systems Manager Patch Manager`** im zweiten Feld ein.
6. Wählen Sie Anwenden aus.
7. Fügen Sie weitere Filter hinzu, um Ihre Ergebnisse einzugrenzen.
8. Wählen Sie in der Ergebnisliste den Titel eines Erkenntnisses aus, zu dem Sie weitere Informationen wünschen.

   Auf der rechten Seite des Bildschirms wird ein Bereich mit weiteren Informationen zur Ressource, dem erkannten Problem und einer empfohlenen Lösung geöffnet.

> **Important**
>
> Derzeit meldet Security Hub CSPM den Ressourcentyp aller verwalteten Knoten als. `EC2 Instance` Dazu gehören lokale Server und virtuelle Maschinen (VMs), die Sie für die Verwendung mit Systems Manager registriert haben.

Schweregradklassifizierungen

Die Liste der Erkenntnisse für **`Systems Manager Patch Manager`** enthält einen Bericht über den Schweregrad des Befundes. Zu den Schweregraden gehören die folgenden, vom niedrigsten zum höchsten:

- INFORMATIV – Es wurde kein Problem gefunden.
- NIEDRIG — Das Problem muss nicht behoben werden.
- MITTEL – Das Problem muss angegangen werden, aber ist nicht dringend.
- HOCH – Das Problem muss vorrangig behandelt werden.
- KRITISCH – Das Problem muss sofort behoben werden, um eine Eskalation zu vermeiden.

Der Schweregrad wird durch das schwerwiegendste nicht konforme Paket auf einer Instance bestimmt. Da Sie mehrere Patch-Baselines mit verschiedenen Schweregraden haben können, wird der höchste Schweregrad von allen nicht konformen Paketen gemeldet. Nehmen wir zum Beispiel an, Sie haben zwei nicht konforme Pakete, wobei der Schweregrad von Paket A „Kritisch“ und der von Paket B „Gering“ ist. „Kritisch“ wird als Schweregrad angegeben werden.

Beachten Sie, dass das Schweregradfeld direkt mit dem Feld Patch Manager `Compliance` korreliert. Dies ist ein Feld, das Sie einzelnen Patches zuweisen, die der Regel entsprechen. Da dieses `Compliance`-Feld einzelnen Patches zugewiesen ist, wird es nicht auf der Ebene der Patch-Zusammenfassung wiedergegeben.

Verwandter Inhalt

- Erkenntnisse im AWS Security Hub -Benutzerhandbuch
- Multi-Account Patch-Compliance mit Patch Manager und Security Hub im AWS -Management & Governance Blog

Typische Erkenntnis von Patch Manager

Patch Managersendet Ergebnisse mithilfe des Security Finding Formats (ASFF) an AWS Security Hub CSPM.

Hier ist ein Beispiel für ein typisches Ergebnis von Patch Manager.


```
{
  "SchemaVersion": "2018-10-08",
  "Id": "arn:aws:patchmanager:us-east-2:111122223333:instance/i-02573cafcfEXAMPLE/
document/AWS-RunPatchBaseline/run-command/d710f5bd-04e3-47b4-82f6-df4e0EXAMPLE",
  "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/ssm-patch-manager",
  "GeneratorId": "d710f5bd-04e3-47b4-82f6-df4e0EXAMPLE",
  "AwsAccountId": "111122223333",
  "Types": [
    "Software & Configuration Checks/Patch Management/Compliance"
  ],
  "CreatedAt": "2021-11-11T22:05:25Z",
  "UpdatedAt": "2021-11-11T22:05:25Z",
  "Severity": {
    "Label": "INFORMATIONAL",
    "Normalized": 0
  },
  "Title": "Systems Manager Patch Summary - Managed Instance Non-Compliant",
  "Description": "This AWS control checks whether each instance that is managed by AWS
 Systems Manager is in compliance with the rules of the patch baseline that applies to
 that instance when a compliance Scan runs.",
  "Remediation": {
    "Recommendation": {
      "Text": "For information about bringing instances into patch compliance, see
 'Remediating out-of-compliance instances (Patch Manager)'.",
      "Url": "https://docs.aws.amazon.com/systems-manager/latest/userguide/patch-
compliance-remediation.html"
    }
  },
  "SourceUrl": "https://us-east-2.console.aws.amazon.com/systems-manager/fleet-manager/
i-02573cafcfEXAMPLE/patch?region=us-east-2",
  "ProductFields": {
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-2::product/aws/ssm-
patch-manager/arn:aws:patchmanager:us-east-2:111122223333:instance/i-02573cafcfEXAMPLE/
document/AWS-RunPatchBaseline/run-command/d710f5bd-04e3-47b4-82f6-df4e0EXAMPLE",
    "aws/securityhub/ProductName": "Systems Manager Patch Manager",
    "aws/securityhub/CompanyName": "AWS"
  },
```

```
  "Resources": [
    {
      "Type": "AwsEc2Instance",
      "Id": "i-02573cafcfEXAMPLE",
      "Partition": "aws",
      "Region": "us-east-2"
    }
  ],
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ACTIVE",
  "PatchSummary": {
    "Id": "pb-0c10e65780EXAMPLE",
    "InstalledCount": 45,
    "MissingCount": 2,
    "FailedCount": 0,
    "InstalledOtherCount": 396,
    "InstalledRejectedCount": 0,
    "InstalledPendingReboot": 0,
    "OperationStartTime": "2021-11-11T22:05:06Z",
    "OperationEndTime": "2021-11-11T22:05:25Z",
    "RebootOption": "NoReboot",
    "Operation": "SCAN"
  }
}
```


Aktivieren und Konfigurieren der Integration

Um die Patch Manager Integration mit Security Hub CSPM zu verwenden, müssen Sie Security Hub CSPM aktivieren. Informationen zum Aktivieren von Security Hub CSPM finden Sie unter Security Hub CSPM einrichten im Benutzerhandbuch.AWS Security Hub

Das folgende Verfahren beschreibt, wie Security Hub CSPM integriert Patch Manager wird, wenn Security Hub CSPM bereits aktiv, aber die Patch Manager Integration ausgeschaltet ist. Sie müssen diesen Vorgang nur abschließen, wenn die Integration manuell deaktiviert wurde.

Um die CSPM-Integration Patch Manager zur Security Hub hinzuzufügen

1. Wählen Sie im Navigationsbereich Patch Manager aus.
2. Wählen Sie die Registerkarte Einstellungen.

–oder–

Wenn Sie in der aktuellen AWS-Region zum ersten Mal auf Patch Manager zugreifen, wählen Sie Mit einer Übersicht beginnen und dann die Registerkarte Einstellungen aus.

3. Wählen Sie im Abschnitt Export to Security Hub CSPM rechts neben Die Ergebnisse der Patch-Konformität werden nicht nach Security Hub exportiert die Option Aktivieren aus.

So beenden Sie das Senden von Ergebnissen

Um das Senden von Ergebnissen an Security Hub CSPM zu beenden, können Sie entweder die Security Hub CSPM-Konsole oder die API verwenden.

Weitere Informationen finden Sie in folgenden Themen im AWS Security Hub -Benutzerhandbuch:

- Deaktivieren und Aktivieren des Flows von Ergebnissen aus einer Integration (Konsole)
- Deaktivierung des Flusses von Erkenntnissen aus einer Integration (Security Hub CSPM API,) AWS CLI

Arbeiten mit Patch Manager-Ressourcen unter Verwendung der AWS CLI

Der Abschnitt enthält Beispiele für AWS Command Line Interface (AWS CLI) -Befehle, mit denen Sie Konfigurationsaufgaben für Patch Manager ein Tool in ausführen können AWS Systems Manager.

Eine Veranschaulichung der Verwendung von AWS CLI zum Patchen einer Serverumgebung mithilfe einer benutzerdefinierten Patch-Baseline finden Sie unterTutorial: Patchen Sie eine Serverumgebung mit dem AWS CLI.

Weitere Informationen zur Verwendung von AWS CLI for AWS Systems Manager Tasks finden Sie im AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Themen

- AWS CLI Befehle für Patch-Baselines
- AWS CLI Befehle für Patchgruppen
- AWS CLI Befehle zum Anzeigen von Patch-Zusammenfassungen und Details
- AWS CLI Befehle zum Scannen und Patchen verwalteter Knoten

AWS CLI Befehle für Patch-Baselines

Beispielbefehle für Patch-Baselines

- Erstellen einer Patch-Baseline
- Erstellen einer Patch-Baseline mit benutzerdefinierten Repositorys für verschiedene Betriebssystemversionen
- Aktualisieren einer Patch-Baseline
- Umbenennen einer Patch-Baseline
- Löschen einer Patch-Baseline
- Auflisten aller Patch-Baselines
- Listet alle von AWS-bereitgestellten Patch-Baselines auf
- Auflisten der eigenen Patch-Baselines
- Anzeigen einer Patch-Baseline
- Abrufen einer Standard-Patch-Baseline
- Eine benutzerdefinierte Patch-Baseline als Standard festlegen
- Setzen Sie eine AWS Patch-Baseline als Standard zurück
- Markieren einer Patch-Baseline
- Auflisten aller Tags für eine Patch-Baseline
- Entfernen eines Tags aus einer Patch-Baseline

Erstellen einer Patch-Baseline

Der folgende Befehl erstellt eine Patch-Baseline, die alle kritischen und wichtigen Sicherheitsaktualisierungen für Windows Server 2012 R2 fünf Tage nach ihrer Veröffentlichung genehmigt. Patches wurden auch für die Listen „Genehmigt“ und „Zurückgewiesen“ angegeben. Darüber hinaus wurde die Patch-Baseline mit Tags versehen, um sie für die Produktionsumgebung freizugeben.

Linux & macOS

```
aws ssm create-patch-baseline \
    --name "Windows-Server-2012R2" \
    --tags "Key=Environment,Value=Production" \
    --description "Windows Server 2012 R2, Important and Critical security updates"
 \
```

```
    --approved-patches "KB2032276,MS10-048" \
    --rejected-patches "KB2124261" \
    --rejected-patches-action "ALLOW_AS_DEPENDENCY" \
    --approval-rules
 "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Important,Critical
{Key=CLASSIFICATION,Values=SecurityUpdates},
{Key=PRODUCT,Values=WindowsServer2012R2}]},ApproveAfterDays=5}]"
```

Windows Server

```
aws ssm create-patch-baseline ^
    --name "Windows-Server-2012R2" ^
    --tags "Key=Environment,Value=Production" ^
    --description "Windows Server 2012 R2, Important and Critical security updates"
 ^
    --approved-patches "KB2032276,MS10-048" ^
    --rejected-patches "KB2124261" ^
    --rejected-patches-action "ALLOW_AS_DEPENDENCY" ^
    --approval-rules
 "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Important,Critical
{Key=CLASSIFICATION,Values=SecurityUpdates},
{Key=PRODUCT,Values=WindowsServer2012R2}]},ApproveAfterDays=5}]"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

Erstellen einer Patch-Baseline mit benutzerdefinierten Repositorys für verschiedene Betriebssystemversionen

Gilt nur für Linux-verwaltete Knoten. Der folgende Befehl zeigt, wie das Patch-Repository für eine bestimmte Version des Amazon Linux-Betriebssystems anzugeben ist. Dieses Beispiel verwendet ein standardmäßig aktiviertes Quell-Repository auf Amazon Linux 2017.09, könnte aber auf ein anderes Quell-Repository angepasst werden, das Sie für einen verwalteten Knoten konfiguriert haben.

> **Note**
>
> Um diesen komplexeren Befehl besser zu erklären, wird die Option `--cli-input-json` mit zusätzlichen, in einer externen JSON-Datei gespeicherten Optionen verwendet.

1. Erstellen Sie eine JSON-Datei mit einem Namen wie `my-patch-repository.json` und fügen Sie den folgenden Inhalt hinzu.

```
{
    "Description": "My patch repository for Amazon Linux 2",
    "Name": "Amazon-Linux-2",
    "OperatingSystem": "AMAZON_LINUX_2",
    "ApprovalRules": {
        "PatchRules": [
            {
                "ApproveAfterDays": 7,
                "EnableNonSecurity": true,
                "PatchFilterGroup": {
                    "PatchFilters": [
                        {
                            "Key": "SEVERITY",
                            "Values": [
                                "Important",
                                "Critical"
                            ]
                        },
                        {
                            "Key": "CLASSIFICATION",
                            "Values": [
                                "Security",
                                "Bugfix"
                            ]
                        },
                        {
                            "Key": "PRODUCT",
                            "Values": [
                                "AmazonLinux2"
                            ]
                        }
                    ]
                }
```

```
            }
        ]
    },
    "Sources": [
        {
            "Name": "My-AL2",
            "Products": [
                "AmazonLinux2"
            ],
            "Configuration": "[amzn-main] \nname=amzn-main-Base
\nmirrorlist=http://repo./$awsregion./$awsdomain//$releasever/main/
mirror.list //nmirrorlist_expire=300//nmetadata_expire=300 \npriority=10
 \nfailovermethod=priority \nfastestmirror_enabled=0 \ngpgcheck=1
 \ngpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-amazon-ga \nenabled=1 \nretries=3
 \ntimeout=5\nreport_instanceid=yes"
        }
    ]
}
```

2. Führen Sie im Verzeichnis, in dem Sie die Datei gespeichert haben, den folgenden Befehl aus.

```
aws ssm create-patch-baseline --cli-input-json file://my-patch-repository.json
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "BaselineId": "pb-0c10e65780EXAMPLE"
}
```

Aktualisieren einer Patch-Baseline

Mit dem folgenden Befehl werden zwei Patches abgelehnt und ein weiterer Patch für eine vorhandenen Patch-Baseline genehmigt.

Weitere Informationen zu akzeptierten Formaten für Listen genehmigter und abgelehnter Patches finden Sie unter Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen.

Linux & macOS

```
aws ssm update-patch-baseline \
    --baseline-id pb-0c10e65780EXAMPLE \
```

```
    --rejected-patches "KB2032276" "MS10-048" \
    --approved-patches "KB2124261"
```

Windows Server

```
aws ssm update-patch-baseline ^
    --baseline-id pb-0c10e65780EXAMPLE ^
    --rejected-patches "KB2032276" "MS10-048" ^
    --approved-patches "KB2124261"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE",
   "Name":"Windows-Server-2012R2",
   "RejectedPatches":[
      "KB2032276",
      "MS10-048"
   ],
   "GlobalFilters":{
      "PatchFilters":[

      ]
   },
   "ApprovalRules":{
      "PatchRules":[
         {
            "PatchFilterGroup":{
               "PatchFilters":[
                  {
                     "Values":[
                        "Important",
                        "Critical"
                     ],
                     "Key":"MSRC_SEVERITY"
                  },
                  {
                     "Values":[
                        "SecurityUpdates"
                     ],
                     "Key":"CLASSIFICATION"
                  },
```

```
                {
                   "Values":[
                      "WindowsServer2012R2"
                   ],
                   "Key":"PRODUCT"
                }
             ]
          },
          "ApproveAfterDays":5
       }
    ]
 },
 "ModifiedDate":1481001494.035,
 "CreatedDate":1480997823.81,
 "ApprovedPatches":[
    "KB2124261"
 ],
 "Description":"Windows Server 2012 R2, Important and Critical security updates"
}
```

Umbenennen einer Patch-Baseline

Linux & macOS

```
aws ssm update-patch-baseline \
    --baseline-id pb-0c10e65780EXAMPLE \
    --name "Windows-Server-2012-R2-Important-and-Critical-Security-Updates"
```

Windows Server

```
aws ssm update-patch-baseline ^
    --baseline-id pb-0c10e65780EXAMPLE ^
    --name "Windows-Server-2012-R2-Important-and-Critical-Security-Updates"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE",
   "Name":"Windows-Server-2012-R2-Important-and-Critical-Security-Updates",
   "RejectedPatches":[
      "KB2032276",
      "MS10-048"
```

```
   ],
   "GlobalFilters":{
      "PatchFilters":[

      ]
   },
   "ApprovalRules":{
      "PatchRules":[
         {
            "PatchFilterGroup":{
               "PatchFilters":[
                  {
                     "Values":[
                        "Important",
                        "Critical"
                     ],
                     "Key":"MSRC_SEVERITY"
                  },
                  {
                     "Values":[
                        "SecurityUpdates"
                     ],
                     "Key":"CLASSIFICATION"
                  },
                  {
                     "Values":[
                        "WindowsServer2012R2"
                     ],
                     "Key":"PRODUCT"
                  }
               ]
            },
            "ApproveAfterDays":5
         }
      ]
   },
   "ModifiedDate":1481001795.287,
   "CreatedDate":1480997823.81,
   "ApprovedPatches":[
      "KB2124261"
   ],
   "Description":"Windows Server 2012 R2, Important and Critical security updates"
}
```

Löschen einer Patch-Baseline

```
aws ssm delete-patch-baseline --baseline-id "pb-0c10e65780EXAMPLE"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

Auflisten aller Patch-Baselines

```
aws ssm describe-patch-baselines
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineIdentities":[
      {
         "BaselineName":"AWS-DefaultPatchBaseline",
         "DefaultBaseline":true,
         "BaselineDescription":"Default Patch Baseline Provided by AWS.",
         "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE"
      },
      {
         "BaselineName":"Windows-Server-2012R2",
         "DefaultBaseline":false,
         "BaselineDescription":"Windows Server 2012 R2, Important and Critical security
 updates",
         "BaselineId":"pb-0c10e65780EXAMPLE"
      }
   ]
}
```

Nachstehend finden Sie einen weiteren Befehl zur Auflistung aller Patch-Baselines in einer AWS-Region.

Linux & macOS

```
aws ssm describe-patch-baselines \
```

```
    --region us-east-2 \
    --filters "Key=OWNER,Values=[All]"
```

Windows Server

```
aws ssm describe-patch-baselines ^
    --region us-east-2 ^
    --filters "Key=OWNER,Values=[All]"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineIdentities":[
      {
         "BaselineName":"AWS-DefaultPatchBaseline",
         "DefaultBaseline":true,
         "BaselineDescription":"Default Patch Baseline Provided by AWS.",
         "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE"
      },
      {
         "BaselineName":"Windows-Server-2012R2",
         "DefaultBaseline":false,
         "BaselineDescription":"Windows Server 2012 R2, Important and Critical security
 updates",
         "BaselineId":"pb-0c10e65780EXAMPLE"
      }
   ]
}
```

Listet alle von AWS-bereitgestellten Patch-Baselines auf

Linux & macOS

```
aws ssm describe-patch-baselines \
    --region us-east-2 \
    --filters "Key=OWNER,Values=[AWS]"
```

Windows Server

```
aws ssm describe-patch-baselines ^
```

```
    --region us-east-2 ^
    --filters "Key=OWNER,Values=[AWS]"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineIdentities":[
      {
         "BaselineName":"AWS-DefaultPatchBaseline",
         "DefaultBaseline":true,
         "BaselineDescription":"Default Patch Baseline Provided by AWS.",
         "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE"
      }
   ]
}
```

Auflisten der eigenen Patch-Baselines

Linux & macOS

```
aws ssm describe-patch-baselines \
    --region us-east-2 \
    --filters "Key=OWNER,Values=[Self]"
```

Windows Server

```
aws ssm describe-patch-baselines ^
    --region us-east-2 ^
    --filters "Key=OWNER,Values=[Self]"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineIdentities":[
      {
         "BaselineName":"Windows-Server-2012R2",
         "DefaultBaseline":false,
         "BaselineDescription":"Windows Server 2012 R2, Important and Critical security
 updates",
```

```
         "BaselineId":"pb-0c10e65780EXAMPLE"
      }
   ]
}
```

Anzeigen einer Patch-Baseline

```
aws ssm get-patch-baseline --baseline-id pb-0c10e65780EXAMPLE
```

> **Note**
>
> Bei benutzerdefinierten Patch-Baselines können Sie entweder die Patch-Baseline-ID oder den vollständigen Amazon-Ressourcennamen (ARN) angeben. Für eine von AWS-bereitgestellte Patch-Baseline müssen Sie den vollständigen ARN angeben. Beispiel, `arn:aws:ssm:us-east-2:075727635805:patchbaseline/pb-0c10e65780EXAMPLE`.

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE",
   "Name":"Windows-Server-2012R2",
   "PatchGroups":[
      "Web Servers"
   ],
   "RejectedPatches":[

   ],
   "GlobalFilters":{
      "PatchFilters":[

      ]
   },
   "ApprovalRules":{
      "PatchRules":[
         {
            "PatchFilterGroup":{
               "PatchFilters":[
                  {
                     "Values":[
```

```
                        "Important",
                        "Critical"
                     ],
                     "Key":"MSRC_SEVERITY"
                  },
                  {
                     "Values":[
                        "SecurityUpdates"
                     ],
                     "Key":"CLASSIFICATION"
                  },
                  {
                     "Values":[
                        "WindowsServer2012R2"
                     ],
                     "Key":"PRODUCT"
                  }
               ]
            },
            "ApproveAfterDays":5
         }
      ]
   },
   "ModifiedDate":1480997823.81,
   "CreatedDate":1480997823.81,
   "ApprovedPatches":[

   ],
   "Description":"Windows Server 2012 R2, Important and Critical security updates"
}
```

Abrufen einer Standard-Patch-Baseline

```
aws ssm get-default-patch-baseline --region us-east-2
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/pb-0c10e65780EXAMPLE"
}
```

Eine benutzerdefinierte Patch-Baseline als Standard festlegen

Linux & macOS

```
aws ssm register-default-patch-baseline \
    --region us-east-2 \
    --baseline-id "pb-0c10e65780EXAMPLE"
```

Windows Server

```
aws ssm register-default-patch-baseline ^
    --region us-east-2 ^
    --baseline-id "pb-0c10e65780EXAMPLE"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

Setzen Sie eine AWS Patch-Baseline als Standard zurück

Linux & macOS

```
aws ssm register-default-patch-baseline \
    --region us-east-2 \
    --baseline-id "arn:aws:ssm:us-east-2:123456789012:patchbaseline/
pb-0c10e65780EXAMPLE"
```

Windows Server

```
aws ssm register-default-patch-baseline ^
    --region us-east-2 ^
    --baseline-id "arn:aws:ssm:us-east-2:123456789012:patchbaseline/
pb-0c10e65780EXAMPLE"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
```

```
    "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

Markieren einer Patch-Baseline

Linux & macOS

```
aws ssm add-tags-to-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0c10e65780EXAMPLE" \
    --tags "Key=Project,Value=Testing"
```

Windows Server

```
aws ssm add-tags-to-resource ^
    --resource-type "PatchBaseline" ^
    --resource-id "pb-0c10e65780EXAMPLE" ^
    --tags "Key=Project,Value=Testing"
```

Auflisten aller Tags für eine Patch-Baseline

Linux & macOS

```
aws ssm list-tags-for-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0c10e65780EXAMPLE"
```

Windows Server

```
aws ssm list-tags-for-resource ^
    --resource-type "PatchBaseline" ^
    --resource-id "pb-0c10e65780EXAMPLE"
```

Entfernen eines Tags aus einer Patch-Baseline

Linux & macOS

```
aws ssm remove-tags-from-resource \
    --resource-type "PatchBaseline" \
```

```
    --resource-id "pb-0c10e65780EXAMPLE" \
    --tag-keys "Project"
```

Windows Server

```
aws ssm remove-tags-from-resource ^
    --resource-type "PatchBaseline" ^
    --resource-id "pb-0c10e65780EXAMPLE" ^
    --tag-keys "Project"
```

AWS CLI Befehle für Patchgruppen

Beispielbefehle für Patch-Gruppen

- Erstellen einer Patch-Gruppe
- Registrieren einer Patch-Gruppe „Webserver“ für eine Patch-Baseline
- Registrieren Sie eine Patch-Gruppe „Backend“ mit der AWS bereitgestellten Patch-Baseline
- Anzeigen der Registrierungen für Patch-Gruppen
- Aufheben der Registrierung einer Patch-Gruppe für eine Patch-Baseline

Erstellen einer Patch-Gruppe

> **Note**
>
> Patch-Gruppen werden nicht in Patch-Vorgängen verwendet, die auf Patch-Richtlinien basieren. Weitere Informationen zur Arbeit mit Patch-Richtlinien finden Sie unter Patch-Richtlinienkonfigurationen in Quick Setup.

Um das Organisieren Ihrer Patching-Aufgaben zu erleichtern, empfehlen wir, dass Sie verwaltete Knoten mithilfe von Tags zu Patch-Gruppen hinzufügen. Patch-Gruppen erfordern die Nutzung des Tag-Schlüssels `Patch Group` oder `PatchGroup`. Wenn Sie Tags in EC2 Instanz-Metadaten zugelassen haben, müssen Sie `PatchGroup` (ohne Leerzeichen) verwenden. Sie können einen beliebigen Tag-Wert angeben, aber der Tag-Schlüssel muss `Patch Group` oder `PatchGroup` lauten. Weitere Informationen zu Patch-Gruppen finden Sie unter Patch-Gruppen.

Nachdem Sie Ihre verwalteten Knoten mithilfe von Tags gruppiert haben, fügen Sie den Patch-Gruppenwert einer Patch-Baseline hinzu. Mit der Registrierung der Patch-Gruppe für eine Patch-

Baseline können Sie sicherstellen, dass beim Einspielen von Patches die richtigen Patches installiert werden.

Aufgabe 1: Fügen Sie EC2 Instanzen mithilfe von Tags zu einer Patchgruppe hinzu

> **Note**
>
> Wenn Sie die Amazon Elastic Compute Cloud (Amazon EC2) -Konsole und verwenden AWS CLI, ist es möglich, `Key = PatchGroup` Or-Tags auf Instances anzuwenden`Key = PatchGroup`, die noch nicht für die Verwendung mit Systems Manager konfiguriert sind. Wenn eine EC2 Instance, die Sie voraussichtlich sehen werden, nach dem Anwenden des `Key = PatchGroup` Tags `Patch Group` oder Patch Manager nicht aufgeführt ist, finden Sie unter Tipps Problembehandlung bei der Verfügbarkeit verwalteter Knoten zur Fehlerbehebung.

Führen Sie den folgenden Befehl aus, um das `PatchGroup` Tag einer EC2 Instanz hinzuzufügen.

```
aws ec2 create-tags --resources "i-1234567890abcdef0" --tags
 "Key=PatchGroup,Value=GroupValue"
```

Aufgabe 2: Hinzufügen von verwalteten Knoten zu einer Patch-Gruppe mithilfe von Tags

Führen Sie den folgenden Befehl aus, um das Tag `PatchGroup` einem verwalteten Knoten hinzuzufügen.

Linux & macOS

```
aws ssm add-tags-to-resource \
    --resource-type "ManagedInstance" \
    --resource-id "mi-0123456789abcdefg" \
    --tags "Key=PatchGroup,Value=GroupValue"
```

Windows Server

```
aws ssm add-tags-to-resource ^
    --resource-type "ManagedInstance" ^
    --resource-id "mi-0123456789abcdefg" ^
    --tags "Key=PatchGroup,Value=GroupValue"
```

Aufgabe 3: Hinzufügen einer Patch-Gruppe zu einer Patch-Baseline

Führen Sie den folgenden Befehl aus, um der angegebenen Patch-Baseline einen `PatchGroup`-Tag-Wert zuzuordnen.

Linux & macOS

```
aws ssm register-patch-baseline-for-patch-group \
    --baseline-id "pb-0c10e65780EXAMPLE" \
    --patch-group "Development"
```

Windows Server

```
aws ssm register-patch-baseline-for-patch-group ^
    --baseline-id "pb-0c10e65780EXAMPLE" ^
    --patch-group "Development"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
  "PatchGroup": "Development",
  "BaselineId": "pb-0c10e65780EXAMPLE"
}
```

Registrieren einer Patch-Gruppe „Webserver“ für eine Patch-Baseline

Linux & macOS

```
aws ssm register-patch-baseline-for-patch-group \
    --baseline-id "pb-0c10e65780EXAMPLE" \
    --patch-group "Web Servers"
```

Windows Server

```
aws ssm register-patch-baseline-for-patch-group ^
    --baseline-id "pb-0c10e65780EXAMPLE" ^
    --patch-group "Web Servers"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "PatchGroup":"Web Servers",
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

Registrieren Sie eine Patch-Gruppe „Backend“ mit der AWS bereitgestellten Patch-Baseline

Linux & macOS

```
aws ssm register-patch-baseline-for-patch-group \
    --region us-east-2 \
    --baseline-id "arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE" \
    --patch-group "Backend"
```

Windows Server

```
aws ssm register-patch-baseline-for-patch-group ^
    --region us-east-2 ^
    --baseline-id "arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE" ^
    --patch-group "Backend"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "PatchGroup":"Backend",
   "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/pb-0c10e65780EXAMPLE"
}
```

Anzeigen der Registrierungen für Patch-Gruppen

```
aws ssm describe-patch-groups --region us-east-2
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "PatchGroupPatchBaselineMappings":[
      {
         "PatchGroup":"Backend",
```

```
         "BaselineIdentity":{
            "BaselineName":"AWS-DefaultPatchBaseline",
            "DefaultBaseline":false,
            "BaselineDescription":"Default Patch Baseline Provided by AWS.",
            "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE"
         }
      },
      {
         "PatchGroup":"Web Servers",
         "BaselineIdentity":{
            "BaselineName":"Windows-Server-2012R2",
            "DefaultBaseline":true,
            "BaselineDescription":"Windows Server 2012 R2, Important and Critical
 updates",
            "BaselineId":"pb-0c10e65780EXAMPLE"
         }
      }
   ]
}
```

Aufheben der Registrierung einer Patch-Gruppe für eine Patch-Baseline

Linux & macOS

```
aws ssm deregister-patch-baseline-for-patch-group \
    --region us-east-2 \
    --patch-group "Production" \
    --baseline-id "arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE"
```

Windows Server

```
aws ssm deregister-patch-baseline-for-patch-group ^
    --region us-east-2 ^
    --patch-group "Production" ^
    --baseline-id "arn:aws:ssm:us-east-2:111122223333:patchbaseline/
pb-0c10e65780EXAMPLE"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
```

```
    "PatchGroup":"Production",
    "BaselineId":"arn:aws:ssm:us-east-2:111122223333:patchbaseline/pb-0c10e65780EXAMPLE"
}
```

AWS CLI Befehle zum Anzeigen von Patch-Zusammenfassungen und Details

Beispielbefehle zum Anzeigen von Patch-Zusammenfassungen und -details

- Abrufen aller Patches, die in einer bestimmten Patch-Baseline definiert sind
- Holen Sie sich alle Patches für AmazonLinux 2018.03 mit einer Klassifizierung SECURITY und einem Schweregrad von Critical
- Abrufen aller Patches für Windows Server 2012 mit einem MSRC-Schweregrad von Critical
- Abrufen aller verfügbaren Patches
- Abrufen der zusammengefassten Patch-Zustände pro verwalteten Knoten
- Abrufen der Patch-Compliance-Details für einen verwalteten Knoten
- Anzeigen der Patch-Compliance-Ergebnisse (AWS CLI)

Abrufen aller Patches, die in einer bestimmten Patch-Baseline definiert sind

> **Note**
>
> Dieser Befehl wird nur für Windows Server-Patch-Baselines unterstützt.

Linux & macOS

```
aws ssm describe-effective-patches-for-patch-baseline \
    --region us-east-2 \
    --baseline-id "pb-0c10e65780EXAMPLE"
```

Windows Server

```
aws ssm describe-effective-patches-for-patch-baseline ^
    --region us-east-2 ^
    --baseline-id "pb-0c10e65780EXAMPLE"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "NextToken":"--token string truncated--",
   "EffectivePatches":[
      {
         "PatchStatus":{
            "ApprovalDate":1384711200.0,
            "DeploymentStatus":"APPROVED"
         },
         "Patch":{
            "ContentUrl":"https://support.microsoft.com/en-us/kb/2876331",
            "ProductFamily":"Windows",
            "Product":"WindowsServer2012R2",
            "Vendor":"Microsoft",
            "Description":"A security issue has been identified in a Microsoft
 software
               product that could affect your system. You can help protect your system
               by installing this update from Microsoft. For a complete listing of the
               issues that are included in this update, see the associated Microsoft
               Knowledge Base article. After you install this update, you may have to
               restart your system.",
            "Classification":"SecurityUpdates",
            "Title":"Security Update for Windows Server 2012 R2 Preview (KB2876331)",
            "ReleaseDate":1384279200.0,
            "MsrcClassification":"Critical",
            "Language":"All",
            "KbNumber":"KB2876331",
            "MsrcNumber":"MS13-089",
            "Id":"e74ccc76-85f0-4881-a738-59e9fc9a336d"
         }
      },
      {
         "PatchStatus":{
            "ApprovalDate":1428858000.0,
            "DeploymentStatus":"APPROVED"
         },
         "Patch":{
            "ContentUrl":"https://support.microsoft.com/en-us/kb/2919355",
            "ProductFamily":"Windows",
            "Product":"WindowsServer2012R2",
            "Vendor":"Microsoft",
            "Description":"Windows Server 2012 R2 Update is a cumulative
               set of security updates, critical updates and updates. You
               must install Windows Server 2012 R2 Update to ensure that
```

```
              your computer can continue to receive future Windows Updates,
              including security updates. For a complete listing of the
              issues that are included in this update, see the associated
              Microsoft Knowledge Base article for more information. After
              you install this item, you may have to restart your computer.",
            "Classification":"SecurityUpdates",
            "Title":"Windows Server 2012 R2 Update (KB2919355)",
            "ReleaseDate":1428426000.0,
            "MsrcClassification":"Critical",
            "Language":"All",
            "KbNumber":"KB2919355",
            "MsrcNumber":"MS14-018",
            "Id":"8452bac0-bf53-4fbd-915d-499de08c338b"
         }
      }
     ---output truncated---
```

Holen Sie sich alle Patches für AmazonLinux 2018.03 mit einer Klassifizierung **`SECURITY`** und einem Schweregrad von **`Critical`**

Linux & macOS

```
aws ssm describe-available-patches \
    --region us-east-2 \
    --filters Key=PRODUCT,Values=AmazonLinux2018.03 Key=SEVERITY,Values=Critical
```

Windows Server

```
aws ssm describe-available-patches ^
    --region us-east-2 ^
    --filters Key=PRODUCT,Values=AmazonLinux2018.03 Key=SEVERITY,Values=Critical
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "Patches": [
        {
            "AdvisoryIds": ["ALAS-2011-1"],
            "BugzillaIds": [ "1234567" ],
            "Classification": "SECURITY",
            "CVEIds": [ "CVE-2011-3192"],
```

```
            "Name": "zziplib",
            "Epoch": "0",
            "Version": "2.71",
            "Release": "1.3.amzn1",
            "Arch": "i686",
            "Product": "AmazonLinux2018.03",
            "ReleaseDate": 1590519815,
            "Severity": "CRITICAL"
        }
    ]
}
---output truncated---
```

Abrufen aller Patches für Windows Server 2012 mit einem MSRC-Schweregrad von **`Critical`**

Linux & macOS

```
aws ssm describe-available-patches \
    --region us-east-2 \
    --filters Key=PRODUCT,Values=WindowsServer2012 Key=MSRC_SEVERITY,Values=Critical
```

Windows Server

```
aws ssm describe-available-patches ^
    --region us-east-2 ^
    --filters Key=PRODUCT,Values=WindowsServer2012 Key=MSRC_SEVERITY,Values=Critical
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "Patches":[
      {
         "ContentUrl":"https://support.microsoft.com/en-us/kb/2727528",
         "ProductFamily":"Windows",
         "Product":"WindowsServer2012",
         "Vendor":"Microsoft",
         "Description":"A security issue has been identified that could
           allow an unauthenticated remote attacker to compromise your
           system and gain control over it. You can help protect your
           system by installing this update from Microsoft. After you
           install this update, you may have to restart your system.",
         "Classification":"SecurityUpdates",
```

```
         "Title":"Security Update for Windows Server 2012 (KB2727528)",
         "ReleaseDate":1352829600.0,
         "MsrcClassification":"Critical",
         "Language":"All",
         "KbNumber":"KB2727528",
         "MsrcNumber":"MS12-072",
         "Id":"1eb507be-2040-4eeb-803d-abc55700b715"
      },
      {
         "ContentUrl":"https://support.microsoft.com/en-us/kb/2729462",
         "ProductFamily":"Windows",
         "Product":"WindowsServer2012",
         "Vendor":"Microsoft",
         "Description":"A security issue has been identified that could
           allow an unauthenticated remote attacker to compromise your
           system and gain control over it. You can help protect your
           system by installing this update from Microsoft. After you
           install this update, you may have to restart your system.",
         "Classification":"SecurityUpdates",
         "Title":"Security Update for Microsoft .NET Framework 3.5 on
           Windows 8 and Windows Server 2012 for x64-based Systems (KB2729462)",
         "ReleaseDate":1352829600.0,
         "MsrcClassification":"Critical",
         "Language":"All",
         "KbNumber":"KB2729462",
         "MsrcNumber":"MS12-074",
         "Id":"af873760-c97c-4088-ab7e-5219e120eab4"
      }

---output truncated---
```

Abrufen aller verfügbaren Patches

```
aws ssm describe-available-patches --region us-east-2
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "NextToken":"--token string truncated--",
   "Patches":[
      {
            "Classification": "SecurityUpdates",
            "ContentUrl": "https://support.microsoft.com/en-us/kb/4074588",
```

```
            "Description": "A security issue has been identified in a Microsoft
 software
            product that could affect your system. You can help protect your system by
            installing this update from Microsoft. For a complete listing of the
 issues
            that are included in this update, see the associated Microsoft Knowledge
 Base
            article. After you install this update, you may have to restart your
 system.",
            "Id": "11adea10-0701-430e-954f-9471595ae246",
            "KbNumber": "KB4074588",
            "Language": "All",
            "MsrcNumber": "",
            "MsrcSeverity": "Critical",
            "Product": "WindowsServer2016",
            "ProductFamily": "Windows",
            "ReleaseDate": 1518548400,
            "Title": "2018-02 Cumulative Update for Windows Server 2016 (1709) for x64-
based
            Systems (KB4074588)",
            "Vendor": "Microsoft"
        },
        {
            "Classification": "SecurityUpdates",
            "ContentUrl": "https://support.microsoft.com/en-us/kb/4074590",
            "Description": "A security issue has been identified in a Microsoft
 software
            product that could affect your system. You can help protect your system by
            installing this update from Microsoft. For a complete listing of the issues
 that are included in this update, see the associated Microsoft Knowledge Base article.
 After you install this update, you may have to restart your system.",
            "Id": "f5f58231-ac5d-4640-ab1b-9dc8d857c265",
            "KbNumber": "KB4074590",
            "Language": "All",
            "MsrcNumber": "",
            "MsrcSeverity": "Critical",
            "Product": "WindowsServer2016",
            "ProductFamily": "Windows",
            "ReleaseDate": 1518544805,
            "Title": "2018-02 Cumulative Update for Windows Server 2016 for x64-based
            Systems (KB4074590)",
            "Vendor": "Microsoft"
        }
```

```
      ---output truncated---
```

Abrufen der zusammengefassten Patch-Zustände pro verwalteten Knoten

Die Zusammenfassung pro verwaltetem Knoten gibt Ihnen die Anzahl der Patches in den folgenden Zuständen pro Knoten an: "NotApplicable„, „Fehlend“, „Fehlgeschlagen“, "InstalledOther" und „Installiert“.

Linux & macOS

```
aws ssm describe-instance-patch-states \
    --instance-ids i-08ee91c0b17045407 i-09a618aec652973a9
```

Windows Server

```
aws ssm describe-instance-patch-states ^
    --instance-ids i-08ee91c0b17045407 i-09a618aec652973a9
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "InstancePatchStates":[
      {
            "InstanceId": "i-08ee91c0b17045407",
            "PatchGroup": "",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "6d03d6c5-f79d-41d0-8d0e-00a9aEXAMPLE",
            "InstalledCount": 50,
            "InstalledOtherCount": 353,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 0,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": -1,
            "NotApplicableCount": 671,
            "OperationStartTime": "2020-01-24T12:37:56-08:00",
            "OperationEndTime": "2020-01-24T12:37:59-08:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot"
        },
        {
```

```
            "InstanceId": "i-09a618aec652973a9",
            "PatchGroup": "",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "c7e0441b-1eae-411b-8aa7-973e6EXAMPLE",
            "InstalledCount": 36,
            "InstalledOtherCount": 396,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 3,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": -1,
            "NotApplicableCount": 420,
            "OperationStartTime": "2020-01-24T12:37:34-08:00",
            "OperationEndTime": "2020-01-24T12:37:37-08:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot"
        }
    ---output truncated---
```

Abrufen der Patch-Compliance-Details für einen verwalteten Knoten

```
aws ssm describe-instance-patches --instance-id i-08ee91c0b17045407
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "NextToken":"--token string truncated--",
   "Patches":[
      {
            "Title": "bind-libs.x86_64:32:9.8.2-0.68.rc1.60.amzn1",
            "KBId": "bind-libs.x86_64",
            "Classification": "Security",
            "Severity": "Important",
            "State": "Installed",
            "InstalledTime": "2019-08-26T11:05:24-07:00"
        },
        {
            "Title": "bind-utils.x86_64:32:9.8.2-0.68.rc1.60.amzn1",
            "KBId": "bind-utils.x86_64",
            "Classification": "Security",
            "Severity": "Important",
            "State": "Installed",
            "InstalledTime": "2019-08-26T11:05:32-07:00"
```

```
        },
        {
            "Title": "dhclient.x86_64:12:4.1.1-53.P1.28.amzn1",
            "KBId": "dhclient.x86_64",
            "Classification": "Security",
            "Severity": "Important",
            "State": "Installed",
            "InstalledTime": "2019-08-26T11:05:31-07:00"
        },
    ---output truncated---
```

Anzeigen der Patch-Compliance-Ergebnisse (AWS CLI)

Anzeigen von Patch-Compliance-Ergebnissen für einen einzelnen verwalteten Knoten

Führen Sie den folgenden Befehl in AWS Command Line Interface (AWS CLI) aus, um die Ergebnisse der Patch-Konformität für einen einzelnen verwalteten Knoten anzuzeigen.

```
aws ssm describe-instance-patch-states --instance-id instance-id
```

*instance-id*Ersetzen Sie ihn durch die ID des verwalteten Knotens, für den Sie Ergebnisse anzeigen möchten, im Format `i-02573cafcfEXAMPLE` oder`mi-0282f7c436EXAMPLE`.

Das System gibt unter anderem folgende Informationen zurück.

```
{
    "InstancePatchStates": [
        {
            "InstanceId": "i-02573cafcfEXAMPLE",
            "PatchGroup": "mypatchgroup",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
            "CriticalNonCompliantCount": 2,
            "SecurityNonCompliantCount": 2,
            "OtherNonCompliantCount": 1,
            "InstalledCount": 123,
            "InstalledOtherCount": 334,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 1,
            "FailedCount": 2,
            "UnreportedNotApplicableCount": 11,
```

```
            "NotApplicableCount": 2063,
            "OperationStartTime": "2021-05-03T11:00:56-07:00",
            "OperationEndTime": "2021-05-03T11:01:09-07:00",
            "Operation": "Scan",
            "LastNoRebootInstallOperationTime": "2020-06-14T12:17:41-07:00",
            "RebootOption": "RebootIfNeeded"
        }
    ]
}
```

Um eine Übersicht über die Anzahl der Patches für alle EC2 Instanzen in einer Region anzuzeigen

Der `describe-instance-patch-states` unterstützt das Abrufen von Ergebnissen für jeweils eine verwaltete Instance. Wenn Sie jedoch ein benutzerdefiniertes Skript mit dem `describe-instance-patch-states`-Befehl verwenden, können Sie einen detaillierteren Bericht erstellen.

Wenn beispielsweise das jq-Filter-Tool auf Ihrem lokalen Computer installiert ist, könnten Sie den folgenden Befehl ausführen, um zu ermitteln, welche Ihrer EC2 Instanzen in einer bestimmten Instanz den Status AWS-Region haben. `InstalledPendingReboot`

```
aws ssm describe-instance-patch-states \
    --instance-ids $(aws ec2 describe-instances --region region | jq
 '.Reservations[].Instances[] | .InstanceId' | tr '\n|"' ' ') \
    --output text --query 'InstancePatchStates[*].{Instance:InstanceId,
 InstalledPendingRebootCount:InstalledPendingRebootCount}'
```

*region*steht für die Kennung einer Region, die von AWS-Region unterstützt wird AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

Zum Beispiel:

```
aws ssm describe-instance-patch-states \
    --instance-ids $(aws ec2 describe-instances --region us-east-2 | jq
 '.Reservations[].Instances[] | .InstanceId' | tr '\n|"' ' ') \
    --output text --query 'InstancePatchStates[*].{Instance:InstanceId,
 InstalledPendingRebootCount:InstalledPendingRebootCount}'
```

Das System gibt unter anderem folgende Informationen zurück

```
1       i-02573cafcfEXAMPLE
0       i-0471e04240EXAMPLE
3       i-07782c72faEXAMPLE
6       i-083b678d37EXAMPLE
0       i-03a530a2d4EXAMPLE
1       i-01f68df0d0EXAMPLE
0       i-0a39c0f214EXAMPLE
7       i-0903a5101eEXAMPLE
7       i-03823c2fedEXAMPLE
```

Zusätzlich zu `InstalledPendingRebootCount` können Sie nach den folgenden Anzahltypen suchen:

- `CriticalNonCompliantCount`
- `SecurityNonCompliantCount`
- `OtherNonCompliantCount`
- `UnreportedNotApplicableCount`
- `InstalledPendingRebootCount`
- `FailedCount`
- `NotApplicableCount`
- `InstalledRejectedCount`
- `InstalledOtherCount`
- `MissingCount`
- `InstalledCount`

AWS CLI Befehle zum Scannen und Patchen verwalteter Knoten

Nachdem Sie die folgenden Befehle ausgeführt haben, um nach Patch-Compliance zu scannen oder Patches zu installieren, können Sie mit Befehlen im AWS CLI Befehle zum Anzeigen von Patch-Zusammenfassungen und Details-Abschnitt Informationen zu Patch-Status und -Compliance anzeigen.

Beispielbefehle

- Verwaltete Knoten auf Patch-Compliance scannen (AWS CLI)
- Installieren von Patches auf verwalteten Knoten (AWS CLI)

Verwaltete Knoten auf Patch-Compliance scannen (AWS CLI)

So scannen Sie spezifische verwaltete Knoten auf Patch-Compliance

Führen Sie den folgenden Befehl aus.

Linux & macOS

```
aws ssm send-command \
    --document-name 'AWS-RunPatchBaseline' \
    --targets Key=InstanceIds,Values='i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE' \
    --parameters 'Operation=Scan' \
    --timeout-seconds 600
```

Windows Server

```
aws ssm send-command ^
    --document-name "AWS-RunPatchBaseline" ^
    --targets Key=InstanceIds,Values="i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE" ^
    --parameters "Operation=Scan" ^
    --timeout-seconds 600
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "Command": {
        "CommandId": "a04ed06c-8545-40f4-87c2-a0babEXAMPLE",
        "DocumentName": "AWS-RunPatchBaseline",
        "DocumentVersion": "$DEFAULT",
        "Comment": "",
        "ExpiresAfter": 1621974475.267,
        "Parameters": {
            "Operation": [
                "Scan"
            ]
        },
        "InstanceIds": [],
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "i-02573cafcfEXAMPLE,
```

```
                    i-0471e04240EXAMPLE"
                ]
            }
        ],
        "RequestedDateTime": 1621952275.267,
        "Status": "Pending",
        "StatusDetails": "Pending",
        "TimeoutSeconds": 600,

    ---output truncated---

    }
}
```

So scannen Sie verwaltete Knoten nach Patch-Gruppentag auf Patch-Compliance

Führen Sie den folgenden Befehl aus.

Linux & macOS

```
aws ssm send-command \
    --document-name 'AWS-RunPatchBaseline' \
    --targets Key='tag:PatchGroup',Values='Web servers' \
    --parameters 'Operation=Scan' \
    --timeout-seconds 600
```

Windows Server

```
aws ssm send-command ^
    --document-name "AWS-RunPatchBaseline" ^
    --targets Key="tag:PatchGroup",Values="Web servers" ^
    --parameters "Operation=Scan" ^
    --timeout-seconds 600
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "Command": {
        "CommandId": "87a448ee-8adc-44e0-b4d1-6b429EXAMPLE",
        "DocumentName": "AWS-RunPatchBaseline",
        "DocumentVersion": "$DEFAULT",
        "Comment": "",
```

```
        "ExpiresAfter": 1621974983.128,
        "Parameters": {
            "Operation": [
                "Scan"
            ]
        },
        "InstanceIds": [],
        "Targets": [
            {
                "Key": "tag:PatchGroup",
                "Values": [
                    "Web servers"
                ]
            }
        ],
        "RequestedDateTime": 1621952783.128,
        "Status": "Pending",
        "StatusDetails": "Pending",
        "TimeoutSeconds": 600,

    ---output truncated---

    }
}
```

Installieren von Patches auf verwalteten Knoten (AWS CLI)

So installieren Sie Patches auf spezifischen verwalteten Knoten

Führen Sie den folgenden Befehl aus.

> **Note**
>
> Die anvisierten verwalteten Knoten werden nach Bedarf neu gestartet, um die Patch-Installation abzuschließen. Weitere Informationen finden Sie unter SSM-Befehlsdokument zum Patchen: `AWS-RunPatchBaseline`.

Linux & macOS

```
aws ssm send-command \
    --document-name 'AWS-RunPatchBaseline' \
    --targets Key=InstanceIds,Values='i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE' \
```

```
    --parameters 'Operation=Install' \
    --timeout-seconds 600
```

Windows Server

```
aws ssm send-command ^
    --document-name "AWS-RunPatchBaseline" ^
    --targets Key=InstanceIds,Values="i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE" ^
    --parameters "Operation=Install" ^
    --timeout-seconds 600
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "Command": {
        "CommandId": "5f403234-38c4-439f-a570-93623EXAMPLE",
        "DocumentName": "AWS-RunPatchBaseline",
        "DocumentVersion": "$DEFAULT",
        "Comment": "",
        "ExpiresAfter": 1621975301.791,
        "Parameters": {
            "Operation": [
                "Install"
            ]
        },
        "InstanceIds": [],
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "i-02573cafcfEXAMPLE,
                     i-0471e04240EXAMPLE"
                ]
            }
        ],
        "RequestedDateTime": 1621953101.791,
        "Status": "Pending",
        "StatusDetails": "Pending",
        "TimeoutSeconds": 600,

    ---output truncated---
```

```
    }
}
```

So installieren Sie Patches auf verwalteten Knoten in einer spezifischen Patch-Gruppe

Führen Sie den folgenden Befehl aus.

Linux & macOS

```
aws ssm send-command \
    --document-name 'AWS-RunPatchBaseline' \
    --targets Key='tag:PatchGroup',Values='Web servers' \
    -parameters 'Operation=Install' \
    --timeout-seconds 600
```

Windows Server

```
aws ssm send-command ^
    --document-name "AWS-RunPatchBaseline" ^
    --targets Key="tag:PatchGroup",Values="Web servers" ^
    --parameters "Operation=Install" ^
    --timeout-seconds 600
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "Command": {
        "CommandId": "fa44b086-7d36-4ad5-ac8d-627ecEXAMPLE",
        "DocumentName": "AWS-RunPatchBaseline",
        "DocumentVersion": "$DEFAULT",
        "Comment": "",
        "ExpiresAfter": 1621975407.865,
        "Parameters": {
            "Operation": [
                "Install"
            ]
        },
        "InstanceIds": [],
        "Targets": [
            {
                "Key": "tag:PatchGroup",
                "Values": [
```

```
                    "Web servers"
                ]
            }
        ],
        "RequestedDateTime": 1621953207.865,
        "Status": "Pending",
        "StatusDetails": "Pending",
        "TimeoutSeconds": 600,

    ---output truncated---

    }
}
```

AWS Systems Manager Patch Manager Tutorials

Die Tutorials in diesem Abschnitt zeigenPatch Manager, wie Sie ein Tool in AWS Systems Manager verschiedenen Patch-Szenarien verwenden können.

Themen

- Tutorial: Einen Server in einer IPv6 einzigen Umgebung patchen
- So erstellen Sie eine Patch-Baseline für die Installation von Windows Service Packs mithilfe der Konsole
- Tutorial: Aktualisieren von Anwendungsabhängigkeiten, Patchen eines verwalteten Knotens und Durchführen einer anwendungsspezifischen Zustandsprüfung mithilfe der Konsole
- Tutorial: Patchen Sie eine Serverumgebung mit dem AWS CLI

Tutorial: Einen Server in einer IPv6 einzigen Umgebung patchen

Patch Managerunterstützt das Patchen von Knoten in Umgebungen, in denen nur IPv6 Durch die Aktualisierung der SSM Agent Konfiguration können Patch-Vorgänge so konfiguriert werden, dass nur Aufrufe an IPv6 Dienstendpunkte getätigt werden.

Um einen Server in einer IPv6 reinen Umgebung zu patchen

1. Stellen Sie sicher, dass SSM Agent Version 3.3270.0 oder höher auf dem verwalteten Knoten installiert ist.
2. Navigieren Sie auf dem verwalteten Knoten zur Konfigurationsdatei. SSM Agent Sie finden die `amazon-ssm-agent.json` Datei in den folgenden Verzeichnissen:

- Linux: `/etc/amazon/ssm/`
- macOS: `/opt/aws/ssm/`
- Windows Server: `C:\Program Files\Amazon\SSM`

Falls sie noch nicht `amazon-ssm-agent.json` existiert, kopieren Sie den Inhalt von `amazon-ssm-agent.json.template` unter demselben Verzeichnis nach`amazon-ssm-agent.json`.

3. Aktualisieren Sie den folgenden Eintrag, um die richtige Region festzulegen, und stellen Sie ihn `UseDualStackEndpoint` auf ein`true`:

```
{
 --------
    "Agent": {
        "Region": "region",
        "UseDualStackEndpoint": true
    },
--------
}
```

4. Starten Sie den SSM Agent Dienst mit dem entsprechenden Befehl für Ihr Betriebssystem neu:

   - Linux: `sudo systemctl restart amazon-ssm-agent`
   - Ubuntu Servermit Snap: `sudo snap restart amazon-ssm-agent`
   - macOS: `sudo launchctl stop com.amazon.aws.ssm` gefolgt von `sudo launchctl start com.amazon.aws.ssm`
   - Windows Server: `Stop-Service AmazonSSMAgent` gefolgt von `Start-Service AmazonSSMAgent`

   Die vollständige Liste der Befehle pro Betriebssystem finden Sie unterthe section called “Prüfen des SSM Agent-Status und Starten des Agenten”.

5. Führen Sie einen beliebigen Patchvorgang aus, um sicherzustellen, dass die Patchvorgänge in Ihrer reinen Umgebung erfolgreich sind IPv6. Stellen Sie sicher, dass die Knoten, für die gepatcht wird, eine Verbindung zur Patch-Quelle haben. Sie können die Run Command Ausgabe der Patching-Ausführung überprüfen, um nach Warnungen zu unzugänglichen Repositorys zu suchen. Achten Sie beim Patchen eines Knotens, der in einer IPv6 reinen Umgebung läuft, darauf, dass der Knoten mit der Patch-Quelle verbunden ist. Sie können die Run Command Ausgabe der Patching-Ausführung überprüfen, um nach Warnungen zu unzugänglichen

Repositorys zu suchen. Für DNF-basierte Betriebssysteme ist es möglich, nicht verfügbare Repositorys so zu konfigurieren, dass sie beim Patchen übersprungen werden, wenn die Option auf unter gesetzt ist. `skip_if_unavailable` `True` `/etc/dnf/dnf.conf` Zu den DNF-basierten Betriebssystemen gehören Amazon Linux 2023, Red Hat Enterprise Linux 8 und spätere Versionen, Oracle Linux Rocky Linux AlmaLinux, und CentOS 8 und spätere Versionen. Unter Amazon Linux 2023 ist die `skip_if_unavailable` Option `True` standardmäßig auf eingestellt.

> **Note**
>
> Wenn Sie die Funktionen Install Override List oder Baseline Override verwenden, stellen Sie sicher, dass die angegebene URL vom Knoten aus erreichbar ist. Wenn die SSM Agent Konfigurationsoption auf gesetzt `UseDualStackEndpoint` ist`true`, wird ein Dual-Stack-S3-Client verwendet, wenn eine S3-URL bereitgestellt wird.

So erstellen Sie eine Patch-Baseline für die Installation von Windows Service Packs mithilfe der Konsole

Wenn Sie eine benutzerdefinierte Patch-Baseline erstellen, können Sie angeben, ob alle, einige oder nur ein einziger unterstützter Patch-Typ installiert wird.

In den Patch-Baselines für Windows können Sie `ServicePacks` als einzige Klassifizierungsoption auswählen, um Patching-Updates auf Service Packs einzuschränken. Service Packs können von Patch Manager, einem Tool in AWS Systems Manager, automatisch installiert werden, wenn das Update in Windows Update oder Windows Server Update Services (WSUS) verfügbar ist.

Sie können eine Patch-Baseline konfigurieren, um festzulegen, ob Service Packs für alle Windows-Versionen oder nur für bestimmte Windows-Versionen installiert werden, z. B. Windows 7 oder Windows Server 2016.

Gehen Sie wie folgt vor, um eine benutzerdefinierte Patch-Baseline zu erstellen, die ausschließlich für die Installation aller Service Packs auf Ihren Windows-verwalteten Knoten verwendet wird.

So erstellen Sie eine Patch-Baseline für die Installation von Windows Service Packs (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.

3. Wählen Sie die Registerkarte Patch-Baselines und dann Patch-Baseline erstellen aus.
4. Geben Sie im Feld Name einen Namen für die neue Patch-Baseline ein, z. B. `MyWindowsServicePackPatchBaseline`.
5. (Optional) Geben Sie im Feld Beschreibung eine Beschreibung für diese Patch-Baseline ein.
6. Wählen Sie unter Betriebssystem die Option `Windows` aus.
7. Wenn Sie diese Patch-Baseline direkt nach dem Erstellen als Standard für Windows verwenden möchten, wählen Sie Set this patch baseline as the default patch baseline for Windows Server instances (Diese Patch-Baseline als Standard-Patch-Baseline für Windows Server-Instances festlegen) aus.

   > **Note**
   >
   > Diese Option ist nur verfügbar, wenn Sie vor der Veröffentlichung der Patch-Richtlinien am 22. Dezember 2022 zum ersten Mal auf Patch Manager zugegriffen haben. Weitere Informationen zum Festlegen einer vorhandenen Patch-Baseline als Standard finden Sie unter Festlegen einer vorhandenen Patch-Baseline als Standard.

8. Erstellen Sie im Abschnitt Approval Rules for operating-systems (Genehmigungsregeln für Betriebssysteme) unter Verwendung der Felder ein oder mehrere automatische Genehmigungsregeln.

   - Produkte: Die Betriebssystemversionen, auf die sich die Genehmigungsregel bezieht, z. B. `WindowsServer2012`. Sie können eine, mehr als eine oder alle unterstützten Windows-Versionen auswählen. Die Standardauswahl ist `All`.
   - Classification (Klassifizierung): Wählen Sie `ServicePacks` aus.
   - Severity (Schweregrad): Der Schweregradwert der Patches, auf die die Regel angewendet werden soll. Um sicherzustellen, dass alle Service Packs von der Regel eingeschlossen werden, wählen Sie `All` aus.
   - Automatische Genehmigung: Die Methode zum Auswählen von Patches für die automatische Genehmigung.
     - Patches nach einer bestimmten Anzahl von Tagen genehmigen: Die Anzahl der Tage, die der Patch Manager warten muss, nachdem ein Patch veröffentlicht oder aktualisiert wurde, bevor ein Patch automatisch genehmigt wird. Sie können jede Ganzzahl von Null (0) bis 360 eingeben. Für die meisten Szenarien empfehlen wir, nicht länger als 100 Tage zu warten.

- Patches genehmigen, die bis zu einem bestimmten Datum veröffentlicht wurden: Das Datum der Patch-Veröffentlichung, an dem der Patch Manager automatisch alle Patches anwendet, die bis zu diesem Datum veröffentlicht oder aktualisiert wurden. Wenn Sie beispielsweise den 07. Juli 2023 angeben, werden Patches, die am oder nach dem 08. Juli 2023 veröffentlicht oder zuletzt aktualisiert wurden, nicht automatisch installiert.

- (Optional) Compliance reporting (Compliance-Berichte): Der Schweregrad, den Sie Service Packs zuweisen möchten, die von der Baseline genehmigt wurden, z. B. `High`.

> **Note**
>
> Wenn Sie eine Konformitätsberichtsstufe angeben und der Patch-Status eines genehmigten Service-Packs als `Missing` gemeldet wird, dann entspricht der insgesamt gemeldete Konformitätsschweregrad der Patch-Baseline dem von Ihnen angegebenen Schweregrad.

9. (Optional) Wählen Sie für Tags verwalten ein oder mehrere Tag-Schlüsselname/Wertpaare für die Patch-Baseline aus.

   Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Für diese Patch-Baseline, die der Aktualisierung von Service Packs gewidmet ist, könnten Sie Schlüssel-Wert-Paare angeben, z. B.:

   - `Key=OS,Value=Windows`
   - `Key=Classification,Value=ServicePacks`

10. Wählen Sie die Option Patch-Baseline erstellen.

Tutorial: Aktualisieren von Anwendungsabhängigkeiten, Patchen eines verwalteten Knotens und Durchführen einer anwendungsspezifischen Zustandsprüfung mithilfe der Konsole

In vielen Fällen muss ein verwalteter Knoten neu gestartet werden, nachdem er mit dem neuesten Softwareupdate gepatcht wurde. Ein Neustart eines verwalteten Knotens in der Produktion ohne vorhandene Sicherheitsvorkehrungen kann jedoch mehrere Probleme verursachen, z. B. das Aufrufen von Alarmen, das Aufzeichnen falscher Metrikdaten und das Unterbrechen von Datensynchronisationen.

Diese Anleitung zeigt, wie Sie Probleme wie diese vermeiden können, indem Sie das AWS Systems Manager -Dokument (SSM-Dokument) `AWS-RunPatchBaselineWithHooks` verwenden, um einen komplexen, mehrstufigen Patchvorgang zu erreichen, der Folgendes ausführt:

1. Verhindern neuer Verbindungen mit der Anwendung
2. Installieren von Betriebssystem-Updates
3. Aktualisieren der Paketabhängigkeiten der Anwendung
4. Neustart des Systems
5. Durchführen einer anwendungsspezifischen Zustandsprüfung

Für dieses Beispiel haben wir unsere Infrastruktur auf diese Weise eingerichtet:

- Die anvisierten virtuellen Maschinen werden als verwaltete Knoten mit Systems Manager registriert.
- `Iptables` wird als lokale Firewall verwendet.
- Die auf den verwalteten Knoten gehostete Anwendung wird auf Port 443 ausgeführt.
- Die Anwendung, die auf den verwalteten Knoten gehostet wird, ist eine `nodeJS`-Anwendung.
- Die auf den verwalteten Knoten gehostete Anwendung wird vom pm2-Prozessmanager verwaltet.
- Die Anwendung verfügt bereits über einen angegebenen Zustandsprüfungs-Endpunkt.
- Der Endpunkt der Zustandsprüfung der Anwendung erfordert keine Endbenutzerauthentifizierung. Der Endpunkt ermöglicht eine Zustandsprüfung, die die Anforderungen der Organisation beim Festlegen der Verfügbarkeit erfüllt. (In Ihrer Umgebung reicht es möglicherweise aus, sicherzustellen, dass die `nodeJS`-Anwendung ausgeführt wird und in der Lage ist, auf Anfragen zu warten. In anderen Fällen möchten Sie möglicherweise überprüfen, ob bereits eine Verbindung zur Caching-Ebene oder zur Datenbankebene hergestellt wurde).

Die Beispiele in dieser Anleitung dienen nur zu Demonstrationszwecken und sind nicht dafür gedacht, in Produktionsumgebungen implementiert zu werden. Beachten Sie auch, dass das Lebenszyklus-Hook-Feature von Patch Manager, einem Tool in Systems Manager, mit dem `AWS-RunPatchBaselineWithHooks`-Dokument zahlreiche andere Szenarien unterstützen kann. Im Folgenden finden Sie einige Beispiele.

- Stoppen Sie einen Metriken meldenden Agenten, bevor Sie ihn patchen und neu starten, nachdem der verwaltete Knoten neu gestartet wurde.

- Trennen Sie den verwalteten Knoten vor dem Patchen von einem CRM- oder PCS-Cluster und fügen Sie sie nach dem Neustart des Knoten erneut an.
- Aktualisieren Sie Software von Drittanbietern (z. B. Java, Tomcat, Adobe-Anwendungen usw.) auf Windows Server-Maschinen nach dem Anwenden von Betriebssystem-Updates, jedoch vor dem Neustart des verwalteten Knoten.

So aktualisieren Sie Anwendungsabhängigkeite, patchen einen verwalteten Knoten und führen eine anwendungsspezifische Zustandsprüfung durch

1. Erstellen Sie ein SSM-Dokument für Ihr Vorinstallations-Skript mit dem folgenden Inhalt und geben Sie ihm den Namen `NodeJSAppPrePatch`. Ersetzen Sie *`your_application`* mit dem Namen Ihrer Anwendung.

   Dieses Skript blockiert sofort neue eingehende Anforderungen und lässt fünf Sekunden, damit bereits aktive Anforderungen abgeschlossen werden können, bevor der Patchvorgang gestartet wird. Für die `sleep`-Option geben Sie einen Wert in Sekunden an, der größer ist als die Dauer, bis eingehende Anforderungen normalerweise abgeschlossen werden.

   ```
   # exit on error
   set -e
   # set up rule to block incoming traffic
   iptables -I INPUT -j DROP -p tcp --syn --destination-port 443 || exit 1
   # wait for current connections to end. Set timeout appropriate to your
    application's latency
   sleep 5
   # Stop your application
   pm2 stop your_application
   ```

   Informationen zum Erstellen von SSM-Dokumenten finden Sie unter Erstellen von SSM-Dokumentinhalten.

2. Erstellen Sie ein weiteres SSM-Dokument mit folgendem Inhalt für Ihr Postinstall-Skript, um Ihre Anwendungsabhängigkeiten zu aktualisieren, und nennen Sie es `NodeJSAppPostPatch`. Ersetzen Sie es */your/application/path* durch den Pfad zu Ihrer Anwendung.

   ```
   cd /your/application/path
   npm update
   # you can use npm-check-updates if you want to upgrade major versions
   ```

3. Erstellen Sie ein weiteres SSM-Dokument mit folgendem Inhalt für Ihr `onExit`-Skript, um Ihre Anwendung zu sichern und eine Zustandsprüfung durchzuführen. Nennen Sie dieses SSM-Dokument `NodeJSApp0nExitPatch`. Ersetzen Sie *`your_application`* mit dem Namen Ihrer Anwendung.

```
# exit on error
set -e
# restart nodeJs application
pm2 start your_application
# sleep while your application starts and to allow for a crash
sleep 10
# check with pm2 to see if your application is running
pm2 pid your_application
# re-enable incoming connections
iptables -D INPUT -j DROP -p tcp --syn --destination-port
# perform health check
/usr/bin/curl -m 10 -vk -A "" http://localhost:443/health-check || exit 1
```

4. Erstellen Sie eine Zuordnung in State Manager, einem Tool in AWS Systems Manager, um mithilfe der folgenden Schritte den Vorgang auszuführen:

   1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
   2. Wählen Sie im Navigationsbereich State Manager und anschließend Zuordnung erstellen aus.
   3. Für Name geben Sie einen Namen ein, um den Zweck der Zuordnung zu identifizieren.
   4. Wählen Sie in der Liste Dokument die Option `AWS-RunPatchBaselineWithHooks` aus.
   5. Wählen Sie für Operation die Option Install (Installieren) aus.
   6. (Optional) Für Snapshot-ID, stellen Sie eine GUID bereit, die Sie generieren, um den Vorgang zu beschleunigen und Konsistenz zu gewährleisten. Der GUID-Wert kann so einfach sein wie `00000000-0000-0000-0000-111122223333`.
   7. Für Pre Install Hook Doc Name geben Sie `NodeJSAppPrePatch` ein.
   8. Für Post Install Hook Doc Name geben Sie `NodeJSAppPostPatch` ein.
   9. Geben Sie für On ExitHook Doc-Name den Wert ein`NodeJSApp0nExitPatch`.

5. Für Targets (Ziele), identifizieren Sie Ihre verwalteten Knoten, indem Sie Tags angeben, Knoten manuell auswählen, eine Ressourcengruppe auswählen oder alle verwaltete Knoten auswählen.

6. Für Specify schedule (Zeitplan angeben) geben Sie an, wie oft die Zuordnung ausgeführt werden soll. Für einen verwalteten Knoten ist das Patchen einmal pro Woche beispielsweise eine übliche Kadenz.
7. Wählen Sie im Abschnitt Rate control (Ratensteuerung) Optionen für die Ausführung der Zuordnung auf mehreren verwalteten Knoten aus. Stellen Sie sicher, dass nur ein Teil der verwalteten Knoten gleichzeitig aktualisiert wird. Andernfalls könnte die gesamte oder die meisten Ihrer Flotte gleichzeitig offline geschaltet werden. Weitere Informationen zu Ratensteuerungen finden Sie unter Verstehen von Zielen und Ratensteuerungen in State Manager Zuordnungen.
8. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.
9. Wählen Sie Zuordnung erstellen.

Tutorial: Patchen Sie eine Serverumgebung mit dem AWS CLI

In der folgenden Prozedur wird beschrieben, wie Sie eine Serverumgebung mithilfe einer angepassten Patch-Baseline, Patch-Gruppen und einem Wartungsfenster patchen.

Bevor Sie beginnen

- Installieren oder Aktualisieren des SSM Agent auf Ihren verwalteten Knoten. Um von Linux verwaltete Knoten zu patchen, müssen Ihre Knoten SSM Agent der Version 2.0.834.0 oder höher ausführen. Weitere Informationen finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command.

- Konfigurieren Sie Rollen und Berechtigungen für Maintenance Windows ein Tool in AWS Systems Manager. Weitere Informationen finden Sie unter Einrichten von Maintenance Windows.
- Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

  Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

So konfigurieren Sie Patch Manager und spielen Patches für verwaltete Knoten ein (Befehlszeile)

1. Führen Sie den folgenden Befehl aus, um eine Patch-Baseline für Windows mit dem Namen `Production-Baseline` zu erstellen. Diese Patch-Baseline genehmigt Patches für eine Produktionsumgebung sieben Tage nach ihrer Veröffentlichung oder letzten Aktualisierung. Darüber hinaus wurde die Patch-Baseline markiert, um anzuzeigen, dass sie für eine Produktionsumgebung bestimmt ist.

   > **Note**
   >
   > Der `OperatingSystem`-Parameter und `PatchFilters` variieren je nach Betriebssystem der anvisierten verwalteten Knoten, für die die Patch-Baseline gilt. Weitere Informationen erhalten Sie unter OperatingSystem und PatchFilter.

   Linux & macOS

   ```
   aws ssm create-patch-baseline \
       --name "Production-Baseline" \
       --operating-system "WINDOWS" \
       --tags "Key=Environment,Value=Production" \
       --approval-rules
    "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,Impor
   {Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,ServicePacks,UpdateRollups,CriticalU
    \
       --description "Baseline containing all updates approved for production
    systems"
   ```

   Windows Server

   ```
   aws ssm create-patch-baseline ^
   ```

```
    --name "Production-Baseline" ^
    --operating-system "WINDOWS" ^
    --tags "Key=Environment,Value=Production" ^
    --approval-rules
 "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,Impor
{Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,ServicePacks,UpdateRollups,CriticalU
 ^
    --description "Baseline containing all updates approved for production
 systems"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

2. Führen Sie die folgenden Befehle aus, um die Patch-Baseline „Production-Baseline“ für zwei Patchgruppen zu registrieren. Die Gruppen heißen „Datenbankserver“ und „Front-End-Server“.

   Linux & macOS

   ```
   aws ssm register-patch-baseline-for-patch-group \
       --baseline-id pb-0c10e65780EXAMPLE \
       --patch-group "Database Servers"
   ```

   Windows Server

   ```
   aws ssm register-patch-baseline-for-patch-group ^
       --baseline-id pb-0c10e65780EXAMPLE ^
       --patch-group "Database Servers"
   ```

   Das System gibt unter anderem folgende Informationen zurück

   ```
   {
      "PatchGroup":"Database Servers",
      "BaselineId":"pb-0c10e65780EXAMPLE"
   }
   ```

Linux & macOS

```
aws ssm register-patch-baseline-for-patch-group \
    --baseline-id pb-0c10e65780EXAMPLE \
    --patch-group "Front-End Servers"
```

Windows Server

```
aws ssm register-patch-baseline-for-patch-group ^
    --baseline-id pb-0c10e65780EXAMPLE ^
    --patch-group "Front-End Servers"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "PatchGroup":"Front-End Servers",
   "BaselineId":"pb-0c10e65780EXAMPLE"
}
```

3. Führen Sie die folgenden Befehle aus, um zwei Wartungsfenster für die Produktionsserver zu erstellen. Das erste Zeitfenster beginnt jeden Dienstag um 22:00 Uhr. Das zweite Zeitfenster beginnt jeden Samstag um 22:00 Uhr. Darüber hinaus wird das Wartungsfenster mit Tags versehen, um anzugeben, das es für eine Produktionsumgebung vorgesehen ist.

   Linux & macOS

   ```
   aws ssm create-maintenance-window \
       --name "Production-Tuesdays" \
       --tags "Key=Environment,Value=Production" \
       --schedule "cron(0 0 22 ? * TUE *)" \
       --duration 1 \
       --cutoff 0 \
       --no-allow-unassociated-targets
   ```

   Windows Server

   ```
   aws ssm create-maintenance-window ^
       --name "Production-Tuesdays" ^
       --tags "Key=Environment,Value=Production" ^
   ```

```
    --schedule "cron(0 0 22 ? * TUE *)" ^
    --duration 1 ^
    --cutoff 0 ^
    --no-allow-unassociated-targets
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "WindowId":"mw-0c50858d01EXAMPLE"
}
```

Linux & macOS

```
aws ssm create-maintenance-window \
    --name "Production-Saturdays" \
    --tags "Key=Environment,Value=Production" \
    --schedule "cron(0 0 22 ? * SAT *)" \
    --duration 2 \
    --cutoff 0 \
    --no-allow-unassociated-targets
```

Windows Server

```
aws ssm create-maintenance-window ^
    --name "Production-Saturdays" ^
    --tags "Key=Environment,Value=Production" ^
    --schedule "cron(0 0 22 ? * SAT *)" ^
    --duration 2 ^
    --cutoff 0 ^
    --no-allow-unassociated-targets
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "WindowId":"mw-9a8b7c6d5eEXAMPLE"
}
```

4. Führen Sie die folgenden Befehle aus, um die Server-Patch-Gruppen `Database` und `Front-End` mit ihren jeweiligen Wartungsfenstern zu registrieren.

Linux & macOS

```
aws ssm register-target-with-maintenance-window \
    --window-id mw-0c50858d01EXAMPLE \
    --targets "Key=tag:PatchGroup,Values=Database Servers" \
    --owner-information "Database Servers" \
    --resource-type "INSTANCE"
```

Windows Server

```
aws ssm register-target-with-maintenance-window ^
    --window-id mw-0c50858d01EXAMPLE ^
    --targets "Key=tag:PatchGroup,Values=Database Servers" ^
    --owner-information "Database Servers" ^
    --resource-type "INSTANCE"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "WindowTargetId":"e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
}
```

Linux & macOS

```
aws ssm register-target-with-maintenance-window \
--window-id mw-9a8b7c6d5eEXAMPLE \
--targets "Key=tag:PatchGroup,Values=Front-End Servers" \
--owner-information "Front-End Servers" \
--resource-type "INSTANCE"
```

Windows Server

```
aws ssm register-target-with-maintenance-window ^
    --window-id mw-9a8b7c6d5eEXAMPLE ^
    --targets "Key=tag:PatchGroup,Values=Front-End Servers" ^
    --owner-information "Front-End Servers" ^
    --resource-type "INSTANCE"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "WindowTargetId":"faa01c41-1d57-496c-ba77-ff9caEXAMPLE"
}
```

5. Führen Sie die folgenden Befehle aus, um eine Patch-Aufgabe zu registrieren, die während der entsprechenden Wartungsfenster fehlende Updates auf den Servern `Database` und `Front-End` installiert.

   Linux & macOS

   ```
   aws ssm register-task-with-maintenance-window \
       --window-id mw-0c50858d01EXAMPLE \
       --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
    \
       --task-arn "AWS-RunPatchBaseline" \
       --service-role-arn "arn:aws:iam::123456789012:role/MW-Role" \
       --task-type "RUN_COMMAND" \
       --max-concurrency 2 \
       --max-errors 1 \
       --priority 1 \
       --task-invocation-parameters "RunCommand={Parameters={Operation=Install}}"
   ```

   Windows Server

   ```
   aws ssm register-task-with-maintenance-window ^
       --window-id mw-0c50858d01EXAMPLE ^
       --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
    ^
       --task-arn "AWS-RunPatchBaseline" ^
       --service-role-arn "arn:aws:iam::123456789012:role/MW-Role" ^
       --task-type "RUN_COMMAND" ^
       --max-concurrency 2 ^
       --max-errors 1 ^
       --priority 1 ^
       --task-invocation-parameters "RunCommand={Parameters={Operation=Install}}"
   ```

   Das System gibt unter anderem folgende Informationen zurück

```
{
   "WindowTaskId":"4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
}
```

Linux & macOS

```
aws ssm register-task-with-maintenance-window \
    --window-id mw-9a8b7c6d5eEXAMPLE \
    --targets "Key=WindowTargetIds,Values=faa01c41-1d57-496c-ba77-ff9caEXAMPLE"
 \
    --task-arn "AWS-RunPatchBaseline" \
    --service-role-arn "arn:aws:iam::123456789012:role/MW-Role" \
    --task-type "RUN_COMMAND" \
    --max-concurrency 2 \
    --max-errors 1 \
    --priority 1 \
    --task-invocation-parameters "RunCommand={Parameters={Operation=Install}}"
```

Windows Server

```
aws ssm register-task-with-maintenance-window ^
    --window-id mw-9a8b7c6d5eEXAMPLE ^
    --targets "Key=WindowTargetIds,Values=faa01c41-1d57-496c-ba77-ff9caEXAMPLE"
 ^
    --task-arn "AWS-RunPatchBaseline" ^
    --service-role-arn "arn:aws:iam::123456789012:role/MW-Role" ^
    --task-type "RUN_COMMAND" ^
    --max-concurrency 2 ^
    --max-errors 1 ^
    --priority 1 ^
    --task-invocation-parameters "RunCommand={Parameters={Operation=Install}}"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "WindowTaskId":"8a5c4629-31b0-4edd-8aea-33698EXAMPLE"
}
```

6. Führen Sie den folgenden Befehl aus, um für eine Patch-Gruppe eine allgemeine Zusammenfassung zur Patch-Compliance abzurufen. Die allgemeine Zusammenfassung der Patch-Compliance enthält die Anzahl der verwalteten Knoten mit Patches in den jeweiligen Patch-Zuständen.

   > **Note**
   >
   > Es werden Nullen für die Anzahl der verwalteten Knoten in der Zusammenfassung erwartet, bis die Patch-Aufgabe während des ersten Wartungsfensters ausgeführt wird.

   Linux & macOS

   ```
   aws ssm describe-patch-group-state \
       --patch-group "Database Servers"
   ```

   Windows Server

   ```
   aws ssm describe-patch-group-state ^
       --patch-group "Database Servers"
   ```

   Das System gibt unter anderem folgende Informationen zurück

   ```
   {
      "Instances": number,
      "InstancesWithFailedPatches": number,
      "InstancesWithInstalledOtherPatches": number,
      "InstancesWithInstalledPatches": number,
      "InstancesWithInstalledPendingRebootPatches": number,
      "InstancesWithInstalledRejectedPatches": number,
      "InstancesWithMissingPatches": number,
      "InstancesWithNotApplicablePatches": number,
      "InstancesWithUnreportedNotApplicablePatches": number
   }
   ```

7. Führen Sie den folgenden Befehl aus, um für eine Patch-Gruppe eine Übersicht über den Patch-Zustand auf der Ebene einzelner verwalteter Knoten abzurufen. Die Zusammenfassung pro verwalteter Knoten enthält eine Anzahl von Patches in den jeweiligen Patch-Zuständen pro verwalteten Knoten für eine Patch-Gruppe.

Linux & macOS

```
aws ssm describe-instance-patch-states-for-patch-group \
    --patch-group "Database Servers"
```

Windows Server

```
aws ssm describe-instance-patch-states-for-patch-group ^
    --patch-group "Database Servers"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
   "InstancePatchStates": [
      {
         "BaselineId": "string",
         "FailedCount": number,
         "InstalledCount": number,
         "InstalledOtherCount": number,
         "InstalledPendingRebootCount": number,
         "InstalledRejectedCount": number,
         "InstallOverrideList": "string",
         "InstanceId": "string",
         "LastNoRebootInstallOperationTime": number,
         "MissingCount": number,
         "NotApplicableCount": number,
         "Operation": "string",
         "OperationEndTime": number,
         "OperationStartTime": number,
         "OwnerInformation": "string",
         "PatchGroup": "string",
         "RebootOption": "string",
         "SnapshotId": "string",
         "UnreportedNotApplicableCount": number
      }
   ]
}
```

Beispiele für andere AWS CLI Befehle, die Sie für Ihre Patch Manager Konfigurationsaufgaben verwenden können, finden Sie unterArbeiten mit Patch Manager-Ressourcen unter Verwendung der AWS CLI.

Fehlerbehebung für Patch Manager

Im Folgenden finden Sie Informationen zur Behandlung von Problemen mit Patch Manager, einem Tool in AWS Systems Manager.

Themen

- Problem: Fehler „Invoke-PatchBaselineOperation : Zugriff verweigert“ oder Fehler „Datei kann nicht von S3 heruntergeladen werden“ für baseline_overrides.json
- Problem: Das Patchen schlägt fehl, ohne dass eine offensichtliche Ursache oder Fehlermeldung vorliegt
- Problem: Unerwartete Patch-Compliance-Ergebnisse
- Fehler beim Ausführen von AWS-RunPatchBaseline unter Linux
- Fehler beim Ausführen von AWS-RunPatchBaseline unter Windows Server
- Verwenden von AWS -Support -Automation-Runbooks
- Kontaktaufnahme mit AWS -Support

Problem: Fehler „Invoke-PatchBaselineOperation : Zugriff verweigert“ oder Fehler „Datei kann nicht von S3 heruntergeladen werden“ für **`baseline_overrides.json`**

Problem: Wenn die von Ihrer Patch-Richtlinie festgelegten Patching-Vorgänge ausgeführt werden, erhalten Sie eine Fehlermeldung ähnlich dem folgenden Beispiel.

Example error on Windows Server

```
----------ERROR-------
Invoke-PatchBaselineOperation : Access Denied
At C:\ProgramData\Amazon\SSM\InstanceData\i-02573cafcfEXAMPLE\document\orchestr
ation\792dd5bd-2ad3-4f1e-931d-abEXAMPLE\PatchWindows\_script.ps1:219 char:13
+ $response = Invoke-PatchBaselineOperation -Operation Install -Snapsho ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
+ CategoryInfo : OperationStopped: (Amazon.Patch.Ba...UpdateOpera
tion:InstallWindowsUpdateOperation) [Invoke-PatchBaselineOperation], Amazo
nS3Exception
+ FullyQualifiedErrorId : PatchBaselineOperations,Amazon.Patch.Baseline.Op
erations.PowerShellCmdlets.InvokePatchBaselineOperation
```

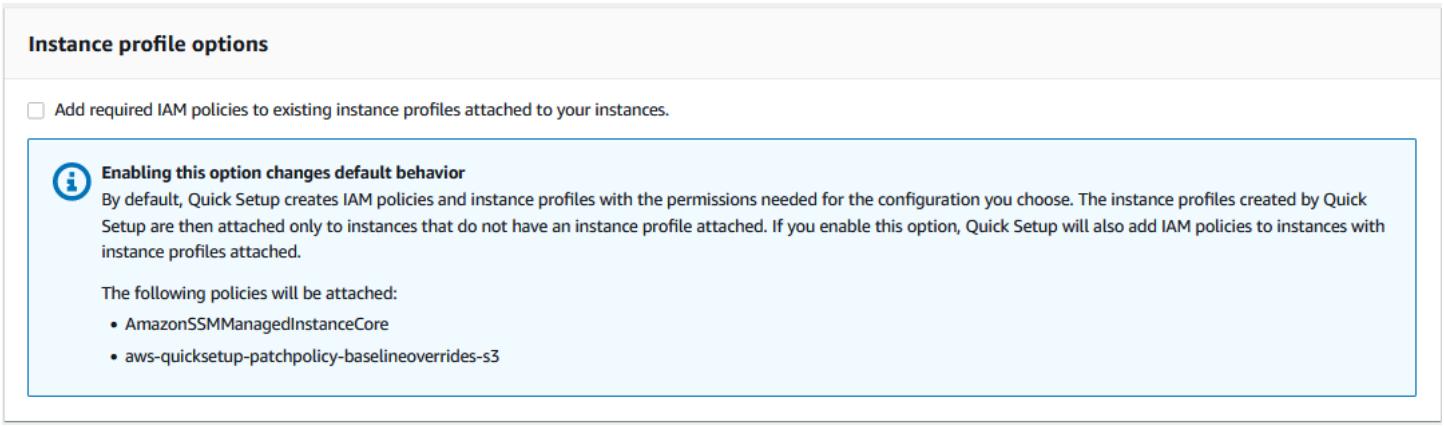
```
failed to run commands: exit status 0xffffffff
```

Example error on Linux

```
[INFO]: Downloading Baseline Override from s3://aws-quicksetup-
patchpolicy-123456789012-abcde/baseline_overrides.json
[ERROR]: Unable to download file from S3: s3://aws-quicksetup-
patchpolicy-123456789012-abcde/baseline_overrides.json.
[ERROR]: Error loading entrance module.
```

Ursache: Sie haben in eine Patch-Richtlinie erstelltQuick Setup, und an einige Ihrer verwalteten Knoten war bereits ein Instanzprofil (für EC2 Instanzen) oder eine Servicerolle (für EC2 Nicht-Computer) angehängt.

Sie haben jedoch das Kontrollkästchen Erforderliche IAM-Richtlinien zu vorhandenen Instance-Profilen hinzufügen, die an Ihre Instances angehängt sind, nicht aktiviert, wie in der folgenden Abbildung dargestellt.



Wenn Sie eine Patch-Richtlinie erstellen, wird auch ein Amazon-S3-Bucket erstellt, in dem die `baseline_overrides.json` Konfigurationsdatei der Richtlinie gespeichert wird. Wenn Sie bei der Erstellung der Richtlinie das Kontrollkästchen Erforderliche IAM-Richtlinien zu bestehenden Instance-Profilen hinzufügen, die mit Ihren Instances verbunden sind, nicht aktivieren, werden die IAM-Richtlinien und Ressourcen-Tags, die für den Zugriff auf `baseline_overrides.json` im S3-Bucket erforderlich sind, nicht automatisch zu Ihren bestehenden IAM-Instance-Profilen und Servicerollen hinzugefügt.

Lösung 1: Löschen Sie die bestehende Patch-Richtlinienkonfiguration und erstellen Sie dann eine neue. Aktivieren Sie dabei das Kontrollkästchen Erforderliche IAM-Richtlinien zu bestehenden Instance-Profilen hinzufügen, die mit Ihren Instances verbunden sind. Diese Auswahl wendet

die mit dieser Quick Setup-Konfiguration erstellten IAM-Richtlinien auf Knoten an, denen bereits ein Instance-Profil oder eine Servicerolle zugewiesen ist. (Quick Setup fügt standardmäßig die erforderlichen Richtlinien zu Instances und Knoten hinzu, die noch nicht über Instance-Profile oder Servicerollen verfügen.) Weitere Informationen finden Sie unter Automatisieren von unternehmensweitem Patching mithilfe einer Quick Setup-Patch-Richtlinie.

Lösung 2: Fügen Sie die erforderlichen Berechtigungen und Tags manuell zu jedem IAM-Instance-Profil und jeder IAM-Servicerolle hinzu, die Sie mit Quick Setup verwenden. Detaillierte Anweisungen finden Sie unter Berechtigungen für den S3-Bucket mit der Patch-Richtlinie.

Problem: Das Patchen schlägt fehl, ohne dass eine offensichtliche Ursache oder Fehlermeldung vorliegt

Problem: Ein Patch-Vorgang schlägt fehl, ohne dass eine Fehlermeldung zurückgegeben wird.

Mögliche Ursache: Wenn mehr als ein Aufruf von `AWS-RunPatchBaseline` gleichzeitig erfolgt, können sie miteinander in Konflikt geraten, sodass Patch-Aufgaben fehlschlagen. Dies wird möglicherweise nicht in den Patchprotokollen angegeben.

Um zu überprüfen, ob sich gleichzeitige Patch-Vorgänge möglicherweise gegenseitig unterbrochen haben, überprüfen Sie den Befehlsverlauf in Run Command, einem Tool in AWS Systems Manager. Prüfen Sie bei einem verwalteten Knoten mit einem Patching-Fehler, ob mehrere Vorgänge innerhalb von 2 Minuten nacheinander versucht haben, die Maschine zu patchen. Dieses Szenario kann manchmal zu einem Fehler führen.

Sie können die AWS Command Line Interface (AWS CLI) auch verwenden, um mithilfe des folgenden Befehls nach gleichzeitigen Patchversuchen zu suchen. Ersetzen Sie den Wert für *`node-id`* durch die ID für Ihren verwalteten Knoten.

```
aws ssm list-commands \
    --filter "key=DocumentName,value=AWS-RunPatchBaseline" \
    --query 'Commands[*].
{CommandId:CommandId,RequestedDateTime:RequestedDateTime,Status:Status}' \
    --instance-id node-id \
    --output table
```

Lösung: Wenn Sie feststellen, dass das Patching aufgrund konkurrierender Patching-Operationen auf demselben verwalteten Knoten fehlgeschlagen ist, passen Sie Ihre Patching-Konfigurationen an, damit dies nicht noch einmal geschieht. Wenn zum Beispiel zwei Wartungsfenster sich überschneidende Patching-Zeiten angeben, entfernen oder ändern Sie eines davon. Wenn in einem

Wartungsfenster eine Patching-Operation angegeben ist, in einer Patch-Richtlinie jedoch eine andere für dieselbe Zeit angegeben ist, sollten Sie die Aufgabe aus dem Wartungsfenster entfernen.

Mögliche Ursache: Beim Patchen verwalteter Knoten kann die Ausführung des Dokuments unterbrochen und als fehlgeschlagen markiert werden, obwohl Patches erfolgreich installiert wurden. Dies kann der Fall sein, wenn das System während des Patchvorgangs einen unerwarteten Neustart einleitet (z. B. um Firmware-Updates oder Funktionen wie) anzuwenden. SecureBoot Der SSM-Agent kann den Status der Dokumentausführung bei externen Neustarts nicht beibehalten und wieder aufnehmen, was dazu führt, dass die Ausführung als fehlgeschlagen gemeldet wird.

Lösung: Um den Status der Patch-Installation nach einer fehlgeschlagenen Ausführung zu überprüfen, führen Sie einen `Scan` Patch-Vorgang aus und überprüfen Sie anschließend die Patch-Kompatibilitätsdaten in Patch Manager, um den aktuellen Konformitätsstatus zu beurteilen.

Wenn Sie feststellen, dass widersprüchliche Patchvorgänge oder externe Neustarts in diesem Szenario nicht die Ursache für den Fehler waren, empfehlen wir Ihnen, sich an uns zu wenden. AWS -Support

Problem: Unerwartete Patch-Compliance-Ergebnisse

Problem: Bei der Überprüfung der nach einem `Scan`-Vorgang generierten Details zur Patching-Compliance enthalten die Ergebnisse Informationen, die nicht die in Ihrer Patch-Baseline festgelegten Regeln widerspiegeln. Beispielsweise wird eine Ausnahme, die Sie der Liste Rejected patches (Abgelehnte Patches) in einer Patch-Baseline hinzugefügt haben, als `Missing` aufgeführt. Oder als `Important` klassifizierte Patches werden als fehlend aufgeführt, obwohl Ihre Patch-Baseline nur `Critical`-Patches angibt.

Ursache: Patch Manager unterstützt derzeit mehrere Methoden zum Ausführen von `Scan`-Operationen:

- Eine in Quick Setup konfigurierte Patch-Richtlinie
- Eine in Quick Setup konfigurierte Host-Management-Option
- Ein Wartungsfenster zum Ausführen eines Patch-`Scan` oder einer `Install`-Aufgabe
- Eine On-Demand Patch now-Operation (Jetzt patchen)

Wenn eine `Scan`-Operation ausgeführt wird, überschreibt dies die Compliance-Details aus dem letzten Scan. Wenn Sie mehr als eine Methode zum Ausführen einer `Scan`-Operation eingerichtet haben und diese unterschiedliche Patch-Baselines mit unterschiedlichen Regeln verwenden, führt dies zu unterschiedlichen Patch-Compliance-Ergebnissen.

Lösung: Um unerwartete Patch-Compliance-Ergebnisse zu vermeiden, empfehlen wir, jeweils nur eine Methode zum Ausführen der Patch Manager `Scan`-Operation zu verwenden. Weitere Informationen finden Sie unter Identifizieren der Ausführung, die Patch-Compliance-Daten erstellt hat.

Fehler beim Ausführen von **`AWS-RunPatchBaseline`** unter Linux

Themen

- Problem: Fehler 'No such file or directory'
- Problem: Fehler 'another process has acquired yum lock'
- Problem: Fehler 'Permission denied / failed to run commands'
- Problem: Fehler 'Unable to download payload'
- Problem: Fehler 'unsupported package manager and python version combination'
- Problem: Patch Manager wendet keine Regeln an, die zum Ausschließen bestimmter Pakete angegeben sind
- Problem: Patching schlägt fehl und Patch Manager meldet, dass die Erweiterung „Servername Indication“ für TLS nicht verfügbar ist
- Problem: Patch Manager meldet 'No more mirrors to try'
- Problem: Patching schlägt fehl mit 'Error code returned from curl is 23'
- Problem: Patching schlägt mit der Meldung 'Error unpacking rpm package...' fehl
- Problem: Patching schlägt fehl mit 'Fehler auf Serviceseite beim Hochladen des Inventars'
- Problem: Das Patchen schlägt fehl und die Meldung „Beim Herunterladen von Paketen sind Fehler aufgetreten“ wird angezeigt
- Problem: Patching schlägt fehl mit der Meldung 'Die folgenden Signaturen konnten nicht verifiziert werden, da der öffentliche Schlüssel nicht verfügbar ist'
- Problem: Das Patchen schlägt fehl und es wird eine Meldung 'NoMoreMirrorsRepoError' angezeigt
- Problem: Das Patchen schlägt mit der Meldung „Payload kann nicht heruntergeladen werden“ fehl
- Problem: Das Patchen schlägt fehl und es wird die Meldung „Installationsfehler: dpkg: Fehler:dpkg-Frontend ist durch einen anderen Prozess gesperrt“ angezeigt
- Problem: Das Patchen auf Ubuntu Server schlägt fehl und es wird der Fehler „dpkg wurde unterbrochen“ angezeigt
- Problem: Das Paketmanager-Dienstprogramm kann eine Paketabhängigkeit nicht auflösen
- Problem: Fehler bei der Abhängigkeit von Zypper-Paketsperren auf verwalteten SLES-Knoten

Problem: Fehler 'No such file or directory'

Problem: Wenn Sie `AWS-RunPatchBaseline` ausführen, schlägt das Patchen mit einem der folgenden Fehler fehl.

```
IOError: [Errno 2] No such file or directory: 'patch-baseline-operations-X.XX.tar.gz'
```

```
Unable to extract tar file: /var/log/amazon/ssm/patch-baseline-operations/patch-
baseline-operations-1.75.tar.gz.failed to run commands: exit status 155
```

```
Unable to load and extract the content of payload, abort.failed to run commands: exit
 status 152
```

Ursache 1: Zwei Befehle zum Ausführen von `AWS-RunPatchBaseline` wurden gleichzeitig auf demselben verwalteten Knoten ausgeführt. Dies erzeugt eine Race-Bedingung, die in der temporären `file patch-baseline-operations*` nicht richtig erstellt oder auf die nicht richtig zugegriffen wird.

Ursache 2: Unzureichender Speicherplatz verbleibt im `/var`-Verzeichnis.

Lösung 1: Stellen Sie sicher, dass kein Wartungsfenster zwei oder mehr Run Command Aufgaben enthält, die `AWS-RunPatchBaseline` mit derselben Prioritätsstufe und auf demselben Ziel ausgeführt werden. IDs Wenn dies der Fall ist, ordnen Sie die Priorität neu an. Run Commandist ein Tool in AWS Systems Manager.

Lösung 2: Stellen Sie sicher, dass jeweils nur ein Wartungsfenster Run Command-Aufgaben ausführt, die `AWS-RunPatchBaseline` auf denselben Zielen und nach demselben Zeitplan verwenden. Ändern Sie in diesem Fall den Zeitplan.

Lösung 3: Stellen Sie sicher, dass nur eine State Manager-Zuordnung `AWS-RunPatchBaseline` nach demselben Zeitplan ausführt und die gleichen verwalteten Knoten anvisiert. State Manager ist ein Tool in AWS Systems Manager.

Lösung 4: Machen Sie genügend Speicherplatz im `/var`-Verzeichnis für die Update-Pakete. frei

Problem: Fehler 'another process has acquired yum lock'

Problem: Wenn Sie `AWS-RunPatchBaseline` ausführen, schlägt das Patchen mit dem folgenden Fehler fehl.

```
12/20/2019 21:41:48 root [INFO]: another process has acquired yum lock, waiting 2 s and
 retry.
```

Ursache: Das `AWS-RunPatchBaseline`-Dokument wurde auf einem verwalteten Knoten ausgeführt, in dem es bereits in einer anderen Operation ausgeführt wird und den `yum`-Paketmanager-Prozess erhalten hat.

Lösung: Stellen Sie sicher, dass keine State Manager-Zuordnung, Aufgaben im Wartungsfenster oder andere Konfigurationen, die `AWS-RunPatchBaseline`nach einem Zeitplan ausführen, ungefähr zur gleichen Zeit denselben verwalteten Knoten als Ziel haben.

Problem: Fehler 'Permission denied / failed to run commands'

Problem: Wenn Sie `AWS-RunPatchBaseline` ausführen, schlägt das Patchen mit dem folgenden Fehler fehl.

```
sh:
/var/lib/amazon/ssm/instanceid/document/orchestration/commandid/PatchLinux/_script.sh:
 Permission denied
failed to run commands: exit status 126
```

Ursache: `/var/lib/amazon/` könnte mit `noexec`-Berechtigungen gemountet sein. Dies ist ein Problem, weil SSM Agent Payload-Skripte auf `/var/lib/amazon/ssm` herunterlädt und sie von diesem Speicherort ausführt.

Lösung: Stellen Sie sicher, dass Sie exklusive Partitionen für `/var/log/amazon` und `/var/lib/amazon` konfiguriert haben und sind mit `exec`-Berechtigungen gemountet sind.

Problem: Fehler 'Unable to download payload'

Problem: Wenn Sie `AWS-RunPatchBaseline` ausführen, schlägt das Patchen mit dem folgenden Fehler fehl.

```
Unable to download payload: https://s3.amzn-s3-demo-bucket.region.amazonaws.com/
aws-ssm-region/patchbaselineoperations/linux/payloads/patch-baseline-operations-
X.XX.tar.gz.failed to run commands: exit status 156
```

Ursache: Der verwaltete Knoten verfügt nicht über die erforderlichen Berechtigungen für den Zugriff auf den angegebenen Amazon Simple Storage Service (Amazon S3)-Bucket.

Lösung: Aktualisieren Sie Ihre Netzwerkkonfiguration so, dass S3-Endpunkte erreichbar sind. Weitere Informationen finden Sie unter den Informationen zum erforderlichen Zugriff auf S3-Buckets für Patch Manager in SSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets.

Problem: Fehler 'unsupported package manager and python version combination'

Problem: Wenn Sie `AWS-RunPatchBaseline` ausführen, schlägt das Patchen mit dem folgenden Fehler fehl.

```
An unsupported package manager and python version combination was found. Apt requires
 Python3 to be installed.
failed to run commands: exit status 1
```

Ursache: Eine unterstützte Version von Python3 ist nicht auf der Instance Debian Server oder Ubuntu Server installiert.

Lösung: Installieren Sie eine unterstützte Version von python3 (3.0–3.12) auf dem Server, die für verwaltete Debian Server- und Ubuntu Server-Knoten erforderlich ist.

Problem: Patch Manager wendet keine Regeln an, die zum Ausschließen bestimmter Pakete angegeben sind

Problem: Sie haben versucht, bestimmte Pakete auszuschließen, indem Sie sie in der `/etc/yum.conf`-Datei im Format `exclude=`*`package-name`* angeben, aber sie werden nicht während der Patch Manager-Operation `Install` ausgeschlossen.

Ursache: Patch Manager enthält keine Ausschlüsse, die in der `/etc/yum.conf`-Datei angegeben sind.

Lösung: Um bestimmte Pakete auszuschließen, erstellen Sie eine benutzerdefinierte Patch-Baseline und eine Regel, um die Pakete auszuschließen, die nicht installiert werden sollen.

Problem: Patching schlägt fehl und Patch Manager meldet, dass die Erweiterung „Servername Indication“ für TLS nicht verfügbar ist

Problem: Der Patchvorgang gibt die folgende Meldung aus.

```
/var/log/amazon/ssm/patch-baseline-operations/urllib3/util/ssl_.py:369:
SNIMissingWarning: An HTTPS request has been made, but the SNI (Server Name Indication)
 extension
to TLS is not available on this platform. This might cause the server to present an
 incorrect TLS
```

```
certificate, which can cause validation failures. You can upgrade to a newer version of
 Python
to solve this.
For more information, see https://urllib3.readthedocs.io/en/latest/advanced-
usage.html#ssl-warnings
```

Ursache: Diese Meldung zeigt keinen Fehler an. Stattdessen ist dies eine Warnung, dass die ältere Version von Python, die mit dem Betriebssystem vertrieben wird, TLS Server Name Indication nicht unterstützt. Das Systems Manager Manager-Patch-Payload-Skript gibt diese Warnung aus, wenn eine Verbindung zu AWS APIs diesem unterstützenden SNI hergestellt wird.

Lösung: Um Patching-Fehler zu beheben, wenn diese Meldung gemeldet wird, überprüfen Sie den Inhalt der `stdout`- und `stderr`-Dateien. Wenn Sie die Patch-Baseline nicht so konfiguriert haben, dass diese Dateien in einem S3-Bucket oder in Amazon CloudWatch Logs gespeichert werden, können Sie die Dateien am folgenden Speicherort auf Ihrem verwalteten Linux-Node finden.

`/var/lib/amazon/ssm/`*`instance-id`*`/document/orchestration/`*`Run-Command-execution-id`*`/awsrunShellScript/PatchLinux`

Problem: Patch Manager meldet 'No more mirrors to try'

Problem: Der Patchvorgang gibt die folgende Meldung aus.

```
[Errno 256] No more mirrors to try.
```

Ursache: Die auf dem verwalteten Knoten konfigurierten Repositorys funktionieren nicht richtig. Mögliche Gründe hierfür sind:

- Das yum-Cache ist beschädigt.
- Eine Repository-URL kann aufgrund von Netzwerkproblemen nicht erreicht werden.

Lösung: Patch Manager verwendet den Standard-Paketmanager des verwalteten Knoten, um die Patching-Operation durchzuführen. Überprüfen Sie, ob Repositorys richtig konfiguriert sind und funktionieren.

Problem: Patching schlägt fehl mit 'Error code returned from curl is 23'

Problem: Eine Patching-Operation, die `AWS-RunPatchBaseline` verwendet, schlägt mit einer Fehlermeldung ähnlich der folgenden fehl:

```
05/01/2025 17:04:30 root [ERROR]: Error code returned from curl is 23
```

Ursache: Das auf Ihren Systemen verwendete Curl-Tool verfügt nicht über die erforderlichen Rechte, um in das Dateisystem zu schreiben. Dies kann vorkommen, wenn das Standard-Curl-Tool des Paketmanagers durch eine andere Version ersetzt wurde, beispielsweise durch eine, die mit snap installiert wurde.

Lösung: Wenn die vom Paketmanager bereitgestellte curl-Version deinstalliert wurde, während eine andere Version installiert wurde, installieren Sie sie erneut.

Wenn Sie mehrere curl-Versionen installiert halten müssen, stellen Sie sicher, dass sich die mit dem Paketmanager verknüpfte Version im ersten in der `PATH`-Variable aufgeführten Verzeichnis befindet. Sie können dies überprüfen, indem Sie den Befehl `echo $PATH` ausführen, um die aktuelle Reihenfolge der Verzeichnisse zu sehen, die auf Ihrem System auf ausführbare Dateien überprüft werden.

Problem: Patching schlägt mit der Meldung 'Error unpacking rpm package...' fehl

Problem: Ein Patch-Vorgang schlägt mit einem Fehler ähnlich dem folgenden fehl:

```
Error : Error unpacking rpm package python-urllib3-1.25.9-1.amzn2.0.2.noarch
python-urllib3-1.25.9-1.amzn2.0.1.noarch was supposed to be removed but is not!
failed to run commands: exit status 1
```

Ursache 1: Wenn ein bestimmtes Paket in mehreren Paket-Installationsprogrammen vorhanden ist, z. B. sowohl in `pip` als auch in `yum` oder `dnf`, kann es bei der Verwendung des Standard-Paketmanagers zu Konflikten kommen.

Ein häufiges Beispiel ist das `urllib3`-Paket, das sich in `pip`, `yum` und `dnf` befindet.

Ursache 2: Das `python-urllib3`-Paket ist beschädigt. Dies kann passieren, wenn die Paketdateien von `pip` installiert oder aktualisiert wurden, nachdem das `rpm`-Paket zuvor von `yum` oder `dnf` installiert wurde.

Lösung: Entfernen Sie das `python-urllib3`-Paket aus Pip, indem Sie den Befehl `sudo pip uninstall urllib3` ausführen, und behalten Sie das Paket nur im Standard-Paketmanager (`yum` oder `dnf`) bei.

Problem: Patching schlägt fehl mit 'Fehler auf Serviceseite beim Hochladen des Inventars'

Problem: Beim Ausführen des `AWS-RunPatchBaseline`-Dokuments erhalten Sie die folgende Fehlermeldung:

```
Encounter service side error when uploading the inventory
```

Ursache: Zwei Befehle zum Ausführen von `AWS-RunPatchBaseline` wurden gleichzeitig auf demselben verwalteten Knoten ausgeführt. Dies führt zu einer Race-Bedingung, wenn der Boto3-Client während Patch-Vorgängen initialisiert wird.

Lösung: Stellen Sie sicher, dass keine State Manager-Zuordnung, Aufgaben im Wartungsfenster oder andere Konfigurationen, die `AWS-RunPatchBaseline`nach einem Zeitplan ausführen, ungefähr zur gleichen Zeit denselben verwalteten Knoten als Ziel haben.

Problem: Das Patchen schlägt fehl und die Meldung „Beim Herunterladen von Paketen sind Fehler aufgetreten“ wird angezeigt

Problem: Beim Patchen erhalten Sie eine Fehlermeldung, die der folgenden ähnelt:

```
YumDownloadError: [u'Errors were encountered while downloading
packages.', u'libxml2-2.9.1-6.el7_9.6.x86_64: [Errno 5] [Errno 12]
Cannot allocate memory', u'libxslt-1.1.28-6.el7.x86_64: [Errno 5]
[Errno 12] Cannot allocate memory', u'libcroco-0.6.12-6.el7_9.x86_64:
[Errno 5] [Errno 12] Cannot allocate memory', u'openldap-2.4.44-25.el7_9.x86_64:
[Errno 5] [Errno 12] Cannot allocate memory',
```

Ursache: Dieser Fehler kann auftreten, wenn auf einem verwalteten Knoten nicht genügend Speicher verfügbar ist.

Lösung: Konfigurieren Sie den Swap-Speicher oder aktualisieren Sie die Instance auf einen anderen Typ, um die Speicherunterstützung zu erhöhen. Starten Sie dann einen neuen Patch-Vorgang.

Problem: Patching schlägt fehl mit der Meldung 'Die folgenden Signaturen konnten nicht verifiziert werden, da der öffentliche Schlüssel nicht verfügbar ist'

Problem: Das Patchen schlägt bei Ubuntu Server mit einer Fehlermeldung ähnlich der folgenden fehl:

```
02/17/2022 21:08:43 root [ERROR]: W:GPG error:
http://repo.mysql.com/apt/ubuntu  bionic InRelease: The following
```

```
signatures couldn't be verified because the public key is not available:
NO_PUBKEY 467B942D3A79BD29, E:The repository ' http://repo.mysql.com/apt/ubuntu bionic
```

Ursache: Der Schlüssel für GNU Privacy Guard (GPG) ist abgelaufen oder fehlt.

Lösung: Aktualisieren Sie den GPG-Schlüssel, oder fügen Sie den Schlüssel erneut hinzu.

Anhand des zuvor gezeigten Fehlers sehen wir zum Beispiel, dass der 467B942D3A79BD29-Schlüssel fehlt und hinzugefügt werden muss. Führen Sie dazu einen der folgenden Befehle aus:

```
sudo apt-key adv --keyserver hkps://keyserver.ubuntu.com --recv-keys 467B942D3A79BD29
```

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 467B942D3A79BD29
```

Oder, um alle Schlüssel zu aktualisieren, führen Sie den folgenden Befehl aus:

```
sudo apt-key adv --keyserver hkps://keyserver.ubuntu.com --refresh-keys
```

Wenn der Fehler danach weiterhin auftritt, empfehlen wir, das Problem an die Organisation zu melden, die das Repository verwaltet. Bis ein Fix verfügbar ist, können Sie die `/etc/apt/sources.list`-Datei so bearbeiten, dass das Repository während des Patchvorgangs ausgelassen wird.

Öffnen Sie dazu die `sources.list`-Datei zur Bearbeitung, suchen Sie die Zeile für das Repository und fügen Sie am Anfang der Zeile ein #-Zeichen ein, um sie auszukommentieren. Speichern und schließen Sie dann die Datei.

Problem: Das Patchen schlägt fehl und es wird eine Meldung 'NoMoreMirrorsRepoError' angezeigt

Problem: Sie erhalten eine Fehlermeldung ähnlich der folgenden:

```
NoMoreMirrorsRepoError: failure: repodata/repomd.xml from pgdg94: [Errno 256] No more
 mirrors to try.
```

Ursache: Im Quell-Repository ist ein Fehler aufgetreten.

Lösung: Wir empfehlen, das Problem der Organisation zu melden, die das Repository verwaltet. Bis der Fehler behoben ist, können Sie das Repository auf Betriebssystemebene deaktivieren.

Führen Sie dazu den folgenden Befehl aus und ersetzen Sie den Wert für *repo-name* durch Ihren Repository-Namen:

```
yum-config-manager --disable repo-name
```

Im Folgenden sehen Sie ein Beispiel.

```
yum-config-manager --disable pgdg94
```

Nachdem Sie diesen Befehl ausgeführt haben, führen Sie einen weiteren Patch-Vorgang aus.

Problem: Das Patchen schlägt mit der Meldung „Payload kann nicht heruntergeladen werden“ fehl

Problem: Sie erhalten eine Fehlermeldung ähnlich der folgenden:

```
Unable to download payload:
https://s3.dualstack.eu-west-1.amazonaws.com/aws-ssm-eu-west-1/patchbaselineoperations/
linux/payloads/patch-baseline-operations-1.83.tar.gz.
failed  to run commands: exit status 156
```

Ursache: Die Konfiguration des verwalteten Knotens ist fehlerhaft oder unvollständig.

Lösung: Versichern Sie sich, dass der verwaltete Knoten wie folgt konfiguriert ist:

- Ausgehende TCP-443-Regel in der Sicherheitsgruppe.
- Ausgehende TCP-443-Regel in NACL.
- TCP-Regel 1024-65535 für eingehenden Datenverkehr in NACL.
- NAT/IGW in der Routing-Tabelle, um Konnektivität zu einem S3-Endpunkt bereitzustellen. Wenn die Instance keinen Internetzugang hat, stellen Sie ihr Konnektivität mit dem S3-Endpunkt zur Verfügung. Fügen Sie dazu einen S3-Gateway-Endpunkt in der VPC hinzu und integrieren Sie ihn in die Routing-Tabelle des verwalteten Knotens.

Problem: Das Patchen schlägt fehl und es wird die Meldung „Installationsfehler: dpkg: Fehler:dpkg-Frontend ist durch einen anderen Prozess gesperrt“ angezeigt

Problem: Das Patchen schlägt mit einem Fehler ähnlich dem folgenden fehl:

```
install errors: dpkg: error: dpkg frontend is locked by another process
failed to run commands: exit status 2
```

```
Failed to install package; install status Failed
```

Ursache: Der Paketmanager führt bereits einen anderen Prozess auf einem verwalteten Knoten auf Betriebssystemebene aus. Wenn der Abschluss dieses anderen Prozesses viel Zeit in Anspruch nimmt, kann es bei der Patch-Operation von Patch Manager zu einem Timeout kommen und ein Fehler auftreten.

Lösung: Führen Sie nach Abschluss des anderen Prozesses, der den Paketmanager verwendet, einen neuen Patchvorgang aus.

Problem: Das Patchen auf Ubuntu Server schlägt fehl und es wird der Fehler „dpkg wurde unterbrochen“ angezeigt

Problem: Auf Ubuntu Server schlägt das Patchen mit einem Fehler ähnlich dem folgenden fehl:

```
E: dpkg was interrupted, you must manually run
'dpkg --configure -a' to correct the problem.
```

Ursache: Ein oder mehrere Pakete sind falsch konfiguriert.

Lösung: Führen Sie die folgenden Schritte aus:

1. Prüfen Sie, welche Pakete betroffen sind und welche Probleme bei den einzelnen Paketen bestehen, indem Sie nacheinander die folgenden Befehle ausführen:

   ```
   sudo apt-get check
   ```

   ```
   sudo dpkg -C
   ```

   ```
   dpkg-query -W -f='${db:Status-Abbrev} ${binary:Package}\n' | grep -E ^.[^nci]
   ```

2. Korrigieren Sie die fehlerhaften Pakete, indem Sie den folgenden Befehl ausführen:

   ```
   sudo dpkg --configure -a
   ```

3. Wenn der vorherige Befehl das Problem nicht vollständig behoben hat, führen Sie den folgenden Befehl aus:

   ```
   sudo apt --fix-broken install
   ```

Problem: Das Paketmanager-Dienstprogramm kann eine Paketabhängigkeit nicht auflösen

Problem: Der native Paketmanager auf dem verwalteten Knoten kann eine Paketabhängigkeit nicht auflösen und das Patchen schlägt fehl. Das folgende Beispiel für eine Fehlermeldung weist auf diese Art von Fehler auf einem Betriebssystem hin, das `yum` als Paketmanager verwendet.

```
09/22/2020 08:56:09 root [ERROR]: yum update failed with result code: 1,
message: [u'rpm-python-4.11.3-25.amzn2.0.3.x86_64 requires rpm = 4.11.3-25.amzn2.0.3',
u'awscli-1.18.107-1.amzn2.0.1.noarch requires python2-botocore = 1.17.31']
```

Ursache: Patch Manager verwendet auf Linux-Betriebssystemen den systemeigenen Paketmanager auf der Maschine, um Patch-Operationen wie `yum`, `dnf`, `apt` und `zypper` auszuführen. Die Anwendungen erkennen, installieren, aktualisieren oder entfernen abhängige Pakete bei Bedarf automatisch. Einige Bedingungen können jedoch dazu führen, dass der Paketmanager einen Abhängigkeitsvorgang nicht abschließen kann, wie zum Beispiel:

- Auf dem Betriebssystem sind mehrere widersprüchliche Repositorys konfiguriert.
- Auf eine Remote-Repository-URL kann aufgrund von Netzwerkproblemen nicht zugegriffen werden.
- Im Repository wurde ein Paket für die falsche Architektur gefunden.

Lösung: Das Patchen kann aufgrund eines Abhängigkeitsproblems aus einer Vielzahl von Gründen fehlschlagen. Wir empfehlen Ihnen daher, sich an uns AWS -Support zu wenden, um Ihnen bei der Problembehebung behilflich zu sein.

Problem: Fehler bei der Abhängigkeit von Zypper-Paketsperren auf verwalteten SLES-Knoten

Problem: Wenn Sie `AWS-RunPatchBaseline` mit dem `Install`-Vorgang auf SUSE Linux Enterprise Server-Instances ausführen, schlägt das Patchen fehl und es treten Fehler bei der Abhängigkeitsprüfung auf, die mit Paketsperren zusammenhängen. Sie könnten Fehlermeldungen ähnlich der folgenden erhalten:

```
Problem: mock-pkg-has-dependencies-0.2.0-21.adistro.noarch requires mock-pkg-standalone
 = 0.2.0, but this requirement cannot be provided
  uninstallable providers: mock-pkg-standalone-0.2.0-21.adistro.noarch[local-repo]
 Solution 1: remove lock to allow installation of mock-pkg-
standalone-0.2.0-21.adistro.noarch[local-repo]
 Solution 2: do not install mock-pkg-has-dependencies-0.2.0-21.adistro.noarch
```

```
 Solution 3: break mock-pkg-has-dependencies-0.2.0-21.adistro.noarch by ignoring some
 of its dependencies

Choose from above solutions by number or cancel [1/2/3/c] (c): c
```

In diesem Beispiel ist das Paket `mock-pkg-standalone` gesperrt, was Sie überprüfen könnten, indem Sie `sudo zypper locks` ausführen und in der Ausgabe nach diesem Paketnamen suchen.

Oder Sie sehen möglicherweise Protokolleinträge, die auf Fehlschläge bei der Abhängigkeitsprüfung hinweisen:

```
Encountered a known exception in the CLI Invoker:
 CLIInvokerError(error_message='Dependency check failure during commit process',
 error_code='4')
```

> **Note**
>
> Dieses Problem tritt nur während `Install`-Vorgängen auf. `Scan`-Vorgänge wenden keine Paketsperren an und sind auch nicht von vorhandenen Sperren betroffen.“

Ursache: Dieser Fehler tritt auf, wenn Zypper-Paketsperren die Installation oder Aktualisierung von Paketen aufgrund von Abhängigkeitskonflikten verhindern. Paketsperren können aus verschiedenen Gründen vorhanden sein:

- Vom Kunden angewendete Sperren: Sie oder Ihr Systemadministrator haben Pakete manuell mithilfe von Zypper-Befehlen gesperrt, wie z. B. `zypper addlock`.
- Von Patch Manager abgelehnte Patches: Patch Manager wendet automatisch Paketsperren an, wenn Sie Pakete in der Liste der abgelehnten Patches Ihrer Patch-Baseline angeben, um deren Installation zu verhindern.
- Restsperren aufgrund unterbrochener Vorgänge: In seltenen Fällen, in denen ein Patch-Vorgang unterbrochen wurde (z. B. durch einen Systemneustart), konnte zuvor Patch Manager temporäre Sperren aufheben, restliche Paketsperren verblieben jedoch eventuell auf Ihrem verwalteten Knoten.

Lösung: Gehen Sie je nach Ursache folgendermaßen vor, um Probleme mit der Zypper-Paketsperre zu beheben:

Schritt 1: Identifizieren gesperrter Pakete

Verbinden Sie sich mit Ihrem verwalteten SLES-Knoten und führen Sie den folgenden Befehl aus, um alle derzeit gesperrten Pakete aufzulisten:

```
sudo zypper locks
```

Schritt 2: Ermitteln der Quelle der Sperren

- Wenn es sich bei den gesperrten Paketen um Pakete handelt, die Sie aus Gründen der Systemstabilität absichtlich gesperrt haben, sollten Sie überlegen, ob sie gesperrt bleiben müssen oder ob sie für das Patching vorübergehend entsperrt werden können.
- Wenn die gesperrten Pakete mit Einträgen in der Liste der abgelehnten Patches Ihrer Patch-Baseline übereinstimmen, handelt es sich wahrscheinlich um Restsperren, die auf einen unterbrochenen Patch-Vorgang zurückzuführen sind. Patch Manager wendet diese Sperren während des normalen Betriebs vorübergehend an und entfernt sie automatisch, wenn der Vorgang abgeschlossen ist. Sie können die Pakete entweder aus der Liste der abgelehnten Pakete entfernen oder Ihre Patch-Baseline-Regeln ändern.
- Wenn Sie die gesperrten Pakete nicht erkennen und sie nicht absichtlich gesperrt wurden, handelt es sich möglicherweise um Restsperren aus einem zuvor unterbrochenen Patch-Vorgang.

Schritt 3: Entfernen von Sperren nach Notwendigkeit

Mit diesem Befehl können Sie bestimmte Paketsperren entfernen:

```
sudo zypper removelock package-name
```

Um alle Paketsperren zu entfernen (mit Vorsicht verwenden), führen Sie folgenden Befehl aus:

```
sudo zypper cleanlocks
```

Schritt 4: Aktualisieren Ihrer Patch-Baseline (falls zutreffend)

Wenn die Sperren durch abgelehnte Patches in Ihrer Patch-Baseline verursacht wurden:

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Patch Manager aus.

3. Wählen Sie die Registerkarte Patch-Baselines und dann Ihre benutzerdefinierte Patch-Baseline aus.
4. Wählen Sie Aktionen, Patch-Baseline ändern aus.
5. Überprüfen Sie im Abschnitt Abgelehnte Patches die aufgelisteten Pakete und entfernen Sie alle Pakete, deren Installation zulässig sein sollte.
6. Wählen Sie Änderungen speichern aus.

Schritt 5: Wiederholen des Patch-Vorgangs

Nachdem Sie die problematischen Sperren entfernt und gegebenenfalls Ihre Patch-Baseline aktualisiert haben, führen Sie das `AWS-RunPatchBaseline`-Dokument erneut aus.

> **Note**
>
> Wenn während der `Install`-Vorgänge Patch Manager Sperren für abgelehnte Patches anwendet, ist es so konzipiert, dass diese Sperren nach Abschluss des Patch-Vorgangs automatisch aufgehoben werden. Wenn Sie diese Sperren bei der Ausführung von `sudo zypper locks` sehen, bedeutet das, dass ein früherer Patch-Vorgang unterbrochen wurde, bevor die Aufhebung durchgeführt werden konnte. Wenn ein Patch-Vorgang jedoch unterbrochen wird, ist möglicherweise eine manuelle Aufhebung erforderlich, wie in diesem Verfahren beschrieben.

Prävention: Vermeiden Sie zukünftige Zypper-Sperrkonflikte wie folgt:

- Prüfen Sie die Liste der abgelehnten Patches Ihrer Patch-Baseline sorgfältig, um sicherzustellen, dass sie nur Pakete enthält, die Sie wirklich ausschließen möchten.
- Vermeiden Sie es, Pakete, die möglicherweise als Abhängigkeiten für Sicherheitsupdates benötigt werden, manuell zu sperren.
- Wenn Sie Pakete manuell sperren müssen, dokumentieren Sie die Gründe dafür und überprüfen Sie die Sperren regelmäßig.
- Stellen Sie sicher, dass die Patch-Vorgänge erfolgreich abgeschlossen werden und nicht durch Systemneustarts oder andere Faktoren unterbrochen werden.
- Überwachen Sie Patch-Vorgänge bis zum Abschluss und vermeiden Sie es, sie durch Systemneustarts oder andere Aktionen zu unterbrechen, die das ordnungsgemäße Löschen temporärer Sperren verhindern könnten.

Fehler beim Ausführen von **`AWS-RunPatchBaseline`** unter Windows Server

Themen

- Problem: Nicht übereinstimmende Produktpaare family/product
- Problem: AWS-RunPatchBaseline-Ausgabe gibt einen HRESULT (Windows Server) zurück
- Problem: Der verwaltete Knoten hat keinen Zugriff auf Windows Update Catalog oder WSUS
- Problem: PatchBaselineOperations PowerShell Das Modul kann nicht heruntergeladen werden
- Problem: fehlende Patches

Problem: Nicht übereinstimmende Produktpaare family/product

Problem: Wenn Sie eine Patch-Baseline in der Systems Manager-Konsole erstellen, geben Sie eine Produktfamilie und ein Produkt an. Beispiel:

- Product Family (Produktfamilie): `Office`

  Produkt: `Office 2016`

Ursache: Wenn Sie versuchen, eine Patch-Baseline mit einem nicht übereinstimmenden family/product Produktpaar zu erstellen, wird eine Fehlermeldung angezeigt. Dies kann folgende Ursachen haben:

- Sie haben eine gültige Kombination aus Produktfamilie und Produktpaar ausgewählt, dann jedoch die Auswahl der Produktfamilie entfernt.
- Sie haben ein Produkt aus der Unterliste Obsolete or mismatched options (Veraltete oder nicht übereinstimmende Optionen) statt aus der Unterliste Available and matching options (Verfügbare und übereinstimmende Optionen) ausgewählt.

  Artikel in der Produktunterliste Veraltete oder nicht übereinstimmende Optionen wurden möglicherweise fälschlicherweise über ein SDK oder AWS Command Line Interface den Befehl () eingegeben.AWS CLI`create-patch-baseline` Dadurch kann es zu einem Schreibfehler oder einer falschen Zuordnung eines Produkts zu einer Produktfamilie kommen. Ein Produkt kann auch in der Unterliste Obsolete or mismatched options (Veraltete oder nicht übereinstimmende Optionen) enthalten sein, wenn es für eine vorherige Patch-Baseline angegeben wurde, aber keine Patches für das Produkt von Microsoft verfügbar sind.

Lösung: Um dieses Problem in der Konsole zu vermeiden, wählen Sie immer Optionen aus den Unterlisten Currently available options (Derzeit verfügbare Optionen) aus.

Um diejenigen Produkte anzuzeigen, für die Patches verfügbar sind, können Sie auch den Befehl `describe-patch-properties` in der AWS CLI oder den API-Befehl `DescribePatchProperties` verwenden.

Problem: **`AWS-RunPatchBaseline`**-Ausgabe gibt einen **`HRESULT`** (Windows Server) zurück

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
----------ERROR-------
Invoke-PatchBaselineOperation : Exception Details: An error occurred when
attempting to search Windows Update.
Exception Level 1:
 Error Message: Exception from HRESULT: 0x80240437
 Stack Trace: at WUApiLib.IUpdateSearcher.Search(String criteria)..
(Windows updates)
11/22/2020 09:17:30 UTC | Info | Searching for Windows Updates.
11/22/2020 09:18:59 UTC | Error | Searching for updates resulted in error: Exception
 from HRESULT: 0x80240437
----------ERROR-------
failed to run commands: exit status 4294967295
```

Ursache: Diese Ausgabe weist darauf hin, dass das systemeigene Windows Update die Patchvorgänge nicht ausführen APIs konnte.

Lösung: Überprüfen Sie den `HResult`-Code in den folgenden Themen auf microsoft.com, um Schritte zur Fehlerbehebung zum Beheben des Fehlers zu ermitteln:

- Windows-Update-Fehlercodes nach Komponenten
- Häufige Fehler und Abhilfemaßnahmen für Windows Update

Problem: Der verwaltete Knoten hat keinen Zugriff auf Windows Update Catalog oder WSUS

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
Downloading PatchBaselineOperations PowerShell module from https://s3.aws-api-
domain/path_to_module.zip to C:\Windows\TEMP\Amazon.PatchBaselineOperations-1.29.zip.

Extracting PatchBaselineOperations zip file contents to temporary folder.
```

```
Verifying SHA 256 of the PatchBaselineOperations PowerShell module files.

Successfully downloaded and installed the PatchBaselineOperations PowerShell module.

Patch Summary for

PatchGroup :

BaselineId :

Baseline : null

SnapshotId :

RebootOption : RebootIfNeeded

OwnerInformation :

OperationType : Scan

OperationStartTime : 1970-01-01T00:00:00.0000000Z

OperationEndTime : 1970-01-01T00:00:00.0000000Z

InstalledCount : -1

InstalledRejectedCount : -1

InstalledPendingRebootCount : -1

InstalledOtherCount : -1

FailedCount : -1

MissingCount : -1

NotApplicableCount : -1

UnreportedNotApplicableCount : -1

EC2AMAZ-VL3099P - PatchBaselineOperations Assessment Results - 2020-12-30T20:59:46.169

----------ERROR-------
```

```
Invoke-PatchBaselineOperation : Exception Details: An error occurred when attempting to
 search Windows Update.

Exception Level 1:

Error Message: Exception from HRESULT: 0x80072EE2

Stack Trace: at WUApiLib.IUpdateSearcher.Search(String criteria)

at
 Amazon.Patch.Baseline.Operations.PatchNow.Implementations.WindowsUpdateAgent.SearchForUpdates(

searchCriteria)

At C:\ProgramData\Amazon\SSM\InstanceData\i-02573cafcfEXAMPLE\document\orchestration
\3d2d4864-04b7-4316-84fe-eafff1ea58

e3\PatchWindows\_script.ps1:230 char:13

+ $response = Invoke-PatchBaselineOperation -Operation Install -Snapsho ...

+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

+ CategoryInfo : OperationStopped:
 (Amazon.Patch.Ba...UpdateOperation:InstallWindowsUpdateOperation) [Inv

oke-PatchBaselineOperation], Exception

+ FullyQualifiedErrorId : Exception Level 1:

Error Message: Exception Details: An error occurred when attempting to search Windows
 Update.

Exception Level 1:

Error Message: Exception from HRESULT: 0x80072EE2

Stack Trace: at WUApiLib.IUpdateSearcher.Search(String criteria)

at
 Amazon.Patch.Baseline.Operations.PatchNow.Implementations.WindowsUpdateAgent.SearchForUpdates(
 searc
```

```
---Error truncated----
```

Ursache: Dieser Fehler kann mit den Windows Update-Komponenten oder einer fehlenden Konnektivität zum Windows Update Catalog oder Windows Server Update Services (WSUS) zusammenhängen.

Lösung: Bestätigen Sie, dass der verwaltete Knoten über ein Internet-Gateway, ein NAT-Gateway oder eine NAT-Instance eine Verbindung zum Microsoft Update Catalog hergestellt hat. Wenn Sie WSUS verwenden, bestätigen Sie, dass der verwaltete Knoten eine Verbindung zum WSUS-Server in Ihrer Umgebung hat. Wenn Konnektivität für das beabsichtigte Ziel verfügbar ist, überprüfen Sie die Microsoft-Dokumentation auf andere mögliche Ursachen für `HResult 0x80072EE2`. Dies kann auf ein Problem auf Betriebssystemebene hinweisen.

Problem: PatchBaselineOperations PowerShell Das Modul kann nicht heruntergeladen werden

Problem: Sie haben eine Fehlermeldung wie die folgende erhalten.

```
Preparing to download PatchBaselineOperations PowerShell module from S3.

Downloading PatchBaselineOperations PowerShell module from https://s3.aws-api-
domain/path_to_module.zip to C:\Windows\TEMP\Amazon.PatchBaselineOperations-1.29.zip.
----------ERROR-------

C:\ProgramData\Amazon\SSM\InstanceData\i-02573cafcfEXAMPLE\document\orchestration
\aaaaaaaa-bbbb-cccc-dddd-4f6ed6bd5514\

PatchWindows\_script.ps1 : An error occurred when executing PatchBaselineOperations:
 Unable to connect to the remote server

+ CategoryInfo : NotSpecified: (:) [Write-Error], WriteErrorException

+ FullyQualifiedErrorId : Microsoft.PowerShell.Commands.WriteErrorException,_script.ps1

failed to run commands: exit status 4294967295
```

Lösung: Überprüfen Sie die Konnektivität und Berechtigungen für Amazon Simple Storage Service (Amazon S3) des verwalteten Knoten. Für die Rolle des verwalteten Knotens AWS Identity and Access Management (IAM) müssen die unter angegebenen Mindestberechtigungen verwendet werden. SSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets Der Knoten muss über den Amazon-S3-Gateway-Endpunkt, das NAT-Gateway oder das Internet-Gateway mit dem Amazon-S3-Endpunkt kommunizieren. Weitere Informationen zu den VPC-Endpunktanforderungen für AWS

Systems Manager SSM Agent (SSM Agent) finden Sie unterVerbessern Sie die Sicherheit von EC2 Instanzen mithilfe von VPC-Endpunkten für Systems Manager.

Problem: fehlende Patches

Problem: `AWS-RunPatchbaseline` wurde erfolgreich abgeschlossen, aber es fehlen einige Patches.

Nachfolgend finden Sie einige häufige Auslöser und deren Lösungen.

Ursache 1: Die Baseline ist nicht effektiv.

Lösung 1: Führen Sie die folgenden Schritte aus, um zu überprüfen, ob dies die Ursache ist.

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie die Registerkarte Befehlsverlauf und dann den Befehl aus, dessen Baseline Sie überprüfen möchten.
4. Wählen Sie den verwalteten Knoten aus, dem Patches fehlen.
5. Wählen Sie Schritt 1 – Ausgabe aus und finden Sie den `BaselineId`-Wert.
6. Aktivieren Sie die zugewiesene Patch-Baseline-Konfiguration, d. h. Betriebssystem, Produktname, Klassifizierung und Schweregrad für die Patch-Baseline.
7. Rufen Sie den Microsoft Update Catalog auf.
8. Suchen Sie im Artikel der Microsoft Knowledge Base IDs (KB) (z. B. KB3216916).
9. Stellen Sie sicher, dass der Wert unter Product (Produkt) dem Ihres verwalteten Knotens entspricht, und wählen Sie den entsprechenden Title (Titel) aus. Ein neues Fenser Details aktualisieren wird geöffnet.
10. In der Registerkarte Übersicht müssen Klassifizierung und Schweregrad des MSRC der Patch-Baseline-Konfiguration entsprechen, die Sie zuvor gefunden haben.

Ursache 2: Das Patch wurde ersetzt.

Lösung 2: Führen Sie die folgenden Schritte aus, um zu überprüfen, ob dies der Fall ist.

1. Rufen Sie den Microsoft Update Catalog auf.
2. Suchen Sie im Artikel der Microsoft Knowledge Base IDs (KB) (z. B. KB3216916).

3. Stellen Sie sicher, dass der Wert unter Product (Produkt) dem Ihres verwalteten Knotens entspricht, und wählen Sie den entsprechenden Title (Titel) aus. Ein neues Fenser Details aktualisieren wird geöffnet.
4. Gehen Sie zur Registerkarte Paketdetails. Suchen Sie nach einem Eintrag unter dem Header Dieses Update wurde durch die folgenden Updates ersetzt:.

Ursache 3: Dasselbe Patch hat möglicherweise unterschiedliche KB-Nummern, da die WSUS- und Windows-Online-Updates von Microsoft als unabhängige Versionskanäle behandelt werden.

Lösung 3: Überprüfen Sie die Berechtigung des Patches. Wenn das Paket unter WSUS nicht verfügbar ist, installieren Sie OS Build 14393.3115. Wenn das Paket für alle Betriebssystem-Builds verfügbar ist, installieren Sie OS-Builds 18362.1256 und 18363.1256.

Verwenden von AWS -Support -Automation-Runbooks

AWS -Support stellt zwei Automation-Runbooks zur Verfügung, mit denen Sie bestimmte Probleme im Zusammenhang mit Patches beheben können.

- `AWSSupport-TroubleshootWindowsUpdate`— Das `AWSSupport-TroubleshootWindowsUpdate`Runbook wird verwendet, um Probleme zu identifizieren, bei denen die Windows Server Updates für Amazon Elastic Compute Cloud (Amazon EC2) Windows Server -Instances fehlschlagen könnten.
- `AWSSupport-TroubleshootPatchManagerLinux` – Das `AWSSupport-TroubleshootPatchManagerLinux`-Runbook behebt häufig auftretende Probleme, die zu einem Patch-Fehler auf Linux-basierten verwalteten Knoten führen können, mithilfe von Patch Manager. Das Hauptziel dieses Runbooks besteht darin, die Hauptursache des Fehlers beim Patch-Befehl zu ermitteln und einen Plan zur Behebung vorzuschlagen.

> **Note**
>
> Für die Ausführung von Automation-Runbooks fallen Gebühren an. Weitere Informationen finden Sie unter AWS Systems Manager Preise für Automatisierung.

Kontaktaufnahme mit AWS -Support

Wenn Sie Problembehandlungs-Lösungen in diesem Abschnitt oder im bschnitt zu Systems-Manager-Problemen in AWS re:Post nicht finden können und einen Developer-, Business- oder

Enterprise- Support -Plan haben, können Sie unter AWS -Support einen technischen Supportfall erstellen.

Sammeln Sie die folgenden Artikel Support, bevor Sie Kontakt aufnehmen:

- SSM-Agent-Protokolle
- Run Command-Befehls-ID, Wartungsfenster-ID oder Automatisierungsausführungs-ID
- Sammeln Sie für von Windows Server verwaltete Knoten auch Folgendes:
  - `%PROGRAMDATA%\Amazon\PatchBaselineOperations\Logs`, wie auf der Windows-Registerkarte von Wie Patches installiert werden beschrieben
  - Windows Update-Protokolle: Für Windows Server 2012 R2 und älter verwenden Sie `%windir%/WindowsUpdate.log`. Führen Sie für Windows Server 2016 und neuere Versionen zuerst den PowerShell Befehl aus, `Get-WindowsUpdateLog`bevor Sie `%windir%/WindowsUpdate.log`
- Sammeln Sie für Linux-verwaltete Knoten auch Folgendes:
  - Der Inhalt des Verzeichnisses `/var/lib/amazon/ssm/`*`instance-id`*`/document/orchestration/`*`Run-Command-execution-id`*`/awsrunShellScript/PatchLinux`

AWS Systems Manager Run Command

Mithilfe Run Command eines Tools in AWS Systems Manager können Sie die Konfiguration Ihrer verwalteten Knoten remote und sicher verwalten. Ein verwalteter Knoten ist jede Amazon Elastic Compute Cloud (Amazon EC2) -Instanz oder EC2 Nicht-Maschine in Ihrer Hybrid- und Multi-Cloud-Umgebung, die für Systems Manager konfiguriert wurde. Run Commandermöglicht es Ihnen, allgemeine Verwaltungsaufgaben zu automatisieren und einmalige Konfigurationsänderungen in großem Umfang durchzuführen. Sie können Run Command von AWS-Managementkonsole, dem AWS Command Line Interface (AWS CLI) AWS Tools for Windows PowerShell, oder dem verwenden AWS SDKs. Run Commandwird ohne zusätzliche Kosten angeboten. Um mit Run Command zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Run Command aus.

Administratoren verwenden Run Command zum Installieren oder Bootstrapping von Anwendungen, Entwickeln einer Bereitstellungs-Pipeline, Erfassen von Protokolldateien nach Entfernung einer Instance aus einer Auto-Scaling-Gruppe, zum Anschließen von Instances an eine Windows-Domain und noch mehr.

Aufgrund der verteilten Natur des Systems, das die API unterstützt, folgt die Run Command-API einem eventuellen Konsistenzmodell. Dies bedeutet, dass das Ergebnis eines von Ihnen

ausgeführten API-Befehls, der sich auf Ihre Ressourcen auswirkt, möglicherweise nicht sofort für alle nachfolgenden Befehle, die Sie ausführen, sichtbar ist. Sie sollten dies berücksichtigen, wenn Sie einen API-Befehl ausführen, der unmittelbar auf einen vorherigen API-Befehl folgt.

Erste Schritte

Die folgende Tabelle enthält Informationen, die Ihnen bei den ersten Schritten mit Run Command helfen werden.

Thema	Details
Einrichten von verwalteten Knoten für AWS Systems Manager	Stellen Sie sicher, dass Sie die Einrichtu ngsvoraussetzungen für Ihre Amazon Elastic Compute Cloud (Amazon EC2) -Instances und EC2 Nicht-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung erfüllt haben.
Verwalten von Knoten in Hybrid- und Multi-Clo ud-Umgebungen mit Systems Manager	(Optional) Registrieren Sie lokale Server und VMs mit, AWS damit Sie sie mithilfe von verwalten können. Run Command
the section called “Verwalten von Edge-Gerä ten mit Systems Manager”	(Optional) Konfigurieren Sie Edge-Geräte, damit Sie sie mit Run Command verwalten können.
Ausführen von Befehlen auf verwalteten Knoten	Erfahren Sie, wie Sie mit der AWS-Manag ementkonsole einen Befehl ausführen, der einen oder mehrere verwaltete Knoten anvisiert .
Walkthroughs für Run Command	Erfahren Sie, wie Sie Befehle entweder mit Tools für Windows PowerShell oder mit dem AWS CLI ausführen.

EventBridge Unterstützung

Dieses Systems Manager Manager-Tool wird in EventBridge Amazon-Regeln sowohl als Ereignistyp als auch als Zieltyp unterstützt. Weitere Informationen finden Sie unter Überwachung von Systems

Manager-Ereignissen mit Amazon EventBridge und Referenz: EventBridge Amazon-Ereignistypen und -muster für Systems Manager.

Weitere Informationen

- Aus Run Command der Ferne auf einer EC2 Instance (10-minütiges Tutorial)
- Systems Manager-Service Quotas im Allgemeine Amazon Web Services-Referenz
- AWS Systems Manager API Reference

Themen

- Einrichten von Run Command
- Ausführen von Befehlen auf verwalteten Knoten
- Verwendung von Beendigungscodes in Befehlen
- Grundlegendes zu Befehlsstatus
- Walkthroughs für Run Command
- Fehlerbehebung von Systems Manager Run Command

Einrichten von Run Command

Bevor Sie Knoten mithilfe eines Tools in verwalten können Run Command AWS Systems Manager, konfigurieren Sie eine AWS Identity and Access Management (IAM-) Richtlinie für jeden Benutzer, der Befehle ausführt. Wenn Sie in Ihren IAM-Richtlinien globale Bedingungsschlüssel für die `SendCommand`-Aktion verwenden, müssen Sie den `aws:ViaAWSService`-Bedingungsschlüssel angeben und den booleschen Wert auf `true` setzen. Im Folgenden wird ein -Beispiel gezeigt.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/YourDocument"
```

```
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceVpce": [
                        "vpce-1234567890abcdef0"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/YourDocument"
            ],
            "Condition": {
                "Bool": {
                    "aws:ViaAWSService": "true"
                }
            }
        }
    ]
}
```

Sie müssen Ihre Knoten auch für Systems Manager konfigurieren. Weitere Informationen finden Sie unter Einrichten von verwalteten Knoten für AWS Systems Manager.

Wir empfehlen, die folgenden optionalen Einrichtungsaufgaben durchzuführen, um den Sicherheitsstatus und die day-to-day Verwaltung Ihrer verwalteten Knoten zu minimieren.

Überwachen Sie Befehlsausführungen mit Amazon EventBridge

Sie können es verwenden EventBridge , um Statusänderungen der Befehlsausführung zu protokollieren. Sie können eine Regel erstellen, die ausgeführt wird, sobald ein Statusübergang oder ein Übergang zu einem oder mehreren Status stattfindet, die für sie von Interesse sind. Sie können auch Run Command als Zielaktion angeben, wann ein EventBridge Ereignis eintritt. Weitere Informationen finden Sie unter Konfigurieren von EventBridge für Systems-Manager-Ereignisse.

Überwachen Sie Befehlsausführungen mithilfe von Amazon Logs CloudWatch

Sie können so konfigurierenRun Command, dass alle Befehlsausgaben und Fehlerprotokolle regelmäßig an eine CloudWatch Amazon-Protokollgruppe gesendet werden. Sie können diese Ausgabeprotokolle nahezu in Echtzeit überwachen, nach bestimmten Phrasen, Werten oder Mustern suchen und auf der Grundlage der Suche Warnungen erstellen. Weitere Informationen finden Sie unter Konfigurieren von Amazon CloudWatch Logs für Run Command.

Den Zugriff von Run Command auf bestimmte verwaltete Knoten beschränken

Sie können die Fähigkeit eines Benutzers einschränken, Befehle auf verwalteten Knoten auszuführen, indem Sie AWS Identity and Access Management (IAM) verwenden. Insbesondere können Sie eine IAM-Richtlinie mit der Bedingung erstellen, dass der Benutzer nur Befehle auf verwalteten Knoten ausführen kann, die mit bestimmten Tags gekennzeichnet sind. Weitere Informationen finden Sie unter Den Zugriff von Run Command anhand von Tags beschränken.

Den Zugriff von Run Command anhand von Tags beschränken

In diesem Abschnitt wird beschrieben, wie die Fähigkeit eines Benutzers eingeschränkt werden kann, Befehle für verwaltete Knoten auszuführen, indem eine Tag-Bedingung in einer IAM-Richtlinie angegeben wird. Zu den verwalteten Knoten gehören EC2 Amazon-Instances und EC2 Nicht-Knoten in einer Hybrid- und Multi-Cloud-Umgebung, die für Systems Manager konfiguriert sind. Die Informationen werden zwar nicht explizit dargestellt, Sie können aber auch den Zugriff auf verwaltete AWS IoT Greengrass Kerngeräte einschränken. Zuerst müssen Sie Ihre AWS IoT Greengrass -Geräte markieren. Weitere Informationen finden Sie unter Markieren Ihrer AWS IoT Greengrass Version 2 -Ressourcen im AWS IoT Greengrass Version 2 -Entwicklerhandbuch.

Sie können die Befehlsausführung auf bestimmte verwaltete Knoten beschränken, indem Sie eine IAM-Richtlinie erstellen, die eine Bedingung enthält, dass der Benutzer Befehle nur auf Knoten mit bestimmten Tags ausführen kann. Im folgenden Beispiel darf der Benutzer () verwenden, indem er ein beliebiges SSM-Dokument Run Command (`Effect: Allow, Action: ssm:SendCommand`) auf einem beliebigen Knoten (`Resource: arn:aws:ssm:*:*:document/*Resource: arn:aws:ec2:*:*:instance/*`) verwendet, unter der Bedingung, dass es sich bei dem Knoten um einen Finanzknoten WebServer (`ssm:resourceTag/Finance: WebServer`) handelt. Wenn der Benutzer einen Befehl an einen Knoten sendet, der nicht markiert ist oder ein anderes Tag als `Finance: WebServer` hat, zeigen die Ausführungsergebnisse `AccessDenied`.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ssm:SendCommand"
         ],
         "Resource":[
            "arn:aws:ssm:*:*:document/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "ssm:SendCommand"
         ],
         "Resource":[
            "arn:aws:ec2:*:*:instance/*"
         ],
         "Condition":{
            "StringLike":{
               "ssm:resourceTag/Finance":[
                  "WebServers"
               ]
            }
         }
      }
   ]
}
```

Sie können IAM-Richtlinien erstellen, die es einem Benutzer erlauben, Befehle auf verwalteten Knoten auszuführen, die mit mehreren Tags markiert sind. Mit der folgenden Richtlinie hat der Benutzer die Möglichkeit, Befehle auf verwalteten Knoten auszuführen, die über zwei Tags verfügen. Wenn ein Benutzer einen Befehl an einen verwalteten Knoten sendet, der nicht mit beiden dieser Tags markiert ist, zeigen die Ausführungsergebnisse `AccessDenied`.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ssm:SendCommand"
         ],
         "Resource":"*",
         "Condition":{
            "StringLike":{
               "ssm:resourceTag/tag_key1":[
                  "tag_value1"
               ],
               "ssm:resourceTag/tag_key2":[
                  "tag_value2"
               ]
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "ssm:SendCommand"
         ],
         "Resource":[
            "arn:aws:ssm:us-west-1::document/AWS-*",
            "arn:aws:ssm:us-east-2::document/AWS-*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "ssm:UpdateInstanceInformation",
            "ssm:ListCommands",
            "ssm:ListCommandInvocations",
            "ssm:GetDocument"
         ],
         "Resource":"*"
      }
   ]
```

```
}
```

Sie können auch IAM-Richtlinien erstellen, die es einem Benutzer erlauben, Befehle auf mehreren Gruppen von markierten verwalteten Knoten auszuführen. Die folgende Beispiel-Richtlinie erlaubt dem Benutzer die Ausführung von Befehlen entweder für eine der Gruppen von markierten Knoten oder für beide Gruppen.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ssm:SendCommand"
         ],
         "Resource":"*",
         "Condition":{
            "StringLike":{
               "ssm:resourceTag/tag_key1":[
                  "tag_value1"
               ]
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "ssm:SendCommand"
         ],
         "Resource":"*",
         "Condition":{
            "StringLike":{
               "ssm:resourceTag/tag_key2":[
                  "tag_value2"
               ]
            }
         }
      },
      {
```

```
            "Effect":"Allow",
            "Action":[
               "ssm:SendCommand"
            ],
            "Resource":[
               "arn:aws:ssm:us-west-1::document/AWS-*",
               "arn:aws:ssm:us-east-2::document/AWS-*"
            ]
         },
         {
            "Effect":"Allow",
            "Action":[
               "ssm:UpdateInstanceInformation",
               "ssm:ListCommands",
               "ssm:ListCommandInvocations",
               "ssm:GetDocument"
            ],
            "Resource":"*"
         }
      ]
}
```

Weitere Informationen zum Erstellen von IAM-Richtlinien finden Sie unter Verwaltete Richtlinien und eingebundene Richtlinien im IAM-Benutzerhandbuch. Weitere Informationen über das Markieren verwalteter Knoten finden Sie unter Tag-Editor im AWS -Ressourcengruppen -Benutzerhandbuch.

Ausführen von Befehlen auf verwalteten Knoten

Dieser Abschnitt enthält Informationen darüber, wie Befehle aus der AWS Systems Manager -Konsole zu verwalteten Knoten gesendet werden. Dieser Abschnitt enthält auch Informationen zum Abbrechen eines Befehls.

Beachten Sie, dass Run Command Befehle nicht erfolgreich ausführen kann, wenn Ihr Knoten mit der `noexec`-Mount-Option für das Verzeichnis var konfiguriert ist.

> **Important**
>
> Wenn Sie einen Befehl mit Run Command senden, schließen Sie keine vertraulichen Informationen ein, die als Klartext formatiert sind, z. B. Passwörter, Konfigurationsdaten oder andere Geheimnisse. Alle Systems Manager Manager-API-Aktivitäten in Ihrem Konto werden in einem S3-Bucket für AWS CloudTrail Protokolle protokolliert. Dies bedeutet, dass

jeder Benutzer mit Zugriff auf den S3-Bucket die Klartextwerte dieser Geheimnisse anzeigen kann. Aus diesem Grund empfehlen wir, `SecureString`-Parameter zu erstellen und zu verwenden, um die sensiblen Daten zu verschlüsseln, die Sie in Ihren Systems-Manager-Operationen verwenden.
Weitere Informationen finden Sie unter Einschränken des Zugriffs auf Parameter Store-Parameter mithilfe von IAM-Richtlinien.

Aufbewahrung des Ausführungsverlaufs

Der Verlauf der einzelnen Befehle steht für bis zu 30 Tage zur Verfügung. Außerdem können Sie eine Kopie aller Protokolldateien in Amazon Simple Storage Service speichern oder einen Audit-Trail aller API-Aufrufe in AWS CloudTrail aufzeichnen.

Ähnliche Informationen

Weitere Informationen zum Ausführen von Befehlen über andere Tools finden Sie in den folgenden Themen:

- Exemplarische Vorgehensweise: Verwenden Sie das mit AWS Tools for Windows PowerShellRun Commandoder die Beispiele im AWS Systems Manager Abschnitt der AWS -Tools für PowerShell Cmdlet-Referenz.
- Exemplarische Vorgehensweise: Verwenden Sie das mit AWS CLIRun Command oder die Beispiele in der SSM-CLI-Referenz

Inhalt

- Ausführen von Befehlen über die Konsole
- Ausführen von Befehlen mit einer bestimmten Dokumentversion
- Ausführen von Befehlen in großem Maßstab
- Stornieren eines Befehls

Ausführen von Befehlen über die Konsole

Sie können ein Tool inRun Command, von verwenden AWS Systems Manager, um verwaltete Knoten AWS-Managementkonsole zu konfigurieren, ohne sich bei ihnen anmelden zu müssen. Dieses Thema enthält ein Beispiel für die Ausführung eines SSM Agent-Updates auf einem verwalteten Knoten mithilfe von Run Command.

Bevor Sie beginnen

Bevor Sie einen Befehl mit Run Command senden, müssen Sie überprüfen, ob Ihre verwalteten Knoten die Systems-Manager-Einrichtungs-Anforderungen erfüllen.

So senden Sie einen Befehl mittels Run Command

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Befehl ausführen aus.
4. Wählen Sie in der Liste Befehlsdokument ein Systems Manager-Dokument.
5. Geben Sie im Abschnitt Befehlsparameter Werte für erforderliche Parameter an.
6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Weitere Parameter:

   - Geben Sie im Feld Kommentar Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (Sekunden) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

8. Für Ratenregelung:

   - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele

verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wählen Sie einen CloudWatch Alarm aus, der auf Ihren Überwachungsbefehl angewendet werden soll. Um Ihrem Befehl einen CloudWatch Alarm hinzuzufügen, muss der IAM-Principal, der den Befehl ausführt, über die entsprechende Berechtigung verfügen`iam:createServiceLinkedRole`. Weitere Informationen zu CloudWatch Alarmen finden Sie unter CloudWatch Amazon-Alarme verwenden. Beachten Sie, dass ausstehende Befehlsaufrufe nicht ausgeführt werden, wenn Ihr Alarm aktiviert wird.
10. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Bei den S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, handelt es sich um die Berechtigungen des Instance-Profils (für EC2 Instances) oder der IAM-Servicerolle (hybridaktivierte Maschinen), die der Instance zugewiesen wurden, nicht um die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

11. Aktivieren Sie das Kontrollkästchen SNS-Benachrichtigungen aktivieren im Abschnitt SNS-Benachrichtigungen, wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

Weitere Informationen zum Konfigurieren von Amazon-SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

12. Klicken Sie auf Ausführen.

Weitere Informationen zum Abbrechen eines Befehls finden Sie unter the section called “Stornieren eines Befehls”.

Erneutes Ausführen von Befehlen

Systems Manager enthält zwei Optionen, mit denen Sie einen Befehl auf der Seite Run Command (Befehl ausführen) in der Systems Manager-Konsole erneut ausführen können.

- Rerun (Erneut ausführen): Über diese Schaltfläche können Sie denselben Befehl ausführen, ohne Änderungen daran vorzunehmen.
- In neu kopieren: Über diese Schaltfläche kopieren Sie die Einstellungen eines Befehls in einen neuen Befehl und erhalten die Möglichkeit, diese Einstellungen zu bearbeiten, bevor Sie den Befehl ausführen.

So führen Sie einen Befehl erneut aus

1. Öffnen Sie die Konsole unter. AWS Systems Manager https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie einen Befehl aus, der erneut ausgeführt werden soll. Sie können einen Befehl unmittelbar nach der Ausführung von der Befehlsdetailseite aus erneut ausführen. Sie können auch einen Befehl auswählen, den Sie zuvor auf der Registerkarte Command history (Befehlsverlauf) ausgeführt haben.
4. Wählen Sie entweder Rerun (Erneut ausführen) aus, um denselben Befehl ohne Änderungen auszuführen, oder wählen Sie Copy to new (In neuen kopieren) aus, um die Befehlseinstellungen zu bearbeiten, bevor Sie den Befehl ausführen.

Ausführen von Befehlen mit einer bestimmten Dokumentversion

Sie können den Dokumentversionsparameter verwenden, um anzugeben, welche Version eines AWS Systems Manager -Dokuments verwendet werden soll, wenn der Befehl ausgeführt wird. Sie können eine der folgenden Optionen für diesen Parameter festlegen:

- $DEFAULT
- $LATEST
- Versionsnummer:

Gehen Sie wie folgt vor, um einen Befehl unter Verwendung des Dokumentversionsparameters auszuführen.

Linux

Um Befehle mit dem AWS CLI auf lokalen Linux-Computern auszuführen

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Listen Sie alle verfügbaren Dokumente auf

   Dieser Befehl listet alle für Ihr Konto verfügbaren Dokumente auf der Grundlage von AWS Identity and Access Management (IAM-) Berechtigungen auf.

   ```
   aws ssm list-documents
   ```

3. Verwenden Sie den folgenden Befehl, um die verschiedenen Versionen eines Dokuments anzuzeigen. Ersetzen Sie es *document name* durch Ihre eigenen Informationen.

   ```
   aws ssm list-document-versions \
       --name "document name"
   ```

4. Führen Sie mit dem folgenden Befehl einen Befehl aus, der eine SSM-Dokumentversion verwendet. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

```
aws ssm send-command \
    --document-name "AWS-RunShellScript" \
    --parameters commands="echo Hello" \
    --instance-ids instance-ID \
    --document-version '$LATEST'
```

Windows

Um Befehle mit dem AWS CLI auf lokalen Windows-Computern auszuführen

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Listen Sie alle verfügbaren Dokumente auf

   Dieser Befehl listet alle für Ihr Konto verfügbaren Dokumente auf der Grundlage von AWS Identity and Access Management (IAM-) Berechtigungen auf.

   ```
   aws ssm list-documents
   ```

3. Verwenden Sie den folgenden Befehl, um die verschiedenen Versionen eines Dokuments anzuzeigen. Ersetzen Sie es *document name* durch Ihre eigenen Informationen.

   ```
   aws ssm list-document-versions ^
       --name "document name"
   ```

4. Führen Sie mit dem folgenden Befehl einen Befehl aus, der eine SSM-Dokumentversion verwendet. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   ```
   aws ssm send-command ^
       --document-name "AWS-RunShellScript" ^
       --parameters commands="echo Hello" ^
       --instance-ids instance-ID ^
       --document-version "$LATEST"
   ```

PowerShell

Um Befehle mit den Tools für auszuführen PowerShell

1. Installieren und konfigurieren Sie die AWS -Tools für PowerShell (Tools für Windows PowerShell), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren des AWS -Tools für PowerShell.

2. Listen Sie alle verfügbaren Dokumente auf

   Dieser Befehl listet alle für Ihr Konto verfügbaren Dokumente auf der Grundlage von AWS Identity and Access Management (IAM-) Berechtigungen auf.

   ```
   Get-SSMDocumentList
   ```

3. Verwenden Sie den folgenden Befehl, um die verschiedenen Versionen eines Dokuments anzuzeigen. Ersetzen Sie es *document name* durch Ihre eigenen Informationen.

   ```
   Get-SSMDocumentVersionList `
       -Name "document name"
   ```

4. Führen Sie mit dem folgenden Befehl einen Befehl aus, der eine SSM-Dokumentversion verwendet. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   ```
   Send-SSMCommand `
       -DocumentName "AWS-RunShellScript" `
       -Parameter @{commands = "echo helloWorld"} `
       -InstanceIds "instance-ID" `
       -DocumentVersion $LATEST
   ```

Ausführen von Befehlen in großem Maßstab

Sie können Run Command, ein Tool in AWS Systems Manager, verwenden, um Befehle auf einer Flotte verwalteter Knoten auszuführen, indem Sie die `targets` verwenden. Der `targets`-Parameter nimmt eine `Key,Value`-Kombination basierend auf Tags an, die Sie für Ihre verwalteten Knoten angegeben haben. Wenn Sie den Befehl ausführen, versucht das System, den Befehl auf allen verwalteten Knoten auszuführen, die den angegebenen Tags entsprechen. Weitere Informationen zum Markieren von verwalteten Instances finden Sie unter Markieren Ihrer AWS-Ressourcen im

Benutzerhandbuch zum Markieren von AWS-Ressourcen. Weitere Informationen über das Markieren Ihrer verwalteten IoT-Geräte finden Sie unter Markieren Sie Ihre AWS IoT Greengrass Version 2-Ressourcen im AWS IoT Greengrass Version 2-Entwicklerhandbuch.

Sie können auch den `targets`-Parameter verwenden, um eine Liste spezifischer IDs von verwalteten Knoten als Ziel zu wählen, wie im nächsten Abschnitt beschrieben.

Zur Steuerung der Befehlsausführung auf Hunderten und Tausenden von verwalteten Knoten umfasst Run Command auch Parameter für die Einschränkung, wie viele Knoten gleichzeitig eine Anforderung verarbeitet können, und wie viele Fehler durch einen Befehl ausgelöst werden können, bevor der Befehl beendet wird.

Inhalt

- Mehrere verwaltete Knoten anvisieren
- Verwenden von Ratensteuerungen

Mehrere verwaltete Knoten anvisieren

Sie können einen Befehl ausführen und verwaltete Knoten anvisieren, indem Sie Tags, Namen von AWS-Ressourcengruppen oder IDs von verwalteten Knoten angeben.

Die folgenden Beispiele zeigen das Befehlsformat bei Verwendung von Run Command aus der AWS Command Line Interface (AWS CLI). Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen. Die Beispielbefehle werden in diesem Abschnitt sind mit `[...]` abgeschnitten.

Beispiel 1: Angabe von Tags als Ziel

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=tag:tag-name,Values=tag-value \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
```

```
    --targets Key=tag:tag-name,Values=tag-value ^
    [...]
```

Beispiel 2: Angabe einer AWS-Ressourcengruppe nach Namen

Sie können maximal einen Ressourcengruppenamen pro Befehlsaufruf angeben. Wenn Sie eine Ressourcengruppe erstellen, empfehlen wir, `AWS::SSM:ManagedInstance` und `AWS::EC2::Instance` als Ressourcentypen in dem Gruppierungskriterium aufzunehmen.

> **Note**
>
> Zum Senden von Befehlen mit einer Ressourcengruppe als Ziel benötigen Sie AWS Identity and Access Management (IAM)-Berechtigungen zum Auflisten oder Anzeigen der Ressourcen, die zu der Gruppe gehören. Weitere Informationen finden Sie unter Einrichten von Berechtigungen im AWS -Ressourcengruppen-Benutzerhandbuch.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=resource-groups:Name,Values=resource-group-name \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=resource-groups:Name,Values=resource-group-name ^
    [...]
```

Beispiel 3: Angabe einer AWS-Ressource Groups nach Ressourcentyp

Sie können maximal fünf Ressourcengruppentypen pro Befehlsaufruf angeben. Wenn Sie eine Ressourcengruppe erstellen, empfehlen wir, `AWS::SSM:ManagedInstance` und `AWS::EC2::Instance` als Ressourcentypen in dem Gruppierungskriterium aufzunehmen.

> **Note**
>
> Zum Senden von Befehlen mit einer Ressourcengruppe als Ziel benötigen Sie IAM-Berechtigungen zum Auflisten oder Anzeigen der Ressourcen, die zu der Gruppe gehören. Weitere Informationen finden Sie unter Einrichten von Berechtigungen im AWS -Ressourcengruppen-Benutzerhandbuch.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=resource-groups:ResourceTypeFilters,Values=resource-
type-1,resource-type-2 \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=resource-groups:ResourceTypeFilters,Values=resource-
type-1,resource-type-2 ^
    [...]
```

Beispiel 4: Angabe von Instance-IDs als Ziel

Die folgenden Beispiele veranschaulichen, wie verwaltete Knoten mithilfe des `instanceids`-Schlüssels mit dem `targets`-Parameter anvisiert werden können. Sie können diesen Schlüssel verwenden, um verwaltete AWS IoT Greengrass-Core-Geräte anzuvisieren, da jedem Gerät ein mi-*ID_Nummer* zugewiesen ist. Sie können Geräte-IDs in Fleet Manager, einem Tool in AWS Systems Manager, anzeigen.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=instanceids,Values=instance-ID-1,instance-ID-2,instance-ID-3 \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=instanceids,Values=instance-ID-1,instance-ID-2,instance-ID-3 ^
    [...]
```

Wenn Sie verwaltete Knoten für unterschiedliche Umgebungen mit einem `Key` namens `Environment` und `Values` von `Development`, `Test`, `Pre-production` und`Production` markiert haben, könnten Sie einen Befehl an alle verwalteten Knoten in einer dieser Umgebungen mit dem `targets`-Parameter mit der folgenden Syntax senden.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=tag:Environment,Values=Development \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=tag:Environment,Values=Development ^
    [...]
```

Sie könnten weitere verwaltete Knoten in anderen Umgebungen auswählen, indem Sie sie zur `Values`-Liste hinzufügen. Trennen Sie die Elemente durch Kommas.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=tag:Environment,Values=Development,Test,Pre-production \
    [...]
```

Windows

```
aws ssm send-command ^
```

```
    --document-name document-name ^
    --targets Key=tag:Environment,Values=Development,Test,Pre-production ^
    [...]
```

Variation: Anpassung Ihrer Ziele mit mehreren `Key`-Kriterien

Sie können die Anzahl der Ziele für Ihren Befehl verfeinern, indem Sie mehrere `Key` Kriterien berücksichtigen. Wenn Sie mehr als ein `Key`-Kriterium einschließen, wird das System verwaltete Knoten anvisieren, die alle Kriterien erfüllen. Mit dem folgenden Befehl werden alle verwaltete Knoten anvisiert, die für die Finanzabteilung und für die Datenbankserver-Rolle markiert sind.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=tag:Department,Values=Finance Key=tag:ServerRole,Values=Database \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=tag:Department,Values=Finance Key=tag:ServerRole,Values=Database ^
    [...]
```

Variation: Verwenden mehrerer `Key`- und `Value`-Kriterien

Aufbauend auf dem vorherigen Beispiel können Sie mehrere Abteilungen und mehrere Server-Rollen auswählen, indem zusätzliche Elemente in die `Values` Kriterien einfügen.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=tag:Department,Values=Finance,Marketing
 Key=tag:ServerRole,Values=WebServer,Database \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=tag:Department,Values=Finance,Marketing
 Key=tag:ServerRole,Values=WebServer,Database ^
    [...]
```

Variation: Anvisieren markierter verwalteter Knoten mithilfe von mehreren `Values`-Kriterien

Wenn Sie verwaltete Knoten für unterschiedliche Umgebungen mit einem `Key` namens `Department` und `Values` von `Sales` und `Finance` markiert haben, könnten Sie einen Befehl an alle Knoten in diesen Umgebungen mit dem `targets`-Parameter mit der folgenden Syntax senden.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=tag:Department,Values=Sales,Finance \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=tag:Department,Values=Sales,Finance ^
    [...]
```

Sie können maximal fünf Schlüssel und fünf Werte für jeden Schlüssel angeben.

Wenn entweder ein Tag-Schlüssel (der Variablenname) oder eine Tag-Wert Leerzeichen enthält, setzen Sie den Tagschlüssel oder den Wert in Anführungszeichen, wie in den folgenden Beispielen gezeigt.

Beispiel: Leerzeichen in `Value`-Tag

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
```

```
    --targets Key=tag:OS,Values="Windows Server 2016" \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key=tag:OS,Values="Windows Server 2016" ^
    [...]
```

Beispiel: Leerzeichen in `tag`-Schlüssel und `Value`

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key="tag:Operating System",Values="Windows Server 2016" \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --targets Key="tag:Operating System",Values="Windows Server 2016" ^
    [...]
```

Beispiel: Leerzeichen in einem Element in einer Liste von `Values`

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --targets Key=tag:Department,Values="Sales","Finance","Systems Mgmt" \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
```

```
    --targets Key=tag:Department,Values="Sales","Finance","Systems Mgmt" ^
    [...]
```

Verwenden von Ratensteuerungen

Sie können das Tempo steuern, mit dem Befehle an verwaltete Knoten in einer Gruppe gesendet werden, indem Sie Nebenläufigkeits-Kontrollen und Fehlerkontrollen verwenden.

Themen

- Verwenden von Gleichzeitigkeitssteuerungen
- Verwenden von Fehlersteuerungen

Verwenden von Gleichzeitigkeitssteuerungen

Sie können die Anzahl der verwalteten Knoten steuern, die einen Befehl gleichzeitig ausführen, indem Sie den `max-concurrency`-Parameter verwenden (die Concurrency (Nebenläufigkeit)-Optionen auf der Seite Run a command (Befehl ausführen). Sie können entweder eine absolute Anzahl an verwalteten Knoten, z. B. **`10`**, oder einen Prozentsatz des festgelegten Ziels, beispielsweise **`10%`**, angeben. Das Warteschlangensystem liefert den Befehl an einen einzelnen Knoten und wartet, bis das System den ersten Aufruf bestätigt hat, bevor der Befehl an zwei weitere Knoten gesendet wird. Das System sendet exponentiell Befehle an mehrere Knoten, bis das System den Wert `max-concurrency` erreicht hat. Der Standardwert `max-concurrency` beträgt 50. Die folgenden Beispiele zeigen, wie Sie Werte für den Parameter `max-concurrency` angeben.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
    --max-concurrency 10 \
    --targets Key=tag:Environment,Values=Development \
    [...]
```

```
aws ssm send-command \
    --document-name document-name \
    --max-concurrency 10% \
    --targets Key=tag:Department,Values=Finance,Marketing
 Key=tag:ServerRole,Values=WebServer,Database \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --max-concurrency 10 ^
    --targets Key=tag:Environment,Values=Development ^
    [...]
```

```
aws ssm send-command ^
    --document-name document-name ^
    --max-concurrency 10% ^
    --targets Key=tag:Department,Values=Finance,Marketing
 Key=tag:ServerRole,Values=WebServer,Database ^
    [...]
```

Verwenden von Fehlersteuerungen

Sie können auch die Ausführung eines Befehls auf Hunderten oder Tausenden von verwalteten Knoten steuern, indem Sie eine Fehlerbegrenzung mit den `max-errors`-Parametern einstellen (das Feld Error threshold (Fehlerschwelle) auf der Seite Run a command (Befehl ausführen)). Der Parameter gibt an, wie viele Fehler zulässig sind, bevor das System keinen Befehl mehr an zusätzliche verwaltete Knoten sendet. Sie können entweder eine absolute Anzahl an Fehlern, z. B. **`10`**, oder einen Prozentsatz des festgelegten Ziels, beispielsweise **`10%`**, festlegen. Wenn Sie z. B. **`3`** angeben, sendet das System keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Wenn Sie **`0`** angeben, sendet das System keinen weiteren Befehl an zusätzliche verwaltete Knoten, nachdem das erste Fehlerergebnis zurückgegeben wird. Wenn Sie einen Befehl an 50 verwaltete Knoten senden und `max-errors` auf **`10%`** einstellen, sendet das System keinen Befehl mehr an weitere Knoten, wenn der sechste Fehler empfangen wird.

Aufrufe, die bereits einen Befehl ausführen, wenn `max-errors` erreicht ist, dürfen abgeschlossen werden, jedoch können einige dieser Aufrufe ebenso fehlschlagen. Wenn Sie sicherstellen müssen, dass es nicht mehr als `max-errors` fehlgeschlagene Aufrufe geben wird, setzen Sie `max-concurrency` auf **`1`**, sodass die Aufrufe jeweils um eins fortfahren. Die Standardwert für max-errors ist 0. Die folgenden Beispiele zeigen, wie Sie Werte für den Parameter `max-errors` angeben.

Linux & macOS

```
aws ssm send-command \
    --document-name document-name \
```

```
    --max-errors 10 \
    --targets Key=tag:Database,Values=Development \
    [...]
```

```
aws ssm send-command \
    --document-name document-name \
    --max-errors 10% \
    --targets Key=tag:Environment,Values=Development \
    [...]
```

```
aws ssm send-command \
    --document-name document-name \
    --max-concurrency 1 \
    --max-errors 1 \
    --targets Key=tag:Environment,Values=Production \
    [...]
```

Windows

```
aws ssm send-command ^
    --document-name document-name ^
    --max-errors 10 ^
    --targets Key=tag:Database,Values=Development ^
    [...]
```

```
aws ssm send-command ^
    --document-name document-name ^
    --max-errors 10% ^
    --targets Key=tag:Environment,Values=Development ^
    [...]
```

```
aws ssm send-command ^
    --document-name document-name ^
    --max-concurrency 1 ^
    --max-errors 1 ^
    --targets Key=tag:Environment,Values=Production ^
    [...]
```

Stornieren eines Befehls

Sie können versuchen, einen Befehl abzubrechen, solange der Service entweder im Ausstehenden oder Ausführenden Status angezeigt wird. Wenn jedoch ein Befehl sich nach wie vor in einem dieser Zustände befindet, können wir nicht garantieren, dass der Befehl beendet wird und der zugrunde liegenden Prozess gestoppt wurde.

So stornieren Sie einen Befehl mithilfe der Konsole

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie den Befehlsaufruf, den Sie stornieren möchten.
4. Wählen Sie Cancel Command.

So stornieren Sie einen Befehl mithilfe der AWS CLI

Führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

Linux & macOS

```
aws ssm cancel-command \
    --command-id "command-ID" \
    --instance-ids "instance-ID"
```

Windows

```
aws ssm cancel-command ^
    --command-id "command-ID" ^
    --instance-ids "instance-ID"
```

Weitere Informationen über den Status eines stornierten Befehls finden Sie unter Grundlegendes zu Befehlsstatus.

Verwendung von Beendigungscodes in Befehlen

In einigen Fällen müssen Sie möglicherweise mithilfe von Beendigungscodes verwalten, wie Ihre Befehle verarbeitet werden.

Angabe von Beendigungscodes in Befehlen

Mit Run Command, einem Tool in AWS Systems Manager, können Sie Beendigungscodes angeben, um festzulegen, wie Befehle verarbeitet werden. Standardmäßig wird der Beendigungscode des letzten in einem Skript ausgeführten Befehls als Beendigungscode für das gesamte Skript gemeldet. Sie haben beispielsweise ein Skript, das drei Befehle enthält. Der erste schlägt fehl, die folgenden werden jedoch erfolgreich ausgeführt. Da der letzte Befehl erfolgreich war, wird der Status der Ausführung als `succeeded` gemeldet.

Shell-Skripts

Damit das gesamte Skript beim ersten Befehlsfehler fehlschlägt, können Sie eine bedingte Shell-Anweisung einschließen, sodass das Skript beendet wird, wenn ein Befehl vor dem letzten Befehl fehlschlägt. Gehen Sie wie folgt vor.

```
<command 1>
    if [ $? != 0 ]
    then
        exit <N>
    fi
    <command 2>
    <command 3>
```

Im folgenden Beispiel schlägt das gesamte Skript fehl, wenn der erste Befehl fehlschlägt.

```
cd /test
    if [ $? != 0 ]
    then
        echo "Failed"
        exit 1
    fi
    date
```

PowerShell-Skripts

PowerShell erfordert, dass Sie `exit` explizit in Ihren Skripts aufrufen, damit Run Command den Beendigungscode erfolgreich erfassen kann.

```
<command 1>
    if ($?) {<do something>}
    else {exit <N>}
    <command 2>
```

```
    <command 3>
    exit <N>
```

Ein Beispiel:

```
cd C:\
    if ($?) {echo "Success"}
    else {exit 1}
    date
```

Umgang mit Neustarts beim Ausführen von Befehlen

Wenn Sie ein Tool in verwenden Run Command AWS Systems Manager, um Skripts auszuführen, die verwaltete Knoten neu starten, empfehlen wir, dass Sie in Ihrem Skript einen Exit-Code angeben. Wenn Sie versuchen, einen Knoten von einem Skript aus mit einem anderen Verfahren neu zu starten, wird der Ausführungsstatus des Skripts möglicherweise nicht korrekt aktualisiert. Dies passiert auch dann, wenn der Neustart der letzte Schritt in Ihrem Skript ist. Für Windows-verwaltete Knoten geben Sie `exit 3010` in Ihrem Skript an. Für Linux- und macOS-verwaltete Knoten geben Sie `exit 194` an. Der Exit-Code weist AWS Systems Manager Agent (SSM Agent) an, den verwalteten Knoten neu zu starten und das Skript nach Abschluss des Neustarts neu zu starten. Vor dem Neustart informiert SSM Agent den Systems Manager-Service in der Cloud, dass die Kommunikation während des Serverneustarts unterbrochen werden wird.

> **Note**
>
> Das Neustartskript kann nicht Teil eines `aws:runDocument`-Plugins sein. Wenn ein Dokument das Neustartskript enthält und ein anderes Dokument versucht, dieses Dokument über das `aws:runDocument`-Plugin auszuführen, gibt SSM Agent einen Fehler zurück.

Idempotente Skripts erstellen

Bei der Entwicklung von Skripts, die verwaltete Knoten neu starten, machen Sie die Skripts idempotent, damit die Skriptausführung nach dem Neustart an der Stelle fortgesetzt wird, wo sie unterbrochen wurde. Idempotente Skripts verwalten den Status und prüfen, ob die Aktion ausgeführt wurde oder nicht. Dadurch wird verhindert, dass ein Schritt mehrmals ausgeführt wird, wenn er nur einmal ausgeführt werden soll.

Hier finden Sie ein Beispiel für ein idempotentes Skript, das einen verwalteten Knoten mehrfach neu startet.

```
$name = Get current computer name
If ($name -ne $desiredName)
    {
        Rename computer
        exit 3010
    }

$domain = Get current domain name
If ($domain -ne $desiredDomain)
    {
        Join domain
        exit 3010
    }

If (desired package not installed)
    {
        Install package
        exit 3010
    }
```

Beispiele

Die folgenden Skript-Beispiele verwenden Beendigungscodes für den Neustart von verwalteten Knoten. Das Linux-Beispiel installiert Paket-Updates auf Amazon Linux und startet den Knoten dann neu. Das Windows Server-Beispiel installiert den Telnet-Client auf dem Knoten und startet ihn dann neu.

Amazon Linux 2

```
#!/bin/bash
yum -y update
needs-restarting -r
if [ $? -eq 1 ]
then
        exit 194
else
        exit 0
fi
```

Windows

```
$telnet = Get-WindowsFeature -Name Telnet-Client
```

```
if (-not $telnet.Installed)
    {
        # Install Telnet and then send a reboot request to SSM Agent.
        Install-WindowsFeature -Name "Telnet-Client"
        exit 3010
    }
```

Grundlegendes zu Befehlsstatus

Run Command, ein Tool in AWS Systems Manager, sendet detaillierte Informationen zu den verschiedenen Zuständen, die ein Befehl während der Verarbeitung durchläuft, sowie für jeden verwalteten Knoten, der den Befehl verarbeitet. Sie können Befehlsstatus mithilfe der folgenden Methoden überwachen:

- Wählen Sie das Symbol Refresh (Aktualisieren) auf der Registerkarte Commands (Befehle) in der Run Command-Konsolenschnittstelle.
- Rufen Sie list-commands oder list-command-invocations mittels der AWS Command Line Interface (AWS CLI) auf. Sie können auch Get-SSMCommand oder Get-SSMCommandInvocation mittels AWS Tools for Windows PowerShell verwenden.
- Konfigurieren Sie Amazon EventBridge, um auf Änderungen des Status oder Statusänderungen zu reagieren.
- Konfigurieren Sie Amazon Simple Notification Service (Amazon SNS), um Benachrichtigungen für alle Statusänderungen oder bestimmte Status wie `Failed` oder `TimedOut` zu senden.

Run Command-Status

Run Command berichtet Statusdetails für drei Bereiche: Plugins, Aufrufe und eine allgemeinen Compliance-Befehl. Ein Plugin ist ein Code-Ausführungsblock, der im SSM-Dokument des Befehls definiert ist. Weitere Informationen zu den Plugins finden Sie unter Referenz für Befehlsdokument-Plugins.

Wenn Sie einen Befehl an mehrere verwaltete Knoten gleichzeitig senden, ist jede Kopie des Befehls, die jeden Knoten anvisiert, ein Befehlsaufruf. Wenn Sie z. B. das `AWS-RunShellScript`-Dokument verwenden und einen `ifconfig`-Befehl an 20 Linux-Instances senden, hat dieser Befehl 20 Aufrufe. Jeder Befehlsaufruf berichtet einzeln einen Status. Die Plugins für einen bestimmten Befehlsaufruf berichten ebenfalls einzeln einen Berichtstatus.

Schließlich umfasst Run Command einen zusammenfassenden Befehlsstatus für alle Plugins und Aufrufe. Der aggregierte Befehlsstatus kann von dem Status, der von Plugins oder Aufrufen gemeldet wird, wie in den folgenden Tabellen gezeigt, abweichen.

> **Note**
>
> Wenn Sie Befehle mit den Parametern `max-concurrency` oder `max-errors` für eine großen Anzahl von verwalteten Knoten ausführen, spiegelt der Befehlsstatus die durch diese Parameter auferlegten Grenzen wider, wie in den folgenden Tabellen beschrieben. Weitere Informationen zu diesen Parametern finden Sie unter Ausführen von Befehlen in großem Maßstab.

Detaillierter Status für Befehls-Plugins und Aufrufe

Status	Details
Ausstehend	Der Befehl wurde noch nicht an den verwaltet en Knoten gesendet oder wurde nicht vom SSM Agent empfangen. Wird der Befehl nicht vor Ablauf der Zeitspanne, die der Summe aus dem Parameter Timeout (seconds) und dem Parameter Execution timeout entspricht, vom Agenten empfangen, ändert sich der Status in `Delivery Timed Out`.
InProgress	Systems Manager versucht, den Befehl an den verwalteten Knoten zu senden, oder der Befehl wurde vom SSM Agent empfangen und wird auf der Instance ausgeführt. Je nach Ergebnis aller Befehls-Plugins ändert sich der Status in `Success`, `Failed`, `Delivery Timed Out`, oder `Execution Timed Out`. Ausnahme: Wenn der Agent nicht ausgeführt oder auf dem Knoten nicht verfügbar ist, bleibt der Befehlsst atus bei `In Progress`, bis der Agent wieder verfügbar ist oder bis das Ausführungs-Timeou

Status	Details
	t-Limit erreicht ist. Der Status wechselt dann in einen Terminal-Status.
Verzögert	Das System versuchte, den Befehl an den verwalteten Knoten zu senden, war jedoch nicht erfolgreich. Das System startet einen erneuten Versuch.

Status	Details
Herzlichen Glückwunsch	Dieser Status wird unter verschiedenen Bedingungen zurückgegeben. Dieser Status bedeutet nicht, dass der Befehl auf dem Knoten verarbeitet wurde. Der Befehl kann beispiels weise von SSM Agent auf dem verwalteten Knoten empfangen werden und den Beendigun gscode Null zurückgeben, weil Ihre PowerShel l die Ausführung des Befehls `Execution Policy` verhindert. Diese ist ein Terminals tatus. Bedingungen, die dazu führen, dass ein Befehl einen `Success` Status zurückgibt, sind: • Beim Zielen auf eine einzelne Instance wurde der Befehl von SSM Agent auf dem verwalteten Knoten empfangen und gab einen Beendigungscode von Null zurück. • Bei der Ausrichtung auf mehrere Instances hat die Anzahl der fehlgeschlagenen Aufrufe den im Befehl angegebenen Fehlersch wellenwert nicht überschritten. • Beim Targeting auf mehrere Instance war mindestens ein Aufruf erfolgreich, während bei anderen das Timeout abgelaufen ist. Der angegebene Fehlerschwellenwert gilt weiterhin. • Beim Targeting auf ein Tag werden keine Instances gefunden, die dem Tag zugeordne t sind. • Beim Targeting auf ein Tag hat die Anzahl der fehlgeschlagenen Aufrufe den im Befehl angegebenen Fehlerschwellenwert nicht überschritten. • Beim Targeting eines Tags war mindestens ein Aufruf erfolgreich, während bei anderen

Status	Details
	die Zeit abgelaufen ist. Der angegebene Fehlerschwellenwert gilt weiterhin. • Sie haben Anwendungen oder Richtlinien, die auf Betriebssystemebene durchgesetzt werden und die Ausführung eines Befehls verhindern oder überschreiben, was dazu führt, dass der Beendigungscode Null zurückgegeben wird. **Note** Dieselben Bedingungen gelten für die Ausrichtung auf Ressourcengruppen. Um Fehler zu beheben oder weitere Informationen über die Befehlsau sführung zu erhalten, senden Sie einen Befehl, der Fehler oder Ausnahmen handhabt, indem er entsprechende Beendigungscodes (Ausgangscodes für fehlgeschlagenen Befehl (nicht null)) zurückgibt.
DeliveryTimedOut	Der Befehl wurde nicht an den verwaltet en Knoten übermittelt, bevor die gesamte Zeitbeschränkung abgelaufen ist. Gesamtaus fälle werden nicht auf die übergeordnete Befehlsbegrenzung angerechnet `max-error s` , aber sie tragen dazu bei, ob der übergeord nete Befehlsstatus `Success`, `Incomplete` oder `Delivery Timed Out` ist. Diese ist ein Terminalstatus.

Status	Details
ExecutionTimedOut	Die Befehlsautomatisierung begann auf dem verwalteten Knoten, aber der Befehl wurde vor Ablauf des Ausführungs-Timeouts nicht abgeschlossen. Ausführungs-Timeouts zählen als Fehler, wodurch eine Antwort ungleich Null gesendet wird, und Systems Manager beendet den Versuch, die Befehlsautomatisierung auszuführen, und meldet einen Fehlerstatus.
Fehlgeschlagen	Der Befehl war auf dem verwalteten Knoten nicht erfolgreich. Für ein Plugin bedeutet dies, dass der Ergebniscode nicht null war. Für einen Befehlsaufruf bedeutet dies, dass der Ergebniscode für ein oder mehrere Plugins nicht null war. Zeitüberschreitungen beim Aufrufen werden auf das `max-errors` Limit des übergeordneten Befehls angerechnet. Diese ist ein Terminalstatus.
Abgebrochen	Der Befehl wurde beendet, bevor er abgeschlossen wurde. Diese ist ein Terminalstatus.

Status	Details
Unzustellbar	Der Befehl kann nicht an den verwalteten Knoten übermittelt werden. Der Knoten existiert möglicherweise nicht oder antwortet nicht. Unzustellbare Aufrufe werden nicht auf die übergeordnete Befehlsbegrenzung angerechnet `max-errors` , aber sie tragen dazu bei, ob der übergeordnete Befehlsstatus `Success` oder `Incomplete` ist. Wenn beispielsweise alle Aufrufe in einem Befehl den Status `Undeliverable` haben, lautet der zurückgegebene Befehlsstatus `Failed`. Wenn ein Befehl jedoch fünf Aufrufe hat, von denen vier den Status `Undeliverable` zurückgeben und einer den Status `Success` zurückgibt, lautet der Status des übergeordneten Befehls `Success`. Diese ist ein Terminalstatus.
Beendet	Der übergeordnete Befehl hat sein Limit `max-errors` überschritten, und nachfolgende Befehlsaufrufe wurden vom System abgebrochen. Diese ist ein Terminalstatus.

Status	Details
InvalidPlatform	Der Befehl wurde an einen verwalteten Knoten gesendet, der nicht den erforderlichen Plattformen entspricht, wie sie im ausgewähl ten Dokument festgelegt wurden. `Invalid Platform` wird nicht auf die maximale Fehlerbegrenzung des übergeordneten Befehls angerechnet, aber es trägt dazu bei, ob der übergeordnete Befehlsstatus Success oder Failed lautet. Wenn beispielsweise alle Aufrufe in einem Befehl den Status `Invalid Platform` haben, lautet der zurückgegebene Befehlsstatus `Failed`. Wenn ein Befehl jedoch fünf Aufrufe hat, von denen vier den Status `Invalid Platform` zurückgeben und einer den Status `Success` zurückgibt, lautet der Status des übergeordneten Befehls `Success`. Diese ist ein Terminalstatus.

Status	Details
AccessDenied	Der AWS Identity and Access Managemen t (IAM)-Benutzer oder die IAM-Rolle, der/ die den Befehl initiiert, hat keinen Zugriff auf den anvisierten verwalteten Knoten. `Access Denied` zählt nicht in Bezug auf das `max-errors` -Limit des übergeordneten Befehls, es trägt aber dazu bei, ob der übergeord nete Befehlsstatus `Success` oder `Failed` lautet. Wenn beispielsweise alle Aufrufe in einem Befehl den Status `Access Denied` haben, lautet der zurückgegebene Befehlsst atus `Failed`. Wenn ein Befehl jedoch fünf Aufrufe hat, von denen vier den Status `Access Denied` zurückgeben und einer den Status `Success` zurückgibt, lautet der Status des übergeordneten Befehls `Success`. Diese ist ein Terminalstatus.

Detaillierter Status für einen Befehl

Status	Details
Ausstehend	Der Befehl wurde noch von keinem Agenten auf einem verwalteten Knoten empfangen.
InProgress	Der Befehl wurde an mindestens einen verwalteten Knoten gesendet, hat aber keinen endgültigen Status auf allen Knoten erreicht.
Verzögert	Das System versuchte, den Befehl an den Knoten zu senden, war jedoch nicht erfolgreich. Das System startet einen erneuten Versuch.
Herzlichen Glückwunsch	Der Befehl wurde vom SSM Agent auf allen angegebenen oder anvisierten verwalteten

Status	Details
	Knoten empfangen und ein Ausgangscode von null wurde zurückgegeben. Alle Befehlsaufrufe haben einen endgültigen Status erreicht, und der Wert von `max-errors` wurde nicht erreicht. Dieser Status bedeutet nicht, dass der Befehl auf allen angegebenen oder anvisierten verwalteten Knoten erfolgreich verarbeitet wurde. Diese ist ein Terminalstatus. **Note** Um Fehler zu beheben oder weitere Informationen über die Befehlsausführung zu erhalten, senden Sie einen Befehl, der Fehler oder Ausnahmen handhabt, indem er entsprechende Beendigungscodes (Ausgangscodes für fehlgeschlagenen Befehl (nicht null)) zurückgibt.
DeliveryTimedOut	Der Befehl wurde nicht an den verwalteten Knoten übermittelt, bevor die gesamte Zeitbeschränkung abgelaufen ist. Der Wert `max-errors` oder weitere Befehlsaufrufe zeigen den Status `Delivery Timed Out`. Diese ist ein Terminalstatus.
Fehlgeschlagen	Der Befehl war auf dem verwalteten Knoten nicht erfolgreich. Der Wert `max-errors` oder weitere Befehlsaufrufe zeigen den Status `Failed`. Diese ist ein Terminalstatus.

Status	Details
Unvollständig	Der Befehl wurde auf allen verwalteten Knoten versucht und einer oder mehrere der Aufrufe haben nicht den Wert `Success`. Jedoch sind nicht genügend Aufrufe fehlgeschlagen für den Status `Failed`. Diese ist ein Terminalstatus.
Abgebrochen	Der Befehl wurde beendet, bevor er abgeschlossen wurde. Diese ist ein Terminalstatus.
RateExceeded	Die Anzahl der verwalteten Knoten, die durch den Befehl anvisiert wurden, überschritt das Kontingent Ihres Kontos für ausstehende Aufrufe. Das System hat den Befehl vor der Ausführung auf einem Knoten abgebrochen. Diese ist ein Terminalstatus.
AccessDenied	Der Benutzer oder die Rolle, der oder die den Befehl initiiert, hat keinen Zugriff auf die Zielressourcengruppe. `AccessDenied` zählt nicht zum `max-errors` -Limit des übergeordneten Befehls, trägt aber dazu bei, ob der Status des übergeordneten Befehls `Success` oder `Failed` ist. (Wenn beispielsweise alle Aufrufe in einem Befehl den Status `AccessDenied` haben, dann lautet der zurückgegebene Befehlsstatus `Failed`. Wenn ein Befehl jedoch 5 Aufrufe hat, von denen 4 den Status `AccessDenied` anzeigen und 1 davon den Status `Success` anzeigt, dann lautet der Status des übergeordneten Befehls `Success`.) Diese ist ein Terminalstatus.

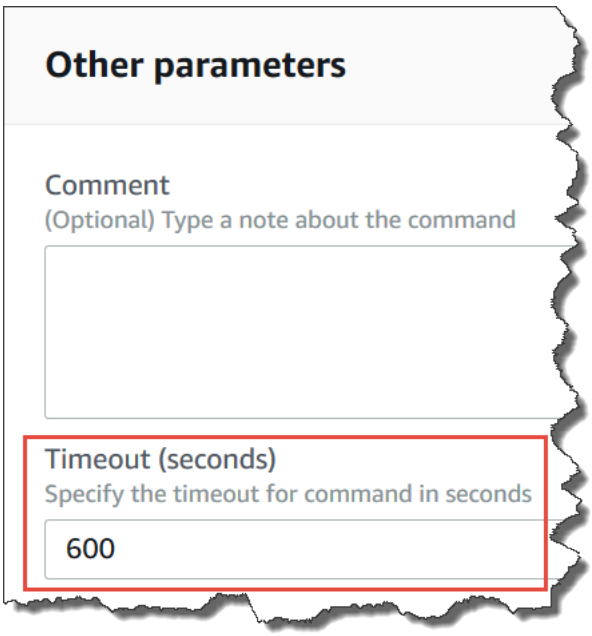
Status	Details
Keine Instances im Tag	Der Tag-Schlüsselpaar-Wert oder die Ressourcengruppe, auf die der Befehl ausgerichtet ist, stimmt mit keinem verwalteten Knoten überein. Diese ist ein Terminalstatus.

Informationen zu Timeout-Werten von Befehlen

Systems Manager erzwingt die folgenden Timeout-Werte bei der Ausführung von Befehlen.

Gesamt-Timeout

Geben Sie in der Systems-Manager-Konsole den Zeitbeschränkungs-Wert im Feld Timeout (seconds) (Zeitbeschränkung (Sekunden)) ein. Nachdem ein Befehl gesendet wurde, prüft Run Command, ob der Befehl abgelaufen ist oder nicht. Wenn ein Befehl das Ablauflimit des Befehls (Gesamtzeitlimit) erreicht, ändert er den Status in `DeliveryTimedOut` für alle Aufrufe, die den Status `InProgress`, `Pending` oder `Delayed` haben.



Technisch gesehen ist die gesamte Zeitbeschränkung (Timeout (Sekunden)) eine Kombination aus zwei Timeout-Werten, wie hier gezeigt:

```
Total timeout = "Timeout(seconds)" from the console + "timeoutSeconds":
"{{ executionTimeout }}" from your SSM document
```

Beispielsweise beträgt der Standardwert von Timeout (seconds) (Timeout (Sekunden)) 600 Sekunden in der Systems Manager-Konsole. Wenn Sie einen Befehl mit dem `AWS-RunShellScript`-SSM-Dokument ausführen, beträgt der Standardwert von „timeoutSeconds“: „{{executionTimeout}}“ 3600 Sekunden, wie im folgenden Dokumentbeispiel gezeigt:

```
"executionTimeout": {
    "type": "String",
    "default": "3600",

"runtimeConfig": {
  "aws:runShellScript": {
    "properties": [
      {
        "timeoutSeconds": "{{ executionTimeout }}"
```

Das bedeutet, dass der Befehl 4 200 Sekunden (70 Minuten) lang ausgeführt wird, bevor das System den Befehlsstatus auf `DeliveryTimedOut` setzt.

Execution Timeout

In der Systems Manager-Konsole geben Sie den Wert für die Ausführungszeitüberschreitung im Feld Execution Timeout an, sofern verfügbar. Nicht alle SSM-Dokumente erfordern die Angabe eines Ausführungs-Timeout. Das Feld Execution Timeout (Ausführungszeitlimit) wird nur angezeigt, wenn ein entsprechender Eingabeparameter im SSM-Dokument definiert wurde. Falls angegeben, muss der Befehl innerhalb dieser Zeitspanne abgeschlossen werden.

> **Note**
>
> Run Command stützt sich auf die SSM Agent-Dokument-Terminalantwort, um zu bestimmen, ob der Befehl an den Agenten übermittelt wurde oder nicht. SSM Agent muss ein `ExecutionTimedOut`-Signal senden, damit ein Aufruf oder Befehl als `ExecutionTimedOut` markiert wird.

Standard-Ausführungs-Timeout

Wenn ein SSM-Dokument nicht erfordert, dass Sie explizit einen Ausführungs-Timeout-Wert angeben, erzwingt Systems Manager den fest programmierten Standard-Ausführungs-Timeout.

Wie Systems Manager Timeouts meldet

Empfängt Systems Manager eine `execution timeout`-Antwort von SSM Agent auf einem Ziel, dann markiert Systems Manager den Befehlsaufruf als `executionTimeout`.

Erhält Run Command keine Dokumentterminalantwort von SSM Agent, wird der Befehlsaufruf als `deliveryTimeout`gekennzeichnet.

Um den Timeout-Status für ein Ziel zu bestimmen, kombiniert SSM Agent alle Parameter und den Inhalt des SSM-Dokuments, um `executionTimeout` zu berechnen. Wenn SSM Agent feststellt, dass ein Befehl einen Timeout hat, sendet es `executionTimeout` an den Service.

Der Standardwert für Timeout (seconds) (Timeout (Sekunden)) beträgt 3600 Sekunden. Der Standardwert für Execution Timeout beträgt ebenfalls 3600 Sekunden. Daher beträgt die gesamte Standard-Timeout für einen Befehl 7200 Sekunden.

> **Note**
>
> SSM Agent verarbeitet `executionTimeout` unterschiedlich, je nach Art des SSM-Dokuments und der Dokumentversion.

Walkthroughs für Run Command

Die Anleitungen in diesem Abschnitt zeigen die Ausführung von Befehlen mit Run Command, einem Tool in AWS Systems Manager, über die AWS Command Line Interface (AWS CLI) oder AWS Tools for Windows PowerShell.

Inhalt

- Aktualisierung von Software mithilfe von Run Command
- Exemplarische Vorgehensweise: Verwenden Sie das mit AWS CLIRun Command
- Exemplarische Vorgehensweise: Verwenden Sie das mit AWS Tools for Windows PowerShellRun Command

Sie können auch Beispiele für Befehle in den folgenden Referenzen anzeigen.

- Systems ManagerAWS CLI-Referenz
- AWS Tools for Windows PowerShell - AWS Systems Manager

Aktualisierung von Software mithilfe von Run Command

In den folgenden Verfahren wird beschrieben, wie Sie Software auf Ihren verwalteten Knoten aktualisieren.

Aktualisierung von SSM Agent mithilfe von Run Command

Im folgenden Verfahren wird beschrieben, wie Sie den SSM Agent auf Ihren verwalteten Knoten aktualisieren können. Sie können entweder auf die neueste Version des SSM Agent aktualisieren oder ein Downgrade auf eine ältere Version durchführen. Wenn Sie den Befehl ausführen, lädt das System die Version von herunter AWS, installiert sie und deinstalliert dann die Version, die vor der Ausführung des Befehls vorhanden war. Wenn während dieses Prozesses ein Fehler auftritt, wird das System auf die Version auf dem Server zurückgesetzt, bevor der Befehl ausgeführt wurde, und der Befehlsstatus zeigt, dass der Befehl fehlgeschlagen ist.

> **Note**
>
> Wenn auf einer Instance macOS-Version 13.0 (Ventura) oder höher ausgeführt wird, muss die Instance über die SSM Agent-Version 3.1.941.0 oder höher verfügen, um das AWS-UpdateSSMAgent-Dokument auszuführen. Wenn auf der Instance eine Version von SSM Agent ausgeführt wird, die vor 3.1.941.0 veröffentlicht wurde, können Sie Ihr SSM Agent aktualisieren, um das AWS-UpdateSSMAgent-Dokument auszuführen, indem Sie die `brew update`- und `brew upgrade amazon-ssm-agent`-Befehle ausführen.

Wenn Sie über SSM Agent-Aktualisierungen benachrichtigt werden möchten, abonnieren Sie die Seite SSM Agent-Versionshinweise auf GitHub.

So aktualisieren Sie SSM Agent mittels Run Command

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Befehl ausführen aus.
4. Wählen Sie in der Liste Befehlsdokument die Option **`AWS-UpdateSSMAgent`** aus.

5. Geben Sie im Abschnitt Command parameters ggf. Werte für die folgenden Parameter an:

   a. (Optional) Geben Sie in Version (Version) die zu installierende SSM Agent-Version ein. Sie können ältere Versionen des Agenten installieren. Wenn Sie keine Version angeben, installiert der Service die neueste Version.

   b. (Optional) Wählen Sie in Allow Downgrade (Downgrade erlauben) die Option true (wahr) aus, um eine frühere SSM Agent-Version zu installieren. Wenn Sie diese Option auswählen, geben Sie die frühere Versionsnummer an. Wählen Sie false, um nur die neueste Version des Dienstes zu installieren.

6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Weitere Parameter:

   - Geben Sie im Feld Kommentar Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (Sekunden) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

8. Für Ratenregelung:

   - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten

Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, sind diejenigen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle (hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, nicht die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

10. Aktivieren Sie das Kontrollkästchen SNS-Benachrichtigungen aktivieren im Abschnitt SNS-Benachrichtigungen, wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon-SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

11. Klicken Sie auf Ausführen.

PowerShell Aktualisierung mit Run Command

Im folgenden Verfahren wird beschrieben, wie Sie auf Ihren verwalteten R2-Knoten Windows Server 2012 und 2012 auf Version 5.1 aktualisieren PowerShell . Das in diesem Verfahren bereitgestellte Skript lädt das Update für Windows Management Framework (WMF) Version 5.1 herunter und startet die Installation des Updates. Der Knoten wird während dieses Prozesses neu gestartet, da dies bei der Installation von WMF 5.1 erforderlich ist. Download und Installation des Updates dauern insgesamt etwa fünf Minuten.

Um zu aktualisieren PowerShell mit Run Command

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Befehl ausführen aus.
4. Wählen Sie in der Liste Command document (Befehlsdokument) die Option **AWS-RunPowerShellScript** aus.
5. Fügen Sie die folgenden Befehle für Ihr Betriebssystem im Abschnitt Befehle ein.

   Windows Server 2012 R2

   ```
   Set-Location -Path "C:\Windows\Temp"

   Invoke-WebRequest "https://go.microsoft.com/fwlink/?linkid=839516" -OutFile
    "Win8.1AndW2K12R2-KB3191564-x64.msu"

   Start-Process -FilePath "$env:systemroot\system32\wusa.exe" -Verb RunAs -
   ArgumentList ('Win8.1AndW2K12R2-KB3191564-x64.msu', '/quiet')
   ```

   Windows Server 2012

   ```
   Set-Location -Path "C:\Windows\Temp"

   Invoke-WebRequest "https://go.microsoft.com/fwlink/?linkid=839513" -OutFile
    "W2K12-KB3191565-x64.msu"

   Start-Process -FilePath "$env:systemroot\system32\wusa.exe" -Verb RunAs -
   ArgumentList ('W2K12-KB3191565-x64.msu', '/quiet')
   ```

6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Weitere Parameter:

   - Geben Sie im Feld Kommentar Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (Sekunden) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

8. Für Ratenregelung:

   - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, sind diejenigen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle (hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, nicht die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die

IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

10. Aktivieren Sie das Kontrollkästchen SNS-Benachrichtigungen aktivieren im Abschnitt SNS-Benachrichtigungen, wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon-SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.
11. Klicken Sie auf Ausführen.

Nachdem der verwaltete Knoten neu gestartet wurde und die Installation des Updates abgeschlossen ist, stellen Sie eine Verbindung zu Ihrem Knoten her, um zu überprüfen, ob das Upgrade auf Version 5.1 PowerShell erfolgreich abgeschlossen wurde. Um die Version von PowerShell auf Ihrem Node zu überprüfen, öffnen Sie die Datei PowerShell und geben Sie die Eingabetaste ein`$PSVersionTable`. Der `PSVersion`-Wert in der Ausgabetabelle zeigt 5.1, wenn das Upgrade erfolgreich war.

Wenn der `PSVersion`-Wert nicht 5.1 ist, zum Beispiel 3.0 oder 4.0, überprüfen Sie die Setup-Protokolle im Event Viewer unter Windows-Protokolle. Diese Protokolle geben an, warum die Update-Installation fehlgeschlagen ist.

Exemplarische Vorgehensweise: Verwenden Sie das mit AWS CLIRun Command

In der folgenden exemplarischen Vorgehensweise wird gezeigt, wie Sie mithilfe von AWS Command Line Interface (AWS CLI) Informationen zu Befehlen und Befehlsparametern anzeigen, Befehle ausführen und den Status dieser Befehle anzeigen.

> **Important**
>
> Nur vertrauenswürdige Administratoren sollten die in diesem Thema aufgeführten AWS Systems Manager vorkonfigurierten Dokumente verwenden dürfen. Die in Systems-Manager-Dokumenten festgelegten Befehle oder Skripts werden mit Administratorberechtigungen auf Ihren verwalteten Knoten ausgeführt. Wenn ein Benutzer über die Berechtigung zum Ausführen der vordefinierten Systems-Manager-Dokumente (alle Dokumente, die mit `AWS-` beginnen) verfügt, hat dieser Benutzer auch Administratorzugriff auf den Knoten. Für alle anderen Benutzer sollten Sie restriktive Dokumente erstellen und sie mit bestimmten Benutzern teilen.

Themen

- Schritt 1: Erste Schritte
- Schritt 2: Ausführen von Shell-Skripten zum Anzeigen von Ressourcendetails
- Schritt 3: Senden einfacher Befehle mit dem AWS-RunShellScript-Dokument
- Schritt 4: Ausführen eines einfachen Python-Skripts mit Run Command
- Schritt 5: Führen Sie ein Bash-Skript mit Run Command aus

Schritt 1: Erste Schritte

Sie müssen entweder über Administratorberechtigungen auf dem verwalteten Knoten verfügen, den Sie konfigurieren möchten, oder Sie müssen über die geeignete Berechtigung in AWS Identity and Access Management (IAM) verfügen. Beachten Sie auch, dass in diesem Beispiel die Region USA Ost (Ohio) (us-east-2) verwendet wird. Run Commandist in den in Systems Manager AWS-Regionen aufgelisteten Dienstendpunkten in der Allgemeine Amazon Web Services-Referenzverfügbar. Weitere Informationen finden Sie unter Einrichten von verwalteten Knoten für AWS Systems Manager.

Um Befehle mit dem auszuführen AWS CLI

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Listen Sie alle verfügbaren Dokumente auf.

   Mit diesem Befehl werden alle verfügbaren Dokumente für Ihr Konto basierend auf IAM-Berechtigungen ausgeführt.

   ```
   aws ssm list-documents
   ```
3. Überprüfen Sie, ob ein verwalteter Knoten zum Empfangen von Befehlen bereit ist.

   Die Ausgabe des folgenden Befehls zeigt, ob verwaltete Knoten online sind.

   Linux & macOS

   ```
   aws ssm describe-instance-information \
       --output text --query "InstanceInformationList[*]"
   ```

Windows

```
aws ssm describe-instance-information ^
    --output text --query "InstanceInformationList[*]"
```

4. Verwenden Sie den folgenden Befehl, um weitere Details zu einem bestimmten verwalteten Knoten anzuzeigen.

> **Note**
>
> Um die Befehle in dieser exemplarischen Vorgehensweise auszuführen, ersetzen Sie die Instanz und den Befehl IDs. Verwenden Sie für verwaltete AWS IoT Greengrass Core-Geräte die Instanz-ID mi- *ID_number* for. Die Befehls-ID wird als Antwort an send-command zurückgegeben. Instanzen IDs sind verfügbar unterFleet Manager, einem Tool in AWS Systems Manager..

Linux & macOS

```
aws ssm describe-instance-information \
    --instance-information-filter-list key=InstanceIds,valueSet=instance-ID
```

Windows

```
aws ssm describe-instance-information ^
    --instance-information-filter-list key=InstanceIds,valueSet=instance-ID
```

Schritt 2: Ausführen von Shell-Skripten zum Anzeigen von Ressourcendetails

Mit Run Command und dem `AWS-RunShellScript`-Dokument können Sie Befehle oder Skripts auf einem verwalteten Knoten ausführen, als ob Sie lokal angemeldet wären.

Die Beschreibung und verfügbare Parameter anzeigen

Führen Sie den folgenden Befehl aus, um eine Beschreibung des Systems Manager JSON-Dokuments anzuzeigen.

Linux & macOS

```
aws ssm describe-document \
    --name "AWS-RunShellScript" \
    --query "[Document.Name,Document.Description]"
```

Windows

```
aws ssm describe-document ^
    --name "AWS-RunShellScript" ^
    --query "[Document.Name,Document.Description]"
```

Führen Sie den folgenden Befehl aus, um die verfügbaren Parameter und Details zu diesen Parametern anzuzeigen.

Linux & macOS

```
aws ssm describe-document \
    --name "AWS-RunShellScript" \
    --query "Document.Parameters[*]"
```

Windows

```
aws ssm describe-document ^
    --name "AWS-RunShellScript" ^
    --query "Document.Parameters[*]"
```

Schritt 3: Senden einfacher Befehle mit dem **`AWS-RunShellScript`**-Dokument

Führen Sie den folgenden Befehl aus, um IP-Informationen für einen Linux-verwalteten Knoten abzurufen.

Wenn Sie einen von Windows Server verwalteten Knoten anvisieren, ändern Sie den `document-name` zu `AWS-RunPowerShellScript` und ändern Sie den `command` von `ifconfig` zu `ipconfig`.

Linux & macOS

```
aws ssm send-command \
    --instance-ids "instance-ID" \
```

```
    --document-name "AWS-RunShellScript" \
    --comment "IP config" \
    --parameters commands=ifconfig \
    --output text
```

Windows

```
aws ssm send-command ^
    --instance-ids "instance-ID" ^
    --document-name "AWS-RunShellScript" ^
    --comment "IP config" ^
    --parameters commands=ifconfig ^
    --output text
```

Abrufen der Befehlsinformation mit Antwortdaten

Mit dem folgenden Befehl wird die Befehls-ID verwendet, die vom vorherigen Befehl zurückgegeben wurde, um die Details und Antwortdaten der Ausführung des Befehls abzurufen. Das System gibt die Antwortdaten zurück, wenn der Befehl abgeschlossen ist. Wenn die Befehlsausführung `"Pending"` oder `"InProgress"` anzeigt, führen Sie diesen Befehl erneut aus, um die Antwortdaten zu sehen.

Linux & macOS

```
aws ssm list-command-invocations \
    --command-id $sh-command-id \
    --details
```

Windows

```
aws ssm list-command-invocations ^
    --command-id $sh-command-id ^
    --details
```

Benutzer identifizieren

Mit dem folgenden Befehl wird der Standard-Benutzer angezeigt, der die Befehle ausführt.

Linux & macOS

```
sh_command_id=$(aws ssm send-command \
```

```
    --instance-ids "instance-ID" \
    --document-name "AWS-RunShellScript" \
    --comment "Demo run shell script on Linux managed node" \
    --parameters commands=whoami \
    --output text \
    --query "Command.CommandId")
```

Abrufen des Befehlsstatus

Mit dem folgenden Befehl wird die Befehls-ID verwendet, um den Status der Befehlsausführung auf dem verwalteten Knoten abzurufen. In diesem Beispiel wird die Befehls-ID verwendet, die im vorherigen Befehl zurückgegeben wurde.

Linux & macOS

```
aws ssm list-commands \
    --command-id "command-ID"
```

Windows

```
aws ssm list-commands ^
    --command-id "command-ID"
```

Abrufen der Befehlsdetails

Mit dem folgenden Befehl wird die Befehls-ID vom vorherigen Befehl verwendet, um den Status der Befehlsausführung pro verwalteten Knoten abzurufen.

Linux & macOS

```
aws ssm list-command-invocations \
    --command-id "command-ID" \
    --details
```

Windows

```
aws ssm list-command-invocations ^
    --command-id "command-ID" ^
    --details
```

Abrufen von Befehlsinformationen mit Antwortdaten für einen bestimmten verwalteten Knoten

Mit dem folgenden Befehl wird die Ausgabe der ursprünglichen `aws ssm send-command`-Anforderung für einen bestimmten verwalteten Knoten zurückgegeben.

Linux & macOS

```
aws ssm list-command-invocations \
    --instance-id instance-ID \
    --command-id "command-ID" \
    --details
```

Windows

```
aws ssm list-command-invocations ^
    --instance-id instance-ID ^
    --command-id "command-ID" ^
    --details
```

Anzeigen der Python-Version

Mit dem folgenden Befehl wird die Version von Python zurückgegeben, die auf einem Knoten ausgeführt wird.

Linux & macOS

```
sh_command_id=$(aws ssm send-command \
    --instance-ids "instance-ID" \
    --document-name "AWS-RunShellScript" \
    --comment "Demo run shell script on Linux Instances" \
    --parameters commands='python -V' \
    --output text --query "Command.CommandId") \
    sh -c 'aws ssm list-command-invocations \
    --command-id "$sh_command_id" \
    --details \
    --query "CommandInvocations[].CommandPlugins[].{Status:Status,Output:Output}"'
```

Schritt 4: Ausführen eines einfachen Python-Skripts mit Run Command

Mit dem folgenden Befehl wird ein einfaches Python-Skript „Hello World“ unter Verwendung von Run Command ausgeführt.

Linux & macOS

```
sh_command_id=$(aws ssm send-command \
    --instance-ids "instance-ID" \
    --document-name "AWS-RunShellScript" \
    --comment "Demo run shell script on Linux Instances" \
    --parameters '{"commands":["#!/usr/bin/python","print \"Hello World from python
\""]}' \
    --output text \
    --query "Command.CommandId") \
    sh -c 'aws ssm list-command-invocations \
    --command-id "$sh_command_id" \
    --details \
    --query "CommandInvocations[].CommandPlugins[].{Status:Status,Output:Output}"'
```

Schritt 5: Führen Sie ein Bash-Skript mit Run Command aus

Die Beispiele in diesem Abschnitt zeigen, wie Sie das folgende Bash-Skript mit Run Commandausführen.

Für Beispiele für die Verwendung von Run Command, um Skripts auszuführen, die an Remote-Speicherorten gespeichert sind, siehe Ausführen von Skripts von Amazon S3 und Ausführen von Skripts von GitHub.

```
#!/bin/bash
yum -y update
yum install -y ruby
cd /home/ec2-user
curl -O https://aws-codedeploy-us-east-2.s3.amazonaws.com/latest/install
chmod +x ./install
./install auto
```

Dieses Skript installiert den AWS CodeDeploy Agenten auf Amazon Linux- und Red Hat Enterprise Linux (RHEL) -Instances, wie unter EC2Amazon-Instance erstellen für CodeDeploy im AWS CodeDeploy Benutzerhandbuch beschrieben.

Das Skript installiert den CodeDeploy Agenten aus einem AWS verwalteten S3-Bucket in der Region USA Ost (Ohio) (us-east-2),. `aws-codedeploy-us-east-2`

Führen Sie ein Bash-Skript in einem Befehl aus AWS CLI

Das folgende Beispiel zeigt, wie Sie das Bash-Skript mit der Option `--parameters` in einen CLI-Befehl einbinden.

Linux & macOS

```
aws ssm send-command \
    --document-name "AWS-RunShellScript" \
    --targets '[{"Key":"InstanceIds","Values":["instance-id"]}]' \
    --parameters '{"commands":["#!/bin/bash","yum -y update","yum
 install -y ruby","cd /home/ec2-user","curl -O https://aws-codedeploy-us-
east-2.s3.amazonaws.com/latest/install","chmod +x ./install","./install auto"]}'
```

Führen Sie ein Bash-Skript in einer JSON-Datei aus

Im folgenden Beispiel wird der Inhalt des Bash-Skripts in einer JSON-Datei gespeichert, und die Datei wird mit der Option `--cli-input-json` in den Befehl aufgenommen.

Linux & macOS

```
aws ssm send-command \
    --document-name "AWS-RunShellScript" \
    --targets "Key=InstanceIds,Values=instance-id" \
    --cli-input-json file://installCodeDeployAgent.json
```

Windows

```
aws ssm send-command ^
    --document-name "AWS-RunShellScript" ^
    --targets "Key=InstanceIds,Values=instance-id" ^
    --cli-input-json file://installCodeDeployAgent.json
```

Der Inhalt der referenzierten `installCodeDeployAgent.json`-Datei ist im folgenden Beispiel dargestellt.

```
{
```

```
    "Parameters": {
        "commands": [
            "#!/bin/bash",
            "yum -y update",
            "yum install -y ruby",
            "cd /home/ec2-user",
            "curl -O https://aws-codedeploy-us-east-2.s3.amazonaws.com/latest/install",
            "chmod +x ./install",
            "./install auto"
        ]
    }
}
```

Exemplarische Vorgehensweise: Verwenden Sie das mit AWS Tools for Windows PowerShellRun Command

In den folgenden Beispielen wird gezeigt, wie Sie mithilfe von Informationen AWS Tools for Windows PowerShell zu Befehlen und Befehlsparametern anzeigen, Befehle ausführen und den Status dieser Befehle anzeigen können. Diese Anleitung umfasst ein Beispiel für jedes der vordefinierten AWS Systems Manager -Dokumente.

> **Important**
>
> Nur vertrauenswürdige Administratoren sollten Systems Manager-vorkonfigurierte Dokumente in diesem Thema verwenden dürfen. Die in Systems-Manager-Dokumenten festgelegten Befehle oder Skripts werden mit einer Administratorberechtigung auf Ihren verwalteten Knoten ausgeführt. Wenn ein Benutzer berechtigt ist, eines der vordefinierten Systems Manager Manager-Dokumente (jedes Dokument, das mit beginnt AWS) auszuführen, hat dieser Benutzer auch Administratorzugriff auf den Knoten. Für alle anderen Benutzer sollten Sie restriktive Dokumente erstellen und sie mit bestimmten Benutzern teilen.

Themen

- Konfigurieren Sie die AWS Tools for Windows PowerShell Sitzungseinstellungen
- Listen Sie alle verfügbaren Dokumente auf
- Führen Sie PowerShell Befehle oder Skripts aus
- Installieren einer Anwendung mithilfe des AWS-InstallApplication-Dokuments
- Installieren Sie ein PowerShell Modul mithilfe des AWS-InstallPowerShellModule JSON-Dokuments

- Verbinden eines verwalteten Knotens mit einer Domain mithilfe des AWS-JoinDirectoryServiceDomain-JSON-Dokuments
- Senden Sie Windows-Metriken mithilfe des AWS-ConfigureCloudWatch Dokuments an Amazon CloudWatch Logs
- Aktivieren oder deaktivieren Sie die automatische Windows-Aktualisierung mithilfe des AWS-ConfigureWindowsUpdate-Dokuments.
- Verwalten von Windows-Updates mit Run Command

Konfigurieren Sie die AWS Tools for Windows PowerShell Sitzungseinstellungen

Angeben Ihrer Anmeldeinformationen

Öffnen Sie Tools für Windows PowerShell auf Ihrem lokalen Computer und führen Sie den folgenden Befehl aus, um Ihre Anmeldeinformationen anzugeben. Sie müssen entweder über Administratorrechte für die verwalteten Knoten verfügen, die Sie konfigurieren möchten, oder Ihnen müssen die entsprechenden Berechtigungen in AWS Identity and Access Management (IAM) erteilt worden sein. Weitere Informationen finden Sie unter Einrichten von verwalteten Knoten für AWS Systems Manager.

```
Set-AWSCredentials –AccessKey key-name –SecretKey key-name
```

Legen Sie einen Standard fest AWS-Region

Führen Sie den folgenden Befehl aus, um die Region für Ihre PowerShell Sitzung festzulegen. Das Beispiel verwendet die Region USA Ost (Ohio) (us-east-2). Run Commandist in den in Systems Manager AWS-Regionen aufgelisteten Dienstendpunkten in der Allgemeine Amazon Web Services-Referenzverfügbar.

```
Set-DefaultAWSRegion `
    -Region us-east-2
```

Listen Sie alle verfügbaren Dokumente auf

Mit diesem Befehl werden alle verfügbaren Dokumente für Ihrem Konto aufgelistet.

```
Get-SSMDocumentList
```

Führen Sie PowerShell Befehle oder Skripts aus

Mit Run Command und dem `AWS-RunPowerShell`-Dokument können Sie Befehle oder Skripts auf einem verwalteten Knoten ausführen, als ob Sie lokal angemeldet wären. Sie können Befehle ausgeben oder einen Pfad zu einem lokalen Skript eingeben, um den Befehl auszuführen.

> **Note**
>
> Informationen zum Neustarten von verwalteten Knoten bei Verwendung von Run Command für den Aufruf von Skripts finden Sie unter Umgang mit Neustarts beim Ausführen von Befehlen.

Die Beschreibung und verfügbare Parameter anzeigen

```
Get-SSMDocumentDescription `
    -Name "AWS-RunPowerShellScript"
```

Weitere Informationen über Parameter anzeigen

```
Get-SSMDocumentDescription `
    -Name "AWS-RunPowerShellScript" | Select -ExpandProperty Parameters
```

Senden Sie einen Befehl mithilfe des **`AWS-RunPowerShellScript`**-Dokuments

Mit dem folgenden Befehl werden der Inhalt des `"C:\Users"`-Verzeichnisses und der Inhalt des `"C:\"`-Verzeichnisses auf zwei verwalteten Knoten angezeigt.

```
$runPSCommand = Send-SSMCommand `
    -InstanceIds @("instance-ID-1", "instance-ID-2") `
    -DocumentName "AWS-RunPowerShellScript" `
    -Comment "Demo AWS-RunPowerShellScript with two instances" `
    -Parameter @{'commands'=@('dir C:\Users', 'dir C:\')}
```

Abrufen der Befehlsabfragedetails

Mit dem folgenden Befehl wird die `CommandId` verwendet, um den Status der Befehlsausführung auf beiden verwalteten Knoten abzurufen. In diesem Beispiel wird die `CommandId` verwendet, die im vorherigen Befehl zurückgegeben wurde.

```
Get-SSMCommand `
    -CommandId $runPSCommand.CommandId
```

Der Status des Befehls in diesem Beispiel kann „Erfolgreich“, „Ausstehend“ oder „Ausstehend“ lauten InProgress.

Abrufen von Befehlsinformationen pro verwalteter Knoten

Mit dem folgenden Befehl wird die `CommandId` vom vorherigen Befehl verwendet, um den Status der Befehlsausführung pro verwalteten Knoten abzurufen.

```
Get-SSMCommandInvocation `
    -CommandId $runPSCommand.CommandId
```

Abrufen von Befehlsinformationen mit Antwortdaten für einen bestimmten verwalteten Knoten

Mit dem folgenden Befehl wird die Ausgabe des ursprünglichen `Send-SSMCommand` für einen bestimmten verwalteten Knoten zurückgegeben.

```
Get-SSMCommandInvocation `
    -CommandId $runPSCommand.CommandId `
    -Details $true `
    -InstanceId instance-ID | Select -ExpandProperty CommandPlugins
```

Abbrechen eines Befehls

Mit dem folgenden Befehl wird `Send-SSMCommand` für das `AWS-RunPowerShellScript`-Dokument abgebrochen.

```
$cancelCommand = Send-SSMCommand `
    -InstanceIds @("instance-ID-1","instance-ID-2") `
    -DocumentName "AWS-RunPowerShellScript" `
    -Comment "Demo AWS-RunPowerShellScript with two instances" `
    -Parameter @{'commands'='Start-Sleep -Seconds 120; dir C:\'}

Stop-SSMCommand -CommandId $cancelCommand.CommandId
```

Überprüfen des Befehlsstatus

Mit dem folgenden Befehl wird der Status des `Cancel`-Befehls überprüft

```
Get-SSMCommand `
    -CommandId $cancelCommand.CommandId
```

Installieren einer Anwendung mithilfe des **`AWS-InstallApplication`**-Dokuments

Mit Run Command und dem `AWS-InstallApplication`-Dokument können Sie Anwendungen auf verwalteten Knoten installieren, reparieren oder deinstallieren. Der Befehl erfordert den Pfad oder die Adresse für ein MSI.

> **Note**
>
> Informationen zum Neustarten von verwalteten Knoten bei Verwendung von Run Command für den Aufruf von Skripts finden Sie unter Umgang mit Neustarts beim Ausführen von Befehlen.

Die Beschreibung und verfügbare Parameter anzeigen

```
Get-SSMDocumentDescription `
    -Name "AWS-InstallApplication"
```

Weitere Informationen über Parameter anzeigen

```
Get-SSMDocumentDescription `
    -Name "AWS-InstallApplication" | Select -ExpandProperty Parameters
```

Senden Sie einen Befehl mithilfe des **`AWS-InstallApplication`**-Dokuments

Mit dem folgenden Befehl wird eine Version von Python auf Ihrerm verwalteten Knoten im unbeaufsichtigten Modus installiert und die Ausgabe in einer lokalen Textdatei auf dem Laufwerk `C:` protokolliert.

```
$installAppCommand = Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-InstallApplication" `
    -Parameter @{'source'='https://www.python.org/ftp/python/2.7.9/python-2.7.9.msi';
 'parameters'='/norestart /quiet /log c:\pythoninstall.txt'}
```

Abrufen von Befehlsinformationen pro verwalteter Knoten

Mit dem folgenden Befehl wird die `CommandId` verwendet, um den Status der Befehlsausführung abzurufen.

```
Get-SSMCommandInvocation `
    -CommandId $installAppCommand.CommandId `
    -Details $true
```

Abrufen von Befehlsinformationen mit Antwortdaten für einen bestimmten verwalteten Knoten

Mit dem folgenden Befehl werden die Ergebnisse der Python-Installation zurückgegeben.

```
Get-SSMCommandInvocation `
    -CommandId $installAppCommand.CommandId `
    -Details $true `
    -InstanceId instance-ID | Select -ExpandProperty CommandPlugins
```

Installieren Sie ein PowerShell Modul mithilfe des **`AWS-InstallPowerShellModule`** JSON-Dokuments

Sie können es verwendenRun Command, um PowerShell Module auf verwalteten Knoten zu installieren. Weitere Informationen zu PowerShell Modulen finden Sie unter PowerShell Windows-Module.

Die Beschreibung und verfügbare Parameter anzeigen

```
Get-SSMDocumentDescription `
    -Name "AWS-InstallPowerShellModule"
```

Weitere Informationen über Parameter anzeigen

```
Get-SSMDocumentDescription `
    -Name "AWS-InstallPowerShellModule" | Select -ExpandProperty Parameters
```

Installieren Sie ein PowerShell Modul

Mit dem folgenden Befehl wird die EZOut ZIP-Datei heruntergeladen, installiert und anschließend ein zusätzlicher Befehl zur Installation des XPS-Viewers ausgeführt. Schließlich wird die Ausgabe dieses Befehls auf einen S3-Bucket mit dem Namen „amzn-s3-demo-bucket“ hochgeladen.

```
$installPSCommand = Send-SSMCommand `
```

```
    -InstanceId instance-ID `
    -DocumentName "AWS-InstallPowerShellModule" `
    -Parameter @{'source'='https://gallery.technet.microsoft.com/EZOut-33ae0fb7/
file/110351/1/EZOut.zip';'commands'=@('Add-WindowsFeature -name XPS-Viewer -restart')}
 `
    -OutputS3BucketName amzn-s3-demo-bucket
```

Abrufen von Befehlsinformationen pro verwalteter Knoten

Mit dem folgenden Befehl wird die `CommandId` verwendet, um den Status der Befehlsausführung abzurufen.

```
Get-SSMCommandInvocation `
    -CommandId $installPSCommand.CommandId `
    -Details $true
```

Abrufen von Befehlsinformationen mit Antwortdaten für den verwalteten Knoten

Mit dem folgenden Befehl wird die Ausgabe des ursprünglichen `Send-SSMCommand`-Befehls für die spezielle `CommandId` zurückgegeben.

```
Get-SSMCommandInvocation `
    -CommandId $installPSCommand.CommandId `
    -Details $true | Select -ExpandProperty CommandPlugins
```

Verbinden eines verwalteten Knotens mit einer Domain mithilfe des **`AWS-JoinDirectoryServiceDomain`**-JSON-Dokuments

Mit Run Command dieser Option können Sie einen verwalteten Knoten schnell einer Domäne hinzufügen. AWS Directory Service Erstellen Sie ein Verzeichnis vor dem Ausführen dieses Befehls. Wir empfehlen außerdem, sich mit der Directory Service besser vertraut zu machen. Weitere Informationen finden Sie im Administrationshandbuch zu AWS Directory Service.

Sie können nur einen verwalteten Knoten mit einer Domain verbinden. Sie können keinen Knoten aus einer Domain entfernen.

> **Note**
>
> Informationen zu verwalteten Knoten bei Verwendung von Run Command für den Aufruf von Skripts finden Sie unter Umgang mit Neustarts beim Ausführen von Befehlen.

Die Beschreibung und verfügbare Parameter anzeigen

```
Get-SSMDocumentDescription `
    -Name "AWS-JoinDirectoryServiceDomain"
```

Weitere Informationen über Parameter anzeigen

```
Get-SSMDocumentDescription `
    -Name "AWS-JoinDirectoryServiceDomain" | Select -ExpandProperty Parameters
```

Verbinden eines verwalteten Knotens mit einer Domain

Der folgende Befehl verbindet einen verwalteten Knoten mit der angegebenen Directory Service Domain und lädt alle generierten Ausgaben in den Amazon Simple Storage Service (Amazon S3) -Beispiel-Bucket hoch.

```
$domainJoinCommand = Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-JoinDirectoryServiceDomain" `
    -Parameter @{'directoryId'='d-example01'; 'directoryName'='ssm.example.com';
 'dnsIpAddresses'=@('192.168.10.195', '192.168.20.97')} `
    -OutputS3BucketName amzn-s3-demo-bucket
```

Abrufen von Befehlsinformationen pro verwalteter Knoten

Mit dem folgenden Befehl wird die `CommandId` verwendet, um den Status der Befehlsausführung abzurufen.

```
Get-SSMCommandInvocation `
    -CommandId $domainJoinCommand.CommandId `
    -Details $true
```

Abrufen von Befehlsinformationen mit Antwortdaten für den verwalteten Knoten

Dieser Befehl gibt die Ausgabe des ursprünglichen `Send-SSMCommand` für die spezifische `CommandId` zurück.

```
Get-SSMCommandInvocation `
    -CommandId $domainJoinCommand.CommandId `
    -Details $true | Select -ExpandProperty CommandPlugins
```

Senden Sie Windows-Metriken mithilfe des **`AWS-ConfigureCloudWatch`** Dokuments an Amazon CloudWatch Logs

Sie können Windows Server Nachrichten in den Anwendungs-, System-, Sicherheits- und Event Tracing for Windows (ETW) -Protokollen an Amazon CloudWatch Logs senden. Wenn Sie die Protokollierung zum ersten Mal aktivieren, sendet Systems Manager alle Protokolle, die innerhalb von 1 Minute generiert werden, sobald Sie mit dem Hochladen von Protokollen für die Anwendungs-, System-, Sicherheits- und ETW-Protokolle beginnen. Protokolle, die davor auftraten, werden nicht berücksichtigt. Wenn Sie die Protokollierung deaktivieren und später wieder aktivieren, sendet Systems Manager Protokolle ab dem Zeitpunkt, an dem die Unterbrechung stattfand. Für alle benutzerdefinierten Protokolldateien und IIS- (Internet Information Services)-Protokolle liest Systems Manager die Protokolldateien von Anfang an. Darüber hinaus kann Systems Manager auch Leistungsindikatordaten an CloudWatch Logs senden.

Wenn Sie zuvor die CloudWatch Integration in EC2 Config aktiviert haben, überschreiben die Systems Manager Manager-Einstellungen alle Einstellungen, die lokal auf dem verwalteten Knoten in der `C:\Program Files\Amazon\EC2ConfigService\Settings\AWS.EC2.Windows.CloudWatch.json` Datei gespeichert sind. Weitere Informationen zur Verwendung von EC2 Config zur Verwaltung von Leistungsindikatoren und Protokollen auf einem einzelnen verwalteten Knoten finden Sie unter Erfassung von Metriken und Protokollen von EC2 Amazon-Instances und lokalen Servern mit dem CloudWatch Agenten im CloudWatch Amazon-Benutzerhandbuch.

Die Beschreibung und verfügbare Parameter anzeigen

```
Get-SSMDocumentDescription `
    -Name "AWS-ConfigureCloudWatch"
```

Weitere Informationen über Parameter anzeigen

```
Get-SSMDocumentDescription `
    -Name "AWS-ConfigureCloudWatch" | Select -ExpandProperty Parameters
```

Senden Sie Anwendungsprotokolle an CloudWatch

Mit dem folgenden Befehl wird der verwaltete Knoten konfiguriert und die Windows-Anwendungsprotokolle dorthin CloudWatch verschoben.

```
$cloudWatchCommand = Send-SSMCommand `
    -InstanceID instance-ID `
```

```
    -DocumentName "AWS-ConfigureCloudWatch" `
    -Parameter @{'properties'='{"engineConfiguration": {"PollInterval":"00:00:15",
 "Components":[{"Id":"ApplicationEventLog",
 "FullName":"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWa
 "Parameters":{"LogName":"Application", "Levels":"7"}},{"Id":"CloudWatch",
 "FullName":"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
 "Parameters":{"Region":"region", "LogGroup":"my-log-group", "LogStream":"instance-
id"}}], "Flows":{"Flows":["ApplicationEventLog,CloudWatch"]}}}'}
```

Abrufen von Befehlsinformationen pro verwalteter Knoten

Mit dem folgenden Befehl wird die `CommandId` verwendet, um den Status der Befehlsausführung abzurufen.

```
Get-SSMCommandInvocation `
    -CommandId $cloudWatchCommand.CommandId `
    -Details $true
```

Abrufen von Befehlsinformationen mit Antwortdaten für einen bestimmten verwalteten Knoten

Der folgende Befehl gibt die Ergebnisse der CloudWatch Amazon-Konfiguration zurück.

```
Get-SSMCommandInvocation `
    -CommandId $cloudWatchCommand.CommandId `
    -Details $true `
    -InstanceId instance-ID | Select -ExpandProperty CommandPlugins
```

Senden Sie Leistungsindikatoren an die CloudWatch Verwendung des Dokuments **AWS-ConfigureCloudWatch**

Mit dem folgenden Demonstrationsbefehl werden Leistungsindikatoren in hochgeladen. CloudWatch Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

```
$cloudWatchMetricsCommand = Send-SSMCommand `
    -InstanceID instance-ID `
    -DocumentName "AWS-ConfigureCloudWatch" `
    -Parameter @{'properties'='{"engineConfiguration": {"PollInterval":"00:00:15",
 "Components":[{"Id":"PerformanceCounter",
 "FullName":"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComp
 "Parameters":{"CategoryName":"Memory", "CounterName":"Available
 MBytes", "InstanceName":"", "MetricName":"AvailableMemory",
 "Unit":"Megabytes","DimensionName":"", "DimensionValue":""}},{"Id":"CloudWatch",
```

```
 "FullName":"AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.Cl
 "Parameters":{"AccessKey":"", "SecretKey":"","Region":"region", "NameSpace":"Windows-
Default"}}], "Flows":{"Flows":["PerformanceCounter,CloudWatch"]}}}'}
```

Aktivieren oder deaktivieren Sie die automatische Windows-Aktualisierung mithilfe des **`AWS-ConfigureWindowsUpdate`**-Dokuments.

Mit Run Command und dem `AWS-ConfigureWindowsUpdate`-Dokument können Sie automatische Windows-Updates auf Ihren von Windows Server verwalteten Knoten aktivieren oder deaktivieren. Mit diesem Befehl wird der Windows Update-Agent konfiguriert, um Windows-Updates an dem Tag und in der Stunde, die Sie angeben, herunterzuladen und zu installieren. Wenn ein Update einen Neustart erfordert, startet der verwaltete Knoten automatisch 15 Minuten nach der Installation der Updates neu. Mit diesem Befehl können Sie konfigurieren, dass Windows Update auf Updates prüft, diese aber nicht installiert. Das Dokument `AWS-ConfigureWindowsUpdate` wird in Windows Server 2012 und späteren Versionen offiziell unterstützt.

Die Beschreibung und verfügbare Parameter anzeigen

```
Get-SSMDocumentDescription `
    -Name "AWS-ConfigureWindowsUpdate"
```

Weitere Informationen über Parameter anzeigen

```
Get-SSMDocumentDescription `
    -Name "AWS-ConfigureWindowsUpdate" | Select -ExpandProperty Parameters
```

Aktivieren des automatischen Windows Updates

Mit dem folgenden Befehl wird Windows Update konfiguriert, um Updates automatisch täglich um 22.00 Uhr herunterzuladen und zu installieren.

```
$configureWindowsUpdateCommand = Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-ConfigureWindowsUpdate" `
    -Parameters @{'updateLevel'='InstallUpdatesAutomatically';
 'scheduledInstallDay'='Daily'; 'scheduledInstallTime'='22:00'}
```

Anzeigen des Befehlsstatus zum Aktivieren von automatischen Windows Updates

Mit dem folgenden Befehl wird die `CommandId` verwendet, um den Status der Befehlsausführung für die Aktivierung von Windows Automatic Updates abzurufen.

```
Get-SSMCommandInvocation `
    -Details $true `
    -CommandId $configureWindowsUpdateCommand.CommandId | Select -ExpandProperty
 CommandPlugins
```

Deaktivieren des automatischen Windows Updates

Mit dem folgenden Befehl wird die Windows-Update-Benachrichtigungsebene herabgesetzt, damit das System prüft, ob Updates vorliegen, diese jedoch nicht automatisch auf dem verwalteten Knoten installiert.

```
$configureWindowsUpdateCommand = Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-ConfigureWindowsUpdate" `
    -Parameters @{'updateLevel'='NeverCheckForUpdates'}
```

Anzeigen des Befehlsstatus zum Deaktivieren von automatischen Windows Updates

Mit dem folgenden Befehl wird die `CommandId` verwendet, um den Status der Befehlsausführung für die Aktivierung von Windows Automatic Updates abzurufen.

```
Get-SSMCommandInvocation `
    -Details $true `
    -CommandId $configureWindowsUpdateCommand.CommandId | Select -ExpandProperty
 CommandPlugins
```

Verwalten von Windows-Updates mit Run Command

Mit Run Command und dem `AWS-InstallWindowsUpdates`-Dokument können Sie Updates für von Windows Server verwaltete Knoten verwalten. Dieser Befehl scannt nach oder installiert fehlende Updates auf Ihren verwalteten Knoten und führt nach der Installation optional einen Neustart durch. Sie können auch die entsprechenden Klassifizierungen und Schweregrade für Aktualisierungen angeben, die in Ihrer Umgebung installiert werden sollen.

> **Note**
>
> Informationen zum Neustarten von verwalteten Knoten bei Verwendung von Run Command für den Aufruf von Skripts finden Sie unter Umgang mit Neustarts beim Ausführen von Befehlen.

Die folgenden Beispiele zeigen, wie Sie die angegebenen Windows Update-Verwaltungsaufgaben durchführen.

Suche nach allen fehlenden Windows-Updates

```
Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-InstallWindowsUpdates" `
    -Parameters @{'Action'='Scan'}
```

Installieren von bestimmten Windows-Updates

```
Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-InstallWindowsUpdates" `
    -Parameters @{'Action'='Install';'IncludeKbs'='kb-ID-1,kb-ID-2,kb-
ID-3';'AllowReboot'='True'}
```

Installieren wichtiger fehlender Windows-Updates

```
Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-InstallWindowsUpdates" `
    -Parameters @{'Action'='Install';'SeverityLevels'='Important';'AllowReboot'='True'}
```

Installieren fehlender Windows-Updates mit bestimmten Ausschlüssen

```
Send-SSMCommand `
    -InstanceId instance-ID `
    -DocumentName "AWS-InstallWindowsUpdates" `
    -Parameters @{'Action'='Install';'ExcludeKbs'='kb-ID-1,kb-
ID-2';'AllowReboot'='True'}
```

Fehlerbehebung von Systems Manager Run Command

Run Command, ein Tool in AWS Systems Manager, bietet Statusdetails zu jeder Befehlsausführung. Weitere Informationen zu den Befehlsstatus-Details finden Sie unter Grundlegendes zu Befehlsstatus. Sie können auch die Informationen in diesem Thema verwenden, um Probleme mit Run Command zu beheben.

Themen

- Einige meiner verwalteten Knoten fehlen
- Ein Schritt in meinem Skript ist fehlgeschlagen, der Gesamtstatus wird jedoch als "Succeeded" (Erfolgreich) angezeigt.
- SSM Agent wird nicht ordnungsgemäß ausgeführt.

Einige meiner verwalteten Knoten fehlen

Auf der Seite Run a command (Einen Befehl ausführen) können Sie, nachdem Sie ein auszuführendes SSM-Dokument ausgewählt und im Abschnitt Targets (Ziele) die manuelle Auswahl von Instances gewählt haben, wird eine Liste von verwalteten Knoten angezeigt, die Sie für die Ausführung des Befehls auswählen können.

Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

Nachdem Sie einen verwalteten Knoten erstellt, aktiviert, neu gestartet oder neu gestartet, Run Command auf einem Knoten installiert oder ein AWS Identity and Access Management (IAM-) Instanzprofil an einen Knoten angehängt haben, kann es einige Minuten dauern, bis der verwaltete Knoten der Liste hinzugefügt wird.

Ein Schritt in meinem Skript ist fehlgeschlagen, der Gesamtstatus wird jedoch als "Succeeded" (Erfolgreich) angezeigt.

Mit Run Command können Sie festlegen, wie Ihre Skripte mit Exit-Codes umgehen. Standardmäßig wird der Beendigungscode des letzten in einem Skript ausgeführten Befehls als Beendigungscode für das gesamte Skript gemeldet. Sie können jedoch eine bedingte Anweisung einschließen, damit das Skript beendet wird, wenn ein Befehl vor dem letzten Befehl fehlschlägt. Weitere Informationen und Beispiele finden Sie unter Angabe von Beendigungscodes in Befehlen.

SSM Agent wird nicht ordnungsgemäß ausgeführt.

Bei Problemen beim Ausführen von Befehlen mittels Run Command liegt möglicherweise ein Problem mit SSM Agent vor. Informationen zum Untersuchen von Problemen mit SSM Agent finden Sie unter Fehlerbehebung für SSM Agent.

AWS Systems Manager Session Manager

Session Managerist ein vollständig verwaltetes AWS Systems Manager Tool. Mit Session Manager können Sie Ihre Amazon Elastic Compute Cloud (Amazon EC2) -Instances, Edge-Geräte, lokalen Server und virtuellen Maschinen (VMs) verwalten. Sie können entweder eine interaktive

browserbasierte Shell mit einem Klick oder die AWS Command Line Interface () verwenden.AWS CLISession Managerbietet sicheres Knotenmanagement, ohne dass eingehende Ports geöffnet, Bastion-Hosts verwaltet oder SSH-Schlüssel verwaltet werden müssen. Session Managerermöglicht Ihnen außerdem die Einhaltung von Unternehmensrichtlinien, die einen kontrollierten Zugriff auf verwaltete Knoten, strenge Sicherheitspraktiken und Protokolle mit Knotenzugriffsdetails vorschreiben, und bietet Endbenutzern gleichzeitig einen einfachen plattformübergreifenden Zugriff mit nur einem Klick auf Ihre verwalteten Knoten. Um mit Session Manager zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Session Manager aus.

Welche Vorteile bietet Session Manager meiner Organisation?

Session Manager bietet die folgenden Vorteile:

- Zentralisierte Kontrolle des Zugriffs auf verwaltete Knoten mit IAM-Richtlinien

  Administratoren erhalten eine zentrale Stelle zum Erteilen und Widerrufen des Zugriffs auf verwaltete Knoten. Wenn Sie ausschließlich AWS Identity and Access Management (IAM-) Richtlinien verwenden, können Sie steuern, welche einzelnen Benutzer oder Gruppen in Ihrem Unternehmen Zugriff darauf haben Session Manager und auf welche verwalteten Knoten sie zugreifen können.
- Keine offenen Ports für eingehenden Datenverkehr und keine Notwendigkeit für die Verwaltung von Bastion-Hosts oder SSH-Ports

  Wenn Sie SSH-Ports für eingehenden Datenverkehr und PowerShell-Remote Ports auf Ihren verwalteten Knoten geöffnet lassen, besteht ein höheres Risiko, dass juristische Stellen nicht autorisierte oder böswillige Befehle auf den verwalteten Knoten ausführen. Session Manager unterstützt Sie bei der Verbesserung des Sicherheitsstatus, da Sie diese Ports für eingehenden Datenverkehr schließen können. Dies befreit Sie von der Notwendigkeit, SSH-Schlüssel und -Zertifikate, Bastion-Hosts und Jumpboxes verwalten zu müssen.
- One-Click-Zugriff auf verwaltete Knoten über die Konsole und die CLI

  Mit der AWS Systems Manager Konsole oder der EC2 Amazon-Konsole können Sie eine Sitzung mit einem einzigen Klick starten. Mit dem AWS CLI können Sie auch eine Sitzung starten, die einen einzelnen Befehl oder eine Befehlsfolge ausführt. Da Berechtigungen für verwaltete Knoten durch IAM-Richtlinien und nicht von SSH-Schlüsseln oder anderen Mechanismen bereitgestellt werden, wird die Verbindungszeit stark reduziert.
- Connect zu EC2 Amazon-Instances und nicht EC2 verwalteten Knoten in Hybrid- und Multi-Cloud-Umgebungen her

Sie können sich sowohl mit Amazon Elastic Compute Cloud (Amazon EC2) -Instances als auch mit EC2 Nicht-Nodes in Ihrer Hybrid- und Multi-Cloud-Umgebung verbinden.

Um eine Verbindung zu EC2 Nicht-Knoten herzustellenSession Manager, müssen Sie zunächst die Stufe Advanced-Instances aktivieren. Die Nutzung des Advanced-Instances-Kontingents ist kostenpflichtig. Es fallen jedoch keine zusätzlichen Gebühren für die Verbindung mit EC2 Instances an, die Sie verwenden. Session Manager Weitere Informationen finden Sie unter Konfigurieren von Instance-Kontingenten.

- Port-Weiterleitung

  Leiten Sie jeden Port in Ihrem verwalteten Knoten an einen lokalen Port auf einem Client um. Stellen Sie danach eine Verbindung mit dem lokalen Port her und greifen Sie auf die Serveranwendung zu, die in dem Knoten ausgeführt wird.

- Plattformübergreifende Unterstützung für Windows, Linux und macOS

  Session Manager unterstützt in einem einzigen Tool sowohl Windows, Linux als auch macOS. Sie müssen beispielsweise keinen SSH-Client für Linux- und macOS-verwaltete Knoten oder eine RDP-Verbindung für Windows Server-verwaltete Knoten verwenden.

- Protokollieren von Sitzungsaktivitäten

  Um betriebs- oder sicherheitsbezogene Anforderungen in Ihrer Organisation zu erfüllen, müssen Sie möglicherweise eine Aufzeichnung der Verbindungen bereitstellen, die mit Ihren verwalteten Knoten hergestellt wurden, und der Befehle, die auf ihnen ausgeführt wurden. Sie können auch Benachrichtigungen empfangen, wenn ein Benutzer in Ihrer Organisation Sitzungsaktivitäten startet oder beendet.

  Die Funktionen für Protokollierung werden durch die Integration mit den folgenden AWS-Services bereitgestellt:

  - AWS CloudTrail— AWS CloudTrail erfasst Informationen über Session Manager API-Aufrufe in Ihrem AWS-Konto und schreibt sie in Protokolldateien, die in einem von Ihnen angegebenen Amazon Simple Storage Service (Amazon S3) -Bucket gespeichert werden. Ein Bucket wird für alle CloudTrail Protokolle Ihres Kontos verwendet. Weitere Informationen finden Sie unter Protokollierung von AWS Systems Manager-API-Aufrufen mit AWS CloudTrail.
  - Amazon Simple Storage Service – Sie können Sitzungsprotokolldaten zu Debugging-Zwecken in einem Amazon S3-Bucket Ihrer Wahl speichern. Protokolldaten können mit oder ohne Verschlüsselung über Ihren AWS KMS key an Ihren Amazon S3-Bucket gesendet werden.

Weitere Informationen finden Sie unter Protokollieren von Sitzungsdaten mithilfe von Amazon S3 (Konsole).

- Amazon CloudWatch Logs — CloudWatch Logs ermöglicht es Ihnen, Protokolldateien aus verschiedenen Quellen zu überwachen, zu speichern und darauf zuzugreifen AWS-Services. Sie können Sitzungsprotokolldaten zu Debugging- und Fehlerbehebungszwecken an eine CloudWatch Logs-Protokollgruppe senden. Protokolldaten können mit oder ohne AWS KMS Verschlüsselung mit Ihrem KMS-Schlüssel an Ihre Protokollgruppe gesendet werden. Weitere Informationen finden Sie unter Protokollierung von Sitzungsdaten mit Amazon CloudWatch Logs (Konsole).
- Amazon EventBridge und Amazon Simple Notification Service — EventBridge ermöglicht es Ihnen, Regeln einzurichten, um zu erkennen, wann Änderungen an den von Ihnen angegebenen AWS Ressourcen vorgenommen werden. Sie können eine Regel erstellen, um zu erkennen, wenn ein Benutzer in Ihrer Organisation eine Sitzung startet oder anhält. Anschließend können Sie über Amazon SNS eine Benachrichtigung (z. B. eine Text- oder E-Mail-Nachricht) über das Ereignis erhalten. Sie können ein CloudWatch Ereignis auch so konfigurieren, dass andere Antworten ausgelöst werden. Weitere Informationen finden Sie unter Überwachen der Sitzungsaktivität mithilfe von Amazon EventBridge (Konsole).

> **Note**
>
> Protokollieren ist für Session Manager-Sitzungen, die eine Verbindung über Port-Weiterleitung oder SSH herstellen, nicht verfügbar. Dies liegt daran, dass SSH alle Sitzungsdaten innerhalb der sicheren TLS-Verbindung zwischen den Endpunkten AWS CLI und den Session Manager Endpunkten verschlüsselt und Session Manager nur als Tunnel für SSH-Verbindungen dient.

An wen richtet sich Session Manager?

- Jeder AWS Kunde, der seine Sicherheitslage verbessern, den betrieblichen Aufwand durch die Zentralisierung der Zugriffskontrolle auf verwalteten Knoten reduzieren und den eingehenden Knotenzugriff reduzieren möchte.
- Datensicherheitsexperten, die den Zugriff auf und die Aktivität von verwalteten Knoten überwachen und verfolgen möchten, Ports für eingehenden Datenverkehr auf verwalteten Knoten schließen möchten oder Verbindungen mit verwalteten Knoten ohne öffentliche IP-Adresse ermöglichen möchten.

- Administratoren, die den Zugriff über eine zentrale Stelle gewähren und widerrufen möchten und Benutzern eine einzige Lösung für von Linux, macOS und Windows Server verwaltete Knoten bereitstellen möchten.
- Benutzer, die mit nur einem Klick im Browser oder AWS CLI ohne Angabe von SSH-Schlüsseln eine Verbindung zu einem verwalteten Knoten herstellen möchten.

Was sind die Hauptfeatures von Session Manager?

- Support für von Windows Server-, Linux- und macOS- verwaltete Knoten

  Session Managerermöglicht es Ihnen, sichere Verbindungen zu Ihren Amazon Elastic Compute Cloud (EC2) -Instances, Edge-Geräten, lokalen Servern und virtuellen Maschinen (VMs) herzustellen. Eine Liste der unter den jeweiligen Betriebssystemen unterstützten Typen finden Sie unter Einrichten von Session Manager.

  > **Note**
  >
  > Session Manager-Support für On-Premises-Server wird nur für das Advanced-Instances-Kontingent bereitgestellt. Weitere Informationen finden Sie unter Aktivieren des Kontingents für erweiterte Instances.

- Zugriff auf Session Manager-Funktionen über Konsole, CLI und SDK

  Sie können Session Manager wie folgt nutzen:

  Die AWS Systems Manager -Konsole bietet sowohl Administratoren als auch Endbenutzern Zugriff auf alle Session Manager-Funktionen. Sie können über die Systems Manager-Konsole jede Aufgabe im Zusammenhang mit Ihren Sitzungen ausführen.

  Die EC2 Amazon-Konsole bietet Endbenutzern die Möglichkeit, eine Verbindung zu EC2 Instances herzustellen, für die ihnen Sitzungsberechtigungen erteilt wurden.

  Die AWS CLI beinhaltet den Zugriff auf Session Manager-Funktionen für Endbenutzer. Sie können eine Sitzung starten, eine Liste von Sitzungen anzeigen und eine Sitzung dauerhaft beenden, indem Sie die verwenden AWS CLI.

> **Note**
>
> Um die Befehle AWS CLI to run session verwenden zu können, müssen Sie Version 1.16.12 der CLI (oder höher) verwenden und das Session Manager Plugin auf Ihrem lokalen Computer installiert haben. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugin für die AWS CLI. Informationen zum Anzeigen des Plug-ins finden Sie GitHub unter. session-manager-plugin

- IAM-Zugriffskontrolle

  Mithilfe von IAM-Richtlinien können Sie steuern, welche Mitglieder Ihrer Organisation Sitzungen mit verwalteten Knoten starten können und auf welche Knoten sie zugreifen können. Sie können auch einen temporären Zugriff auf Ihre verwalteten Knoten bereitstellen. Beispielsweise könnten Sie einem Techniker auf Aufruf (oder einer Gruppe von Technikern auf Abruf) nur für die Dauer ihrer Schicht Zugriff auf Produktionsserver geben.

- Unterstützung für Protokollierung

  Session Manager stellt Ihnen für die Protokollierung von Sitzungsverläufen in Ihrem AWS-Konto durch die Integration mit einer Reihe anderer AWS-Services verschiedene Optionen bereit. Weitere Informationen erhalten Sie unter Protokollieren von Sitzungsaktivitäten und Protokollierung von Sitzungen aktivieren und deaktivieren.

- Konfigurierbare Shell-Profile

  Session Manager bietet Ihnen Optionen zum Konfigurieren von Voreinstellungen innerhalb von Sitzungen. Mit diesen anpassbaren Profilen können Sie Voreinstellungen wie Shell-Einstellungen, Umgebungsvariablen, Arbeitsverzeichnisse und das Ausführen mehrerer Befehle definieren, wenn eine Sitzung gestartet wird.

- Support für die Datenverschlüsselung mit dem Kundenschlüssel

  Sie können so konfigurierenSession Manager, dass die Sitzungsdatenprotokolle, die Sie an einen Amazon Simple Storage Service (Amazon S3) -Bucket senden oder in eine CloudWatch Logs-Protokollgruppe streamen, verschlüsselt werden. Sie können Session Manager auch so konfigurieren, dass die Daten, die zwischen Client-Maschinen und Ihren verwalteten Knoten während der Sitzungen übertragen werden, weiter verschlüsselt werden. Weitere Informationen finden Sie unter Protokollierung von Sitzungen aktivieren und deaktivieren und Konfigurieren von Sitzungspräferenzen.

- AWS PrivateLink Unterstützung für verwaltete Knoten ohne öffentliche IP-Adressen

  Sie können auch VPC-Endpunkte für Systems Manager einrichten, um Ihre Sitzungen weiter AWS PrivateLink zu sichern. AWS PrivateLink begrenzt den gesamten Netzwerkverkehr zwischen Ihren verwalteten Knoten, Systems Manager und Amazon EC2 auf das Amazon-Netzwerk. Weitere Informationen finden Sie unter Verbessern der Sicherheit von EC2 Instances mithilfe von VPC-Endpunkten für Systems Manager.

- Tunneling

  Verwenden Sie in einer Sitzung ein Dokument vom Typ Sitzung AWS Systems Manager (SSM), um Datenverkehr, z. B. HTTP oder ein benutzerdefiniertes Protokoll, zwischen einem lokalen Port auf einem Client-Computer und einem Remote-Port auf einem verwalteten Knoten zu tunneln.

- Interaktive Befehle

  Erstellen Sie ein SSM-Dokument vom Typ Session, das eine Sitzung verwendet, um interaktiv einen einzelnen Befehl auszuführen, sodass Sie verwalten können, was Benutzer auf einem verwalteten Knoten tun können.

Was ist eine Sitzung?

Eine Sitzung ist eine Verbindung zu einem verwalteten Knoten mit Session Manager. Sitzungen basieren auf einem sicheren bidirektionalen Kommunikationskanal zwischen dem Client (Sie) und dem remote verwalteten Knoten, der Eingaben und Ausgaben für Befehle streamt. Der Datenverkehr zwischen einem Client und einem verwalteten Knoten wird mit TLS 1.2 verschlüsselt. Anforderungen zum Aufbau der Verbindung werden mit Sigv4 signiert. Diese bidirektionale Kommunikation ermöglicht interaktive Bash und PowerShell den Zugriff auf verwaltete Knoten. Sie können auch einen AWS Key Management Service (AWS KMS) -Schlüssel verwenden, um Daten über die standardmäßige TLS-Verschlüsselung hinaus weiter zu verschlüsseln.

Angenommen, John ist ein Techniker auf Abruf in Ihrer IT-Abteilung. Er erhält eine Benachrichtigung zu einem Problem, für dessen Bearbeitung er eine Remote-Verbindung mit einem verwalteten Knoten herstellen muss, beispielsweise einem Ausfall, der behoben werden muss, oder eine Anweisung, eine einfache Konfigurationsoption für einen Knoten zu ändern. Mithilfe der AWS Systems Manager Konsole, der EC2 Amazon-Konsole oder der startet John eine Sitzung AWS CLI, die ihn mit dem verwalteten Knoten verbindet, führt Befehle auf dem Knoten aus, die für die Ausführung der Aufgabe erforderlich sind, und beendet dann die Sitzung.

Wenn John den Befehl zum Starten der Sitzung sendet, authentifiziert der Session Manager-Service seine ID, überprüft die ihm von einer IAM-Richtlinie gewährten Berechtigungen, prüft Konfigurationseinstellungen (z. B. die zulässigen Limits für die Sitzungen) und sendet eine Nachricht an, SSM Agent, um die Zwei-Wege-Verbindung zu öffnen. Nach der Herstellung der Verbindung und der Eingabe des nächsten Befehls durch John wird die Befehlsausgabe aus SSM Agent zu diesem Kommunikationskanal hochgeladen und zurück an Johns lokalen Computer gesendet.

Themen

- Einrichten von Session Manager
- Arbeiten mit Session Manager
- Protokollieren von Sitzungsaktivitäten
- Protokollierung von Sitzungen aktivieren und deaktivieren
- Schema des Sitzungsdokuments
- Fehlerbehebung für Session Manager

Einrichten von Session Manager

Führen Sie die Schritte in den folgenden Themen aus, bevor Sie AWS Systems Manager Session Manager für die Verbindung mit den verwalteten Knoten in Ihrem Konto verwenden.

Themen

- Schritt 1: Erfüllen der Session Manager-Voraussetzungen
- Schritt 2: Überprüfen oder Hinzufügen von Instance-Berechtigungen für Session Manager
- Schritt 3: Steuern des Sitzungs-Zugriffs auf verwaltete Knoten
- Schritt 4: Konfigurieren von Sitzungspräferenzen
- Schritt 5: (Optional) Beschränken des Zugriffs auf Befehle in einer Sitzung
- Schritt 6: (Optional) Verwenden von AWS PrivateLink zum Einrichten eines VPC-Endpunkts für Session Manager
- Schritt 7: (Optional) Deaktivieren oder Aktivieren der Administratorberechtigungen für das SSM-Benutzerkonto
- Schritt 8: (Optional) Berechtigungen für SSH-Verbindungen über Session Manager zulassen und steuern

Schritt 1: Erfüllen der Session Manager-Voraussetzungen

Stellen Sie vor der Verwendung von Session Manager sicher, dass Ihre Umgebung den folgenden Anforderungen entspricht.

Session Manager-Voraussetzungen

Anforderung	Description
Unterstützte Betriebssysteme	Session Managerunterstützt die Verbindung zu Amazon Elastic Compute Cloud (Amazon EC2) -Instances sowie zu EC2 Nicht-Maschinen in Ihrer Hybrid- und Multi-Cloud-Umgebung, die die Advanced-Instance-Stufe verwenden. Session Manager unterstützt die folgenden Betriebssystemversionen: **Note** Session Managerunterstützt EC2 Instances, Edge-Geräte sowie lokale Server und virtuelle Maschinen (VMs) in Ihrer Hybrid- und Multi-Cloud-Umgebung, die die Stufe „Advanced -Instances“ verwenden. Weitere Informationen über erweiterte Instances finden Sie unter Konfigurieren von Instance-Kontingenten. Linux und macOS Session Managerunterstützt alle Versionen von Linux undmacOS, die von unterstützt werden. AWS Systems Manager Weitere Informationen finden Sie unter Unterstützte Betriebssysteme und Maschinentypen.

Anforderung	Description
	Windows Session Manager unterstützt Windows Server 2012 und höher. **Note** Microsoft Windows Server 2016 Nano wird nicht unterstützt.

Anforderung	Description
SSM Agent	Auf den verwalteten Knoten, zu denen Sie über Sitzungen eine Verbindung herstellen möchten, muss mindestens AWS Systems Manager SSM Agent Version 2.3.68.0 oder höher installiert sein. Um die Option zum Verschlüsseln von Sitzungsdaten mithilfe eines in AWS Key Management Service (AWS KMS) erstellten Schlüssels verwenden zu können, SSM Agent muss Version 2.3.539.0 oder höher auf dem verwalteten Knoten installiert sein. Um Shell-Profile in einer Sitzung zu verwenden, muss SSM Agent Version 3.0.161.0 oder höher auf dem verwalteten Knoten installiert sein. Um eine Session Manager-Port-Weiterleitung oder SSH-Sitzung zu starten, muss SSM Agent Version 3.0.222.0 oder höher auf dem verwaltet en Knoten installiert sein. Um Sitzungsdaten mit Amazon CloudWatch Logs zu streamen, muss SSM Agent Version 3.0.284.0 oder höher auf dem verwalteten Knoten installiert sein. Informationen zum Ermitteln der auf einer Instance ausgeführten Versionsnummer finden Sie unter Überprüfen der SSM Agent-Ver sionsnummer. Informationen über das manuelle Installieren oder automatische Aktualisieren von SSM Agent finden Sie unter Arbeiten mit SSM Agent. Über das ssm-user-Konto

Anforderung	Description
	Beginnend mit Version 2.3.50.0 von SSM Agent erstellt der Agent unter Verwendung der Root- oder Administratorberechtigungen ein Benutzerkonto auf dem verwalteten Knoten, das `ssm-user` genannt wird. (Auf Versionen vor 2.3.612.0 wird das Konto erstellt, wenn SSM Agent startet oder neu startet. Auf Version 2.3.612.0 und höher wird `ssm-user` erstellt, wenn eine Sitzung auf dem verwalteten Knoten zum ersten Mal gestartet wird.) Sitzungen werden mittels der Anmeldeinformationen für dieses Benutzerkonto gestartet. Weitere Informationen zum Einschränken der administrativen Kontrolle für dieses Konto finden Sie unter Deaktivieren oder Aktivieren der Administratorberechtigungen für das SSM-Benutzerkonto. ssm-user auf Windows Server-Domain-Controller Ab SSM Agent Version 2.3.612.0 wird das `ssm-user`-Konto nicht automatisch auf verwalteten Knoten erstellt, die als Windows Server-Domain-Controller verwendet werden. Um Session Manager auf einer Windows Server-Maschine zu verwenden, die als Domain-Controller verwendet wird, erstellen Sie das `ssm-user`-Konto manuell, sofern es noch nicht vorhanden ist, und weisen dem Benutzer Domain-Administratorberechtigungen zu. Unter Windows Server legt SSM Agent bei jedem Start einer Sitzung ein neues Passwort für das `ssm-user`-Konto fest. Es ist also nicht erforderl

Anforderung	Description
	ich, ein Passwort anzugeben, wenn Sie das Konto erstellen.
Konnektivität mit Endpunkten	In diesem Fall müssen die verwalteten Knoten auch ausgehenden HTTPS-Datenverkehr (Port 443) zu den folgenden Endpunkten zulassen: • ec2-Nachrichten. *region*.amazonaws.com • ssm. *region*.amazonaws.com • SMS-Nachrichten. *region*.amazonaws.com Weitere Informationen finden Sie unter den folgenden Themen: • Referenz: ec2messages, ssmmessages und andere API-Operationen • Wie erstelle ich VPC-Endpoints, sodass ich Systems Manager verwenden kann, um private EC2 Instances ohne Internetzugang zu verwalten? im AWS re:Post Knowledge Center. Alternativ können Sie sich über Schnittst ellenendpunkte mit den erforderlichen Endpunkten verbinden. Weitere Informati onen finden Sie unter Schritt 6: (Optional) Verwenden von AWS PrivateLink zum Einrichte n eines VPC-Endpunkts für Session Manager.

Anforderung	Description
AWS CLI	(Optional) Wenn Sie AWS Command Line Interface (AWS CLI) verwenden, um Ihre Sitzungen zu starten (anstatt die AWS Systems Manager Konsole oder die EC2 Amazon-Konsole zu verwenden), muss Version 1.16.12 oder höher der CLI auf Ihrem lokalen Computer installiert sein. Zum Überprüfen der Version können Sie den Befehl `aws --version` aufrufen. Wenn Sie die CLI installieren oder aktualisieren müssen, finden Sie weitere Informationen unter Installation von AWS Command Line Interface im AWS Command Line Interface Benutzerhandbuch. **Important** Wenn Systems Manager neue Tools hinzugefügt oder Aktualisierungen an den vorhandenen Tools vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Tools und Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent in Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonniere

Anforderung	Description
	n Sie die SSM Agent-Versionshinweise-Seite in GitHub, um Benachrichtigungen über SSM Agent-Updates zu erhalten. Um die CLI zur Verwaltung Ihrer Knoten mit Session Manager verwenden zu können, müssen Sie zunächst das Session Manager-Plugin auf Ihrer lokalen Maschine installieren. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugin für die AWS CLI.
Aktivieren des Advanced-Instances-Kontingents (Hybrid- und Multi-Cloud-Umgebungen)	Um eine Verbindung zu EC2 Computern herzustellen, die keine Computer verwenden Session Manager, müssen Sie die Stufe Advanced-Instances in der Kategorie AWS-Konto und AWS-Region dort, wo Sie Hybrid-Aktivierungen erstellen, aktivieren, um EC2 Maschinen ohne Computer als verwaltete Knoten zu registrieren. Die Nutzung des Advanced-Instances-Kontingents ist kostenpflichtig. Weitere Informationen zum Aktivieren des Advanced-Instances-Kontingent finden Sie unter Konfigurieren von Instance-Kontingenten.

Anforderung	Description
Überprüfen der Berechtigungen für IAM-Servicerollen (Hybrid- und Multi-Cloud-Umgebungen)	Hybrid-aktivierte Knoten verwenden die in der Hybrid-Aktivierung angegebene AWS Identity and Access Management (IAM) -Servicerolle, um mit Systems Manager Manager-API-Vorgängen zu kommunizieren. Diese Servicerolle muss die Berechtigungen enthalten, die zum Herstellen einer Verbindung mit Ihren Hybrid- und Multi-Cloud-Maschinen mit Session Manager erforderlich sind. Wenn Ihre Servicerolle die AWS verwaltete Richtlinie enthält`AmazonSSMManagedInstanceCore`, Session Manager sind die erforderlichen Berechtigungen für bereits bereitgestellt. Wenn Sie feststellen, dass die Servicerolle nicht die erforderlichen Berechtigungen enthält, müssen Sie die verwaltete Instance abmelden und sie bei einer neuen Hybrid-Aktivierung registrieren, die eine IAM-Servicerolle mit den erforderlichen Berechtigungen verwendet. Informationen über das Abmelden verwalteter Instances finden Sie unter Aufheben der Registrierung von verwalteten Knoten in einer Hybrid- und Multi-Cloud-Umgebung. Weitere Informationen zum Erstellen von IAM-Richtlinien mit Session Manager-Berechtigungen finden Sie unter Schritt 2: Überprüfen oder Hinzufügen von Instance-Berechtigungen für Session Manager.

Schritt 2: Überprüfen oder Hinzufügen von Instance-Berechtigungen für Session Manager

Standardmäßig hat AWS Systems Manager keine Berechtigung zur Ausführung von Aktionen auf Ihren Instances. Sie können Instance-Berechtigungen auf Kontoebene mithilfe einerAWS Identity and Access Management (IAM)-Rolle oder auf Instance-Ebene mithilfe eines Instance-

Profils bereitstellen. Wenn Ihr Anwendungsfall dies zulässt, empfehlen wir, mithilfe der Standardkonfiguration für die Host-Verwaltung Zugriff auf Kontoebene zu gewähren. Wenn Sie die Standardkonfiguration für die Host-Verwaltung für Ihr Konto bereits mithilfe der `AmazonSSMManagedEC2InstanceDefaultPolicy`-Richtlinie eingerichtet haben, können Sie mit dem nächsten Schritt fortfahren. Weitere Informationen über die Standardkonfiguration für die Host-Verwaltung finden Sie unter Automatisches Verwalten von EC2 Instanzen mit der Standard-Host-Management-Konfiguration.

Alternativ können Sie auch Instance-Profile verwenden, um Ihren Instances die erforderlichen Berechtigungen zu erteilen. Ein Instance-Profil übergibt eine IAM-Rolle an eine Amazon-EC2-Instance. Sie können ein IAM-Instance-Profil einer Amazon-EC2-Instance beim Starten anfügen oder einer zuvor gestarteten Instance anfügen. Weitere Informationen finden Sie unter Verwenden von Instance-Profilen.

Für On-Premises-Server oder virtuelle Maschinen (VMs) werden Berechtigungen von der IAM-Servicerolle bereitgestellt, die der Hybrid-Aktivierung zugeordnet ist, die zum Anmelden Ihrer On-Premises-Server und VMs bei Systems Manager verwendet wird. On-Premises-Server und VMs verwenden keine Instance-Profile.

Wenn Sie bereits andere Systems-Manager-Tools wie Run Command oder Parameter Store verwenden, ist Ihren Amazon-EC2-Instances möglicherweise bereits ein Instance-Profil mit den für Session Manager erforderlichen Grundberechtigungen angefügt. Wenn Ihren Instances bereits ein Instance-Profil angefügt ist, das die AWS-verwaltete Richtlinie `AmazonSSMManagedInstanceCore` enthält, sind die erforderlichen Berechtigungen für Session Manager bereits vorhanden. Dies gilt auch, wenn die bei Ihrer Hybrid-Aktivierung verwendete IAM-Servicerolle die verwaltete Richtlinie `AmazonSSMManagedInstanceCore` enthält.

In einigen Fällen müssen Sie jedoch möglicherweise die Berechtigungen ändern, die Ihrem Instance-Profil zugeordnet sind. Sie möchten beispielsweise eine engere Gruppe von Instance-Berechtigungen bereitstellen, Sie haben eine benutzerdefinierte Richtlinie für Ihr Instance-Profil erstellt oder Sie möchten Amazon Simple Storage Service (Amazon S3)-Verschlüsselung oder AWS Key Management Service (AWS KMS)-Verschlüsselungsoptionen zum Sichern von Sitzungsdaten verwenden. Führen Sie in diesen Fällen einen der folgenden Schritte aus, um Session Manager-Aktionen auf Ihren Instances auszuführen:

- Einbetten von Berechtigungen für Session Manager-Aktionen in einer benutzerdefinierten IAM-Rolle

Um einer vorhandenen IAM-Rolle, die die von AWS bereitgestellte Standardrichtlinie `AmazonSSMManagedInstanceCore` nicht verwendet, Berechtigungen für Session Manager-Aktionen hinzuzufügen, befolgen Sie die Schritte in Session Manager-Berechtigungen für eine vorhandene IAM-Rolle hinzufügen.

- Erstellen einer benutzerdefinierten IAM-Rolle, die ausschließlich Session Manager-Berechtigungen besitzt

  Um eine IAM-Rolle zu erstellen, die ausschließlich Berechtigungen für Session Manager-Aktionen enthält, befolgen Sie die Schritte in Erstellen einer benutzerdefinierten IAM-Rolle für Session Manager.

- Erstellen und Verwenden einer neuen IAM-Rolle mit Berechtigungen für alle Systems-Manager-Aktionen

  Um eine IAM-Rolle für von Systems Manager verwaltete Instances zu erstellen, die eine von AWS bereitgestellte Standardrichtlinie verwendet, die alle Systems-Manager-Berechtigungen erteilt, befolgen Sie die Schritte unter Konfigurieren von erforderlichen IAM-Instance-Berechtigungen für Systems Manager.

Themen

- Session Manager-Berechtigungen für eine vorhandene IAM-Rolle hinzufügen
- Erstellen einer benutzerdefinierten IAM-Rolle für Session Manager

Session Manager-Berechtigungen für eine vorhandene IAM-Rolle hinzufügen

Gehen Sie wie folgt vor, um einer vorhandenen AWS Identity and Access Management (IAM)-Rolle Session Manager-Berechtigungen hinzuzufügen. Durch das Hinzufügen von Berechtigungen zu einer vorhandenen Rolle können Sie die Sicherheit Ihrer Computerumgebung verbessern, ohne die AWS `AmazonSSMManagedInstanceCore` Richtlinie für Instanzberechtigungen verwenden zu müssen.

> **Note**
>
> Notieren Sie die folgenden Informationen:
>
> - Dieses Verfahren setzt voraus, dass Ihre vorhandene Rolle bereits andere Systems-Manager-`ssm`-Berechtigungen für Aktionen enthält, für die Sie den Zugriff erlauben

möchten. Diese Richtlinie reicht allein nicht aus, um Session Manager verwenden zu können.

- Das folgende Richtlinienbeispiel beinhaltet eine `s3:GetEncryptionConfiguration`-Aktion. Diese Aktion ist erforderlich, wenn Sie in den Session Manager-Protokollierungseinstellungen die Option S3-Protokollverschlüsselung erzwingen gewählt haben.
- Wenn die `ssmmessages:OpenControlChannel` Berechtigung aus den Richtlinien entfernt wird, die mit Ihrem IAM-Instanzprofil oder Ihrer IAM-Dienstrolle verknüpft sind, SSM Agent verliert der verwaltete Knoten die Konnektivität zum Systems Manager Manager-Dienst in der Cloud. Es kann jedoch bis zu 1 Stunde dauern, bis eine Verbindung beendet wird, nachdem die Berechtigung entfernt wurde. Dies ist dasselbe Verhalten wie beim Löschen der IAM-Instanzrolle oder der IAM-Servicerolle.

So fügen Sie Session Manager-Berechtigungen einer vorhandenen Rolle hinzu (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im Navigationsbereich Rollen.
3. Wählen Sie den Namen der Rolle aus, zu der Sie die Berechtigungen hinzufügen möchten.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie Berechtigungen hinzufügen und dann Eingebundene Richtlinie hinzufügen aus.
6. Wählen Sie den Tab JSON.
7. Ersetzen Sie den Inhalt der Standardrichtlinie durch den folgenden Inhalt. *key-name*Ersetzen Sie es durch den Amazon-Ressourcennamen (ARN) des AWS Key Management Service Schlüssels (AWS KMS key), den Sie verwenden möchten.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssmmessages:CreateControlChannel",
```

```
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetEncryptionConfiguration"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-name"
        }
    ]
}
```

Weitere Informationen über die Verwendung eines KMS-Schlüssels zum Verschlüsseln von Sitzungsdaten finden Sie unter So aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole).

Wenn Sie keine AWS KMS Verschlüsselung für Ihre Sitzungsdaten verwenden, können Sie den folgenden Inhalt aus der Richtlinie entfernen.

```
,
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "key-name"
        }
```

8. Wählen Sie Weiter: Tags aus.

9. (Optional) Fügen Sie Tags hinzu, indem Sie Tag hinzufügen auswählen und die bevorzugten Tags für die Richtlinie eingeben.
10. Wählen Sie Weiter: Prüfen aus.
11. Geben Sie auf der Seite Richtlinie prüfen im Feld Name einen Namen für die Inline-Richtlinie ein, z. B. **`SessionManagerPermissions`**.
12. (Optional) Geben Sie im Feld Beschreibung eine Beschreibung für die Richtlinie ein.

    Wählen Sie Richtlinie erstellen aus.

Weitere Informationen über die `ssmmessages`-Aktionen finden Sie unter Referenz: ec2messages, ssmmessages und andere API-Operationen.

Erstellen einer benutzerdefinierten IAM-Rolle für Session Manager

Sie können eine AWS Identity and Access Management (IAM-) Rolle erstellen, die Ihnen Session Manager die Berechtigung erteilt, Aktionen auf Ihren von Amazon EC2 verwalteten Instances durchzuführen. Sie können auch eine Richtlinie hinzufügen, um die Berechtigungen zu gewähren, die für das Senden von Sitzungsprotokollen an Amazon Simple Storage Service (Amazon S3) und Amazon CloudWatch Logs erforderlich sind.

Nachdem Sie die IAM-Rolle erstellt haben, finden Sie Informationen dazu, wie Sie die Rolle an eine Instance anhängen oder ersetzen können, auf der AWS re:Post Website unter Ein Instance-Profil anhängen oder ersetzen. Weitere Informationen zu IAM-Instance-Profilen und -Rollen finden Sie unter Verwenden von Instance-Profilen im IAM-Benutzerhandbuch und IAM-Rollen für Amazon EC2 im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances. Weitere Informationen zum Erstellen einer IAM-Servicerolle für On-Premises-Maschinen finden Sie unter Erstellen der für Systems Manager erforderlichen IAM-Servicerolle in Hybrid- und Multi-Cloud-Umgebungen.

Themen

- Erstellen einer IAM-Rolle mit geringstmöglichen Session Manager-Berechtigungen (Konsole)
- Erstellen einer IAM-Rolle mit Berechtigungen für Amazon S3 Session Manager und CloudWatch Logs (Konsole)

Erstellen einer IAM-Rolle mit geringstmöglichen Session Manager-Berechtigungen (Konsole)

Verwenden Sie das folgende Verfahren, um eine benutzerdefinierte IAM-Rolle mit einer Richtlinie zu erstellen, die ausschließlich Berechtigungen für Session Manager-Aktionen auf Ihren Instances bereitstellt.

So erstellen Sie ein Instance-Profil mit den geringstmöglichen Session Manager-Berechtigungen (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im Navigationsbereich Richtlinien und dann Richtlinie erstellen. (Wenn die Schaltfläche Get Started (Erste Schritte) angezeigt wird, klicken Sie darauf und wählen Sie anschließend Create Policy (Richtlinie erstellen) aus.)
3. Wählen Sie den Tab JSON.
4. Ersetzen Sie den Standardinhalt durch folgende Richtlinie. Um Sitzungsdaten mit AWS Key Management Service (AWS KMS) zu verschlüsseln, ersetzen Sie *key-name* sie durch den Amazon-Ressourcennamen (ARN) der AWS KMS key , die Sie verwenden möchten.

> **Note**
>
> Wenn die `ssmmessages:OpenControlChannel` Berechtigung aus den Richtlinien entfernt wird, die mit Ihrem IAM-Instanzprofil oder Ihrer IAM-Dienstrolle verknüpft sind, SSM Agent verliert der verwaltete Knoten die Konnektivität zum Systems Manager Manager-Dienst in der Cloud. Es kann jedoch bis zu 1 Stunde dauern, bis eine Verbindung beendet wird, nachdem die Berechtigung entfernt wurde. Dies ist dasselbe Verhalten wie beim Löschen der IAM-Instanzrolle oder der IAM-Servicerolle.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:UpdateInstanceInformation",
```

```
                "ssmmessages:CreateControlChannel",
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-name"
        }
    ]
}
```

Weitere Informationen über die Verwendung eines KMS-Schlüssels zum Verschlüsseln von Sitzungsdaten finden Sie unter So aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole).

Wenn Sie keine AWS KMS Verschlüsselung für Ihre Sitzungsdaten verwenden, können Sie den folgenden Inhalt aus der Richtlinie entfernen.

```
,
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "key-name"
        }
```

5. Wählen Sie Weiter: Tags aus.
6. (Optional) Fügen Sie Tags hinzu, indem Sie Tag hinzufügen auswählen und die bevorzugten Tags für die Richtlinie eingeben.
7. Wählen Sie Weiter: Prüfen aus.
8. Geben Sie auf der Seite Richtlinie prüfen im Feld Name einen Namen für die Inline-Richtlinie ein, z. B. **`SessionManagerPermissions`**.
9. (Optional) Geben Sie im Feld Beschreibung eine Beschreibung für die Richtlinie ein.

10. Wählen Sie Richtlinie erstellen aus.
11. Wählen Sie im Navigationsbereich Rollen und dann Rolle erstellen.
12. Wählen Sie auf der Seite Rolle erstellen die Option AWS Dienst und für Anwendungsfall die Option EC2.
13. Wählen Sie Weiter aus.
14. Aktivieren Sie auf der Seite Attached permissions policy (Richtlinie für angefügte Berechtigungen) das Kontrollkästchen links neben dem Namen der Richtlinie, die Sie gerade erstellt haben, z. B. **`SessionManagerPermissions`**.
15. Wählen Sie Weiter aus.
16. Geben Sie auf der Seite Name, review, and create (Benennen, überprüfen und erstellen) für Role name (Rollenname) einen Namen für die IAM-Rolle ein, z. B. **`MySessionManagerRole`**.
17. (Optional) Geben Sie in Role description (Beschreibung der Rolle) eine Beschreibung für das Instance-Profil ein.
18. (Optional) Fügen Sie Tags hinzu, indem SieAdd tag (Tag hinzufügen) auswählen und die bevorzugten Tags für die Richtlinie eingeben.

    Wählen Sie Rolle erstellen aus.

Weitere Informationen zu `ssmmessages`-Aktionen finden Sie unter Referenz: ec2messages, ssmmessages und andere API-Operationen.

Erstellen einer IAM-Rolle mit Berechtigungen für Amazon S3 Session Manager und CloudWatch Logs (Konsole)

Verwenden Sie das folgende Verfahren, um eine benutzerdefinierte IAM-Rolle mit einer Richtlinie zu erstellen, die Berechtigungen für Session Manager-Aktionen auf Ihren Instances bereitstellt. Die Richtlinie bietet auch die erforderlichen Berechtigungen für die Speicherung von Sitzungsprotokollen in Amazon Simple Storage Service (Amazon S3) -Buckets und Amazon CloudWatch Logs-Protokollgruppen.

> **Important**
>
> Um Sitzungsprotokolle an einen Amazon S3-Bucket auszugeben, der zu einem anderen AWS-Konto gehört, müssen Sie die `s3:PutObjectAcl`-Berechtigung dieser IAM-Rollen-Richtlinie hinzufügen. Außerdem müssen Sie sicherstellen, dass die Bucket-Richtlinie kontenübergreifenden Zugriff auf die IAM-Rolle gewährt, die vom besitzenden Konto

verwendet wird, um dem Systems Manager Berechtigungen für verwaltete Instances zu gewähren. Wenn der Bucket die Verschlüsselung des Key Management Service (KMS) verwendet, muss die KMS-Richtlinie des Buckets diesen kontoübergreifenden Zugriff ebenfalls gewähren. Weitere Informationen zur Konfiguration von kontoübergreifenden Bucket-Berechtigungen in Amazon S3 finden Sie unter Gewährung von kontoübergreifenden Bucket-Berechtigungen im Benutzerhandbuch zu Amazon Simple Storage Service. Wenn die kontoübergreifenden Berechtigungen nicht hinzugefügt werden, kann das Konto, das Eigentümer des Amazon-S3-Buckets ist, nicht auf die Sitzungsausgabeprotokolle zugreifen.

Informationen zum Angeben von Präferenzen für das Speichern von Sitzungsprotokollen finden Sie unter Protokollierung von Sitzungen aktivieren und deaktivieren.

So erstellen Sie eine IAM-Rolle mit Berechtigungen für Session Manager Amazon S3 und CloudWatch Logs (Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im Navigationsbereich Richtlinien und dann Richtlinie erstellen. (Wenn die Schaltfläche Get Started (Erste Schritte) angezeigt wird, klicken Sie darauf und wählen Sie anschließend Create Policy (Richtlinie erstellen) aus.)
3. Wählen Sie den Tab JSON.
4. Ersetzen Sie den Standardinhalt durch folgende Richtlinie. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssmmessages:CreateControlChannel",
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel",
                "ssm:UpdateInstanceInformation"
```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/s3-prefix/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetEncryptionConfiguration"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-name"
        },
        {
            "Effect": "Allow",
            "Action": "kms:GenerateDataKey",
            "Resource": "*"
        }
    ]
}
```

5. Wählen Sie Weiter: Tags aus.

6. (Optional) Fügen Sie Tags hinzu, indem Sie Tag hinzufügen auswählen und die bevorzugten Tags für die Richtlinie eingeben.
7. Wählen Sie Weiter: Prüfen aus.
8. Geben Sie auf der Seite Richtlinie prüfen im Feld Name einen Namen für die Inline-Richtlinie ein, z. B. **`SessionManagerPermissions`**.
9. (Optional) Geben Sie im Feld Beschreibung eine Beschreibung für die Richtlinie ein.
10. Wählen Sie Richtlinie erstellen aus.
11. Wählen Sie im Navigationsbereich Rollen und dann Rolle erstellen.
12. Wählen Sie auf der Seite Rolle erstellen die Option AWS Service und für Anwendungsfall die Option EC2.
13. Wählen Sie Weiter aus.
14. Aktivieren Sie auf der Seite Attached permissions policy (Richtlinie für angefügte Berechtigungen) das Kontrollkästchen links neben dem Namen der Richtlinie, die Sie gerade erstellt haben, z. B. **`SessionManagerPermissions`**.
15. Wählen Sie Weiter aus.
16. Geben Sie auf der Seite Name, review, and create (Benennen, überprüfen und erstellen) für Role name (Rollenname) einen Namen für die IAM-Rolle ein, z. B. **`MySessionManagerRole`**.
17. (Optional) Geben Sie im Feld Role description (Rollenbeschreibung) eine Beschreibung für die Rolle ein.
18. (Optional) Fügen Sie Tags hinzu, indem SieAdd tag (Tag hinzufügen) auswählen und die bevorzugten Tags für die Richtlinie eingeben.
19. Wählen Sie Rolle erstellen aus.

Schritt 3: Steuern des Sitzungs-Zugriffs auf verwaltete Knoten

Sie gewähren oder entziehen den Session Manager-Zugriff auf verwaltete Knoten mit Hilfe von AWS Identity and Access Management (IAM)-Richtlinien. Sie können eine Richtlinie erstellen und sie einem IAM-Benutzer oder einer IAM-Gruppe zuordnen, die festlegt, mit welchen verwalteten Knoten sich der Benutzer oder die Gruppe verbinden kann. Sie können auch die Session Manager-API-Operationen festlegen, die der Benutzer oder die Gruppe auf diesen verwalteten Knoten durchführen kann.

Um Ihnen den Einstieg in die IAM-Berechtigungsrichtlinien für Session Manager zu erleichtern, haben wir Beispielrichtlinien für einen Endbenutzer und einen Administrator erstellt. Sie können diese

Richtlinien mit nur geringfügigen Änderungen verwenden. Oder verwenden Sie sie als Leitfaden für die Erstellung benutzerdefinierter IAM-Richtlinien. Weitere Informationen finden Sie unter Muster-IAM-Richtlinien für Session Manager. Informationen dazu, wie Sie IAM-Richtlinien erstellen und diese Benutzern oder Gruppen anfügen, finden Sie unter Erstellen von IAM-Richtlinien und Hinzufügen und Entfernen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Über Sitzungs-ID-ARN-Formate

Beim Erstellen einer IAM-Richtlinie für den Session Manager-Zugriff geben Sie eine Sitzungs-ID als Teil des Amazon-Ressourcennamens (ARN) an. Die Sitzungs-ID enthält den Benutzernamen als Variable. Um dies zu veranschaulichen, finden Sie hier das Format eines Session Manager-ARN und ein Beispiel:

```
arn:aws:ssm:region-id:account-id:session/session-id
```

Zum Beispiel:

```
arn:aws:ssm:us-east-2:123456789012:session/JohnDoe-1a2b3c4d5eEXAMPLE
```

Weitere Informationen zur Verwendung von Variablen in IAM-Richtlinien finden Sie unter IAM-Richtlinienelemente: Variablen.

Themen

- Starten Sie eine Standard-Shell-Sitzung, indem Sie das Standard-Sitzungsdokument in den IAM-Richtlinien angeben
- Starten Sie eine Sitzung mit einem Dokument, indem Sie die Sitzungsdokumente in IAM-Richtlinien angeben
- Muster-IAM-Richtlinien für Session Manager
- Zusätzliche IAM-Beispielrichtlinien für Session Manager

Starten Sie eine Standard-Shell-Sitzung, indem Sie das Standard-Sitzungsdokument in den IAM-Richtlinien angeben

Wenn Sie die Konfiguration Session Manager für Ihre Sitzung vornehmen AWS-Konto oder wenn Sie die Sitzungseinstellungen in der Systems Manager Manager-Konsole ändern, erstellt das System ein SSM-Sitzungsdokument mit dem Namen`SSM-SessionManagerRunShell`. Dies ist das Standard-

Sitzungsdokument. Session Manager verwendet dieses Dokument, um Ihre Sitzungseinstellungen zu speichern, die Informationen wie die folgenden enthalten:

- Ein Ort, an dem Sie Sitzungsdaten speichern möchten, z. B. ein Amazon Simple Storage Service (Amazon S3) -Bucket oder eine Amazon CloudWatch Logs-Protokollgruppe.
- Eine AWS Key Management Service (AWS KMS) Schlüssel-ID zum Verschlüsseln von Sitzungsdaten.
- Ob die Unterstützung von Run As für Ihre Sitzungen erlaubt ist.

Hier sehen Sie ein Beispiel für die Informationen, die im `SSM-SessionManagerRunShell`-Dokument Sitzungseinstellungen enthalten sind.

```
{
  "schemaVersion": "1.0",
  "description": "Document to hold regional settings for Session Manager",
  "sessionType": "Standard_Stream",
  "inputs": {
    "s3BucketName": "amzn-s3-demo-bucket",
    "s3KeyPrefix": "MyS3Prefix",
    "s3EncryptionEnabled": true,
    "cloudWatchLogGroupName": "MyCWLogGroup",
    "cloudWatchEncryptionEnabled": false,
    "kmsKeyId": "1a2b3c4d",
    "runAsEnabled": true,
    "runAsDefaultUser": "RunAsUser"
  }
}
```

Standardmäßig verwendet Session Manager das Standard-Sitzungsdokument, wenn ein Benutzer eine Sitzung von AWS-Managementkonsole aus startet. Dies gilt entweder für Fleet Manager oder Session Manager in der Systems Manager Manager-Konsole oder EC2 Connect in der EC2 Amazon-Konsole. Session Managerverwendet auch das Standardsitzungsdokument, wenn ein Benutzer eine Sitzung mit einem AWS CLI Befehl wie dem folgenden Beispiel startet:

```
aws ssm start-session \
    --target i-02573cafcfEXAMPLE
```

Um eine Standard-Shell-Sitzung zu starten, müssen Sie das Standard-Sitzungsdokument in der IAM-Richtlinie angeben, wie im folgenden Beispiel gezeigt.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "EnableSSMSession",
      "Effect": "Allow",
      "Action": [
        "ssm:StartSession"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:111122223333:instance/instance-id",
        "arn:aws:ssm:us-east-1:111122223333:document/SSM-SessionManagerRunShell"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssmmessages:OpenDataChannel"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Starten Sie eine Sitzung mit einem Dokument, indem Sie die Sitzungsdokumente in IAM-Richtlinien angeben

Wenn Sie den AWS CLI -Befehl start-session mit dem Standard-Sitzungsdokument verwenden, können Sie den Dokumentnamen auslassen. Das System ruft automatisch das `SSM-SessionManagerRunShell`-Sitzungsdokument auf.

In allen anderen Fällen müssen Sie einen Wert für den `document-name`-Parameter angeben. Wenn ein Benutzer den Namen eines Sitzungsdokuments in einem Befehl angibt, überprüft das System seine IAM-Richtlinie, um sicherzustellen, dass er berechtigt ist, auf das Dokument zuzugreifen. Wenn sie nicht berechtigt sind, schlägt die Verbindungsanforderung fehl. In den folgenden Beispielen ist

der `document-name`-Parameter im `AWS-StartPortForwardingSession`-Sitzungsdokument enthalten.

```
aws ssm start-session \
    --target i-02573cafcfEXAMPLE \
    --document-name AWS-StartPortForwardingSession \
    --parameters '{"portNumber":["80"], "localPortNumber":["56789"]}'
```

Ein Beispiel für die Angabe eines Session Manager-Sitzungsdokuments in einer IAM-Richtlinie finden Sie unter Kurzeinführung in Endbenutzerrichtlinien für Session Manager.

> **Note**
>
> Um eine Sitzung mit SSH zu starten, müssen Sie die Konfigurationsschritte auf dem verwalteten Zielknoten and der lokalen Maschine des Benutzers ausführen. Informationen finden Sie unter (Optional) Berechtigungen für SSH-Verbindungen über Session Manager zulasen und steuern.

Muster-IAM-Richtlinien für Session Manager

Verwenden Sie die Beispiele in diesem Abschnitt, um Ihnen bei der Erstellung von AWS Identity and Access Management (IAM-) Richtlinien zu helfen, die die am häufigsten benötigten Zugriffsberechtigungen Session Manager bereitstellen.

> **Note**
>
> Sie können auch eine AWS KMS key Richtlinie verwenden, um zu kontrollieren, welche IAM-Entitäten (Benutzer oder Rollen) Zugriff auf Ihren KMS-Schlüssel erhalten. AWS-Konten Weitere Informationen finden Sie im AWS Key Management Service Entwicklerhandbuch unter Überblick über die Verwaltung des Zugriffs auf Ihre AWS KMS Ressourcen und die Verwendung wichtiger Richtlinien. AWS KMS

Themen

- Kurzeinführung in Endbenutzerrichtlinien für Session Manager
- Kurzeinführung in Administratorrichtlinien für Session Manager

Kurzeinführung in Endbenutzerrichtlinien für Session Manager

Verwenden Sie die folgenden Beispiele, um IAM-Endbenutzerrichtlinien für Session Manager zu erstellen.

Sie können eine Richtlinie erstellen, die es Benutzern ermöglicht, Sitzungen nur von der Session Manager Konsole und AWS Command Line Interface (AWS CLI), nur von der Amazon Elastic Compute Cloud (Amazon EC2) -Konsole oder von allen drei aus zu starten.

Diese Richtlinien bieten Endbenutzern die Möglichkeit, eine Sitzung zu einem bestimmten verwalteten Knoten zu starten und nur ihre eigenen Sitzungen zu beenden. Beispiele für Anpassungen, die Sie möglicherweise für die Richtlinie ausführen sollten, finden Sie unter Zusätzliche IAM-Beispielrichtlinien für Session Manager.

Ersetzen Sie in den folgenden Beispielrichtlinien jede *example resource placeholder* durch Ihre eigenen Informationen.

Lesen Sie die folgenden Abschnitte, um Beispielrichtlinien für den Bereich des Sitzungszugriffs anzuzeigen, den Sie bereitstellen möchten.

Session Manager and Fleet Manager

Verwenden Sie diese Beispielrichtlinie für Provider-Benutzer, die Sitzungen nur über die Session Manager- und die Fleet Manager-Konsolen starten und wiederaufnehmen können.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:us-
east-1:111122223333:instance/i-02573cafcfEXAMPLE",
                "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
            ]
```

```
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeSessions",
                "ssm:GetConnectionStatus",
                "ssm:DescribeInstanceProperties",
                "ec2:DescribeInstances"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-name"
        }
    ]
}
```

Amazon EC2

Verwenden Sie diese Beispielrichtlinie, um Benutzern die Möglichkeit zu geben, Sitzungen nur von der EC2 Amazon-Konsole aus zu starten und fortzusetzen. Diese Richtlinie bietet nicht alle Berechtigungen, die zum Starten von Sitzungen über die Session ManagerKonsole und die AWS CLI Konsole erforderlich sind.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession",
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:us-
east-1:111122223333:instance/i-02573cafcfEXAMPLE",
                "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
            ]
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetConnectionStatus",
                "ssm:DescribeInstanceInformation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:username}-*"
            ]
        }
    ]
```

```
}
```

AWS CLI

Verwenden Sie diese Beispielrichtlinie für Provider-Benutzer, die Sitzungen nur über die AWS CLI starten und wiederaufnehmen können.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession",
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:us-
east-1:111122223333:instance/i-02573cafcfEXAMPLE",
                "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
            ]
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-name"
        }
    ]
}
```

> **Note**
>
> `SSM-SessionManagerRunShell` ist der Standardname des SSM-Dokuments, das Session Manager zum Speichern Ihrer Sitzungskonfiguration erstellt. Sie können stattdessen ein benutzerdefiniertes Sitzungsdokument erstellen und es in dieser Richtlinie angeben. Sie können auch das von AWS bereitgestellte Dokument `AWS-StartSSHSession` für Benutzer verwenden, die Sitzungen mit SSH starten. Weitere Informationen über die für die Unterstützung von Sitzungen mit SSH erforderlichen Konfigurationsschritte finden Sie unter (Optional) Zulassen und Steuern von Berechtigungen für SSH-Verbindungen über Session Manager.
>
> Die `kms:GenerateDataKey`-Berechtigung ermöglicht die Erstellung eines Datenverschlüsselungsschlüssels, der zur Verschlüsselung von Sitzungsdaten verwendet wird. Wenn Sie die Verschlüsselung AWS Key Management Service (AWS KMS) für Ihre Sitzungsdaten verwenden, *key-name* ersetzen Sie sie durch den Amazon-Ressourcennamen (ARN) des KMS-Schlüssels, den Sie verwenden möchten, im Format `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-12345EXAMPLE`. Wenn Sie keine KMS-Schlüsselverschlüsselung für Ihre Sitzungsdaten verwenden möchten, entfernen Sie den folgenden Inhalt aus der Richtlinie.
>
> ```
> {
>             "Effect": "Allow",
>             "Action": [
>                 "kms:GenerateDataKey"
>             ],
>             "Resource": "key-name"
>         }
> ```
>
> Informationen zur Verwendung AWS KMS zur Verschlüsselung von Sitzungsdaten finden Sie unterSo aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole).

Die Genehmigung für SendCommandist für Fälle erforderlich, in denen ein Benutzer versucht, eine Sitzung von der EC2 Amazon-Konsole aus zu starten, die jedoch Session Manager zunächst auf die erforderliche Mindestversion aktualisiert werden SSM Agent muss. Run Commandwird verwendet, um einen Befehl an die Instance zu senden, um den Agenten zu aktualisieren.

Kurzeinführung in Administratorrichtlinien für Session Manager

Verwenden Sie die folgenden Beispiele, um IAM-Administratorrichtlinien für Session Manager zu erstellen.

Diese Richtlinien bieten Administratoren die Möglichkeit, eine Sitzung für verwaltete Knoten zu starten, die mit `Key=Finance,Value=WebServers` markiert sind, sowie die Berechtigung zum Erstellen, Aktualisieren und Löschen von Einstellungen und die Berechtigung, nur ihre eigenen Sitzungen zu beenden. Beispiele für Anpassungen, die Sie möglicherweise für die Richtlinie ausführen sollten, finden Sie unter Zusätzliche IAM-Beispielrichtlinien für Session Manager.

Sie können eine Richtlinie erstellen, die es Administratoren ermöglicht, diese Aufgaben nur von der Session Manager Konsole und AWS CLI nur von der EC2 Amazon-Konsole aus oder von allen drei aus auszuführen.

Ersetzen Sie in den folgenden Beispielrichtlinien jede *`example resource placeholder`* durch Ihre eigenen Informationen.

Lesen Sie die folgenden Abschnitte, um Beispielrichtlinien für die drei Berechtigungsszenarien anzuzeigen.

Session Manager and CLI

Verwenden Sie diese Beispielrichtlinie für Provider-Administratoren, die sitzungsbezogene Aufgaben nur über die Session Manager-Konsole und die AWS CLI ausführen können. Diese Richtlinie bietet nicht alle erforderlichen Berechtigungen, um sitzungsbezogene Aufgaben von der EC2 Amazon-Konsole aus auszuführen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:*:111122223333:instance/*"
            ],
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/Finance": [
                        "WebServers"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeSessions",
                "ssm:GetConnectionStatus",
                "ssm:DescribeInstanceProperties",
                "ec2:DescribeInstances"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:CreateDocument",
                "ssm:UpdateDocument",
                "ssm:GetDocument",
                "ssm:StartSession"
            ],
```

```
            "Resource": "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        }
    ]
}
```

Amazon EC2

Verwenden Sie diese Beispielrichtlinie, um Administratoren die Möglichkeit zu geben, sitzungsbezogene Aufgaben nur von der EC2 Amazon-Konsole aus auszuführen. Diese Richtlinie bietet nicht alle Berechtigungen, die zum Ausführen von sitzungsbezogenen Aufgaben über die Session Manager-Konsole und die AWS CLI-Konsole erforderlich sind.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession",
```

```
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1:111122223333:instance/*"
            ],
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/tag-key": [
                        "tag-value"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
            ]
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetConnectionStatus",
                "ssm:DescribeInstanceInformation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
```

```
            ]
        }
    ]
}
```

Session Manager, CLI, and Amazon EC2

Verwenden Sie diese Beispielrichtlinie, um Administratoren die Möglichkeit zu geben, sitzungsbezogene Aufgaben von der Session Manager Konsole, der und der AWS CLI EC2 Amazon-Konsole aus auszuführen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession",
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1:111122223333:instance/*"
            ],
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/tag-key": [
                        "tag-value"
                    ]
                }
            }
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "ssm:DescribeSessions",
                "ssm:GetConnectionStatus",
                "ssm:DescribeInstanceInformation",
                "ssm:DescribeInstanceProperties",
                "ec2:DescribeInstances"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:CreateDocument",
                "ssm:UpdateDocument",
                "ssm:GetDocument",
                "ssm:StartSession"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        }
    ]
}
```

> **Note**
>
> Die Genehmigung für SendCommandist für Fälle erforderlich, in denen ein Benutzer versucht, eine Sitzung von der EC2 Amazon-Konsole aus zu starten, aber SSM Agent zuerst ein Befehl zum Aktualisieren gesendet werden muss.

Zusätzliche IAM-Beispielrichtlinien für Session Manager

Die folgenden Beispielrichtlinien helfen Ihnen beim Erstellen einer benutzerdefinierten AWS Identity and Access Management (IAM)-Richtlinien für alle Session Manager-Benutzerzugriffsszenarien, die Sie unterstützen müssen.

Themen

- Beispiel 1: Zugriff auf Dokumente in der Konsole gewähren
- Beispiel 2: Beschränken des Zugriffs auf bestimmte verwaltete Knoten
- Beispiel 3: Beschränken des Zugriffs anhand von Tags
- Beispiel 4: Benutzern erlauben, ausschließlich von ihnen gestartete Sitzungen zu beenden
- Beispiel 5: Benutzer erhalten vollständigen (administrativen) Zugriff auf alle Sitzungen

Beispiel 1: Zugriff auf Dokumente in der Konsole gewähren

Sie können Benutzern erlauben, ein benutzerdefiniertes Dokument anzugeben, wenn sie eine Sitzung über die Session-Manager-Konsole starten. Das folgende Beispiel für eine IAM-Richtlinie gewährt die Erlaubnis, auf Dokumente zuzugreifen, deren Namen mit **`SessionDocument-`** den angegebenen AWS-Region und AWS-Konto beginnen.

Um diese Richtlinie zu verwenden, ersetzen Sie jede *`example resource placeholder`* durch Ihre eigenen Informationen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetDocument",
                "ssm:ListDocuments"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/SessionDocument-*"
            ]
        }
```

```
    ]
}
```

> **Note**
>
> Die Session-Manager-Konsole unterstützt nur Sitzungsdokumente mit einem `sessionType` von `Standard_Stream`, die zur Definition von Sitzungseinstellungen verwendet werden. Weitere Informationen finden Sie unter Schema des Sitzungsdokuments.

Beispiel 2: Beschränken des Zugriffs auf bestimmte verwaltete Knoten

Sie können eine IAM-Richtlinie erstellen, die definiert, mit welchen verwalteten Knoten ein Benutzer mithilfe von Session Manager eine Verbindung herstellen darf. Die folgende Richtlinie gewährt einem Benutzer beispielsweise die Berechtigung, seine Sitzungen auf drei bestimmten Knoten zu starten, zu beenden und fortzusetzen. Die Richtlinie schränkt den Benutzer ein, eine Verbindung zu anderen als den angegebenen Knoten herzustellen.

> **Note**
>
> Informationen zu verbundenen Benutzern finden Sie unter Beispiel 4: Benutzern erlauben, ausschließlich von ihnen gestartete Sitzungen zu beenden.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1:111122223333:instance/
i-1234567890EXAMPLE",
                "arn:aws:ec2:us-east-1:111122223333:instance/i-
abcdefghijEXAMPLE",
```

```
                "arn:aws:ec2:us-east-1:111122223333:instance/
i-0e9d8c7b6aEXAMPLE",
                "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
            ]
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       }
    ]
}
```

Beispiel 3: Beschränken des Zugriffs anhand von Tags

Sie können den Zugriff auf verwaltete Knoten anhand bestimmter Tags einschränken. Im folgenden Beispiel darf der Benutzer Sitzungen (`Effect: Allow, Action: ssm:StartSession, ssm:ResumeSession`) auf jedem verwalteten Knoten (`Resource: arn:aws:ec2:region:987654321098:instance/*`) starten und fortsetzen, vorausgesetzt, dass es sich bei dem Knoten um einen Finanzknoten WebServer (`ssm:resourceTag/Finance: WebServer`) handelt. Wenn der Benutzer einen Befehl an einen verwalteten Knoten sendet, der nicht markiert ist oder einen anderen Tag als `Finance: WebServer` hat, enthält das Befehlsergebnis `AccessDenied`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1:111122223333:instance/*"
            ],
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/Finance": [
                        "WebServers"
                    ]
                }
            }
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:TerminateSession",
                "ssm:ResumeSession"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:userid}-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
```

```
                "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
            ]
        }
    ]
}
```

Sie können IAM-Richtlinien erstellen, mit denen ein Benutzer Sitzungen mit verwalteten Knoten starten kann, die mit mehreren Tags markiert sind. Die folgende Richtlinie ermöglicht dem Benutzer das Starten von Sitzungen mit verwalteten Knoten, auf denen beide angegebenen Tags angewendet wurden. Wenn ein Benutzer einen Befehl an einen verwalteten Knoten sendet, der nicht mit beiden Tags markiert ist, enthält das Befehlsergebnis `AccessDenied`.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ssm:StartSession"
         ],
         "Resource":"*",
         "Condition":{
            "StringLike":{
               "ssm:resourceTag/tag-key1":[
                  "tag-value1"
               ],
               "ssm:resourceTag/tag-key2":[
                  "tag-value2"
               ]
            }
         }
      },
      {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       },
      {
```

```
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
            ]
        }
    ]
}
```

Weitere Informationen zum Erstellen von IAM-Richtlinien finden Sie unter Verwaltete Richtlinien und eingebundene Richtlinien im IAM-Benutzerhandbuch. Weitere Informationen zum Taggen von verwalteten Knoten finden Sie unter Tagging your Amazon EC2 Resources im EC2 Amazon-Benutzerhandbuch (Inhalt bezieht sich auf Windows und Linux verwaltete Knoten). Weitere Informationen zur Steigerung des Sicherheitsstatus in Bezug auf nicht autorisierte Befehle auf Root-Ebene auf Ihren verwalteten Knoten finden Sie unter Einschränken des Zugriffs auf Befehle auf Stammebene durch SSM Agent

Beispiel 4: Benutzern erlauben, ausschließlich von ihnen gestartete Sitzungen zu beenden

Session Managerbietet zwei Methoden, um zu steuern, welche Sitzungen ein verbundener Benutzer in Ihrem Netzwerk AWS-Konto beenden darf.

- Verwenden Sie die Variable `{aws:userid}` in einer AWS Identity and Access Management (IAM-) Berechtigungsrichtlinie. Verbundbenutzer können nur von ihnen gestartete Sitzungen beenden. Verwenden Sie für Benutzer ohne Verbundzugriff Methode 1. Verwenden Sie für Verbundbenutzer Methode 2.
- Verwenden Sie Tags, die von AWS Tags in einer IAM-Berechtigungsrichtlinie bereitgestellt werden. Sie nehmen eine Bedingung in die Richtlinie auf, die es Benutzern erlaubt, nur Sitzungen zu beenden, die mit bestimmten Tags versehen sind, die von AWS bereitgestellt wurden. Diese Methode funktioniert für alle Konten, auch für Konten, die Verbundkonten verwenden, um Zugriff IDs zu gewähren. AWS

Methode 1: Gewähren Sie TerminateSession Berechtigungen mithilfe der Variablen **{aws:username}**

Die folgende IAM-Richtlinie ermöglicht es einem Benutzer, alle Sitzungen in Ihrem Konto einzusehen. IDs Benutzer können jedoch nur über von ihnen gestartete Sitzungen mit verwalteten Knoten interagieren. Ein Benutzer, dem die folgende Richtlinie zugewiesen wurde, kann keine Verbindungen mit Sitzungen anderer Benutzer herstellen oder diese beenden. Die Richtlinie verwendet die Variable `{aws:username}`, um dies zu erreichen.

> **Note**
>
> Diese Methode funktioniert nicht für Konten, die Zugriff auf die AWS Nutzung von Federated IDs gewähren.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "ssm:DescribeSessions"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Action": [
                "ssm:TerminateSession"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ssm:*:*:session/${aws:username}-*"
            ]
        }
    ]
}
```

Methode 2: Gewähren Sie TerminateSession Berechtigungen mithilfe von Tags, die bereitgestellt werden von AWS

Sie können steuern, welche Sitzungen ein Benutzer beenden kann, indem Sie eine Bedingung mit bestimmten Tag-Schlüsselvariablen in einer IAM-Richtlinie verwenden. Die Bedingung gibt an, dass der Benutzer nur Sitzungen beenden kann, die mit einer oder beiden dieser spezifischen Tag-Schlüsselvariablen und einem angegebenen Wert gekennzeichnet sind.

Wenn ein Benutzer in Ihrem Umfeld eine Sitzung AWS-Konto startet, Session Manager wendet er zwei Ressourcen-Tags auf die Sitzung an. Das erste Ressourcen-Tag ist `aws:ssmmessages:target-id`, mit dem Sie die ID des Ziels angeben, das der Benutzer beenden darf. Das andere Ressourcen-Tag ist `aws:ssmmessages:session-id`, mit einem Wert im Format *`role-id`*`:`*`caller-specified-role-name`*.

> **Note**
>
> Session Manager unterstützt keine benutzerdefinierten Tags für diese IAM-Zugriffssteuerungsrichtlinie. Sie müssen die unten beschriebenen Resource-Tags verwenden AWS, die von bereitgestellt werden.

`aws:ssmmessages:target-id`

Mit diesem Tag-Schlüssel schließen Sie die ID des verwalteten Knotens als Wert in die Richtlinie ein. Im folgenden Richtlinienblock lässt die Bedingungsanweisung einen Benutzer nur den Knoten i-02573cafcfEXAMPLE beenden.

JSON

```
{
     "Version":"2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                "ssm:TerminateSession"
             ],
             "Resource": "*",
             "Condition": {
                 "StringLike": {
                     "ssm:resourceTag/aws:ssmmessages:target-id": [
```

```
                    "i-02573cafcfEXAMPLE"
                ]
            }
        }
    }
]
}
```

Wenn der Benutzer versucht, eine Sitzung zu beenden, für die ihm diese `TerminateSession`-Berechtigung nicht erteilt wurde, wird eine `AccessDeniedException`-Fehlermeldung angezeigt.

`aws:ssmmessages:session-id`

Dieser Tag-Schlüssel enthält als Wert in der Anforderung zum Starten einer Sitzung eine Variable für die Sitzungs-ID.

Das folgende Beispiel zeigt eine Richtlinie für Fälle, in denen der Aufrufertyp `User` ist. Der Wert, für den Sie für `aws:ssmmessages:session-id` angeben, ist die ID des Benutzers. In diesem Beispiel stellt `AIDIODR4TAW7CSEXAMPLE` die ID eines Benutzers in Ihrem AWS-Konto dar. Um die ID für einen Benutzer in Ihrem abzurufen AWS-Konto, verwenden Sie den IAM-Befehl,`get-user`. Weitere Informationen finden Sie unter get-user im AWS Identity and Access Management Abschnitt des IAM-Benutzerhandbuchs.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
               "ssm:TerminateSession"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "ssm:resourceTag/aws:ssmmessages:session-id": [
                       "AIDIODR4TAW7CSEXAMPLE"
                    ]
                }
            }
        }
```

```
    ]
}
```

Das folgende Beispiel zeigt eine Richtlinie für Fälle, in denen der Aufrufertyp `AssumedRole` ist. Sie können die Variable `{aws:userid}` für den Wert verwenden, den Sie für `aws:ssmmessages:session-id` angeben. Alternativ können Sie eine Rollen-ID für den Wert, den Sie für `aws:ssmmessages:session-id` angeben, fest codieren. Wenn Sie eine Rollen-ID fest codieren, müssen Sie den Wert im Format *`role-id`*`:`*`caller-specified-role-name* angeben. Beispiel, `AIDIODR4TAW7CSEXAMPLE:MyRole`.

> **Important**
>
> Damit System-Tags angewendet werden können, darf die von Ihnen bereitzustellende Rollen-ID nur folgende Zeichen enthalten: Unicode-Buchstaben, 0-9, Leerzeichen, `_`, `.`, `:`, `/`, `=`, `+`, `-`, `@` und `\`.

Verwenden Sie den Befehl, um die Rollen-ID für eine Rolle in Ihrem AWS-Konto abzurufen. `get-caller-identity` Weitere Informationen finden Sie get-caller-identityin der AWS CLI Befehlsreferenz.

JSON

```
{
     "Version":"2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                "ssm:TerminateSession"
             ],
             "Resource": "*",
             "Condition": {
                 "StringLike": {
                     "ssm:resourceTag/aws:ssmmessages:session-id": [
                        "${aws:userid}*"
                     ]
                 }
             }
         }
```

```
    ]
}
```

Wenn ein Benutzer versucht, eine Sitzung zu beenden, für die ihm diese `TerminateSession`-Berechtigung nicht erteilt wurde, wird eine `AccessDeniedException`-Fehlermeldung angezeigt.

`aws:ssmmessages:target-id` und **`aws:ssmmessages:session-id`**

Sie können auch IAM-Richtlinien erstellen, die es einem Benutzer ermöglichen, Sitzungen zu beenden, die mit beiden System-Tags gekennzeichnet sind, wie in diesem Beispiel dargestellt.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ssm:TerminateSession"
         ],
         "Resource":"*",
         "Condition":{
            "StringLike":{
               "ssm:resourceTag/aws:ssmmessages:target-id":[
                  "i-02573cafcfEXAMPLE"
               ],
               "ssm:resourceTag/aws:ssmmessages:session-id":[
                  "${aws:userid}*"
               ]
            }
         }
      }
   ]
}
```

Beispiel 5: Benutzer erhalten vollständigen (administrativen) Zugriff auf alle Sitzungen

Die folgende IAM-Richtlinie ermöglicht Benutzern die vollständige Interaktion mit allen verwalteten Knoten und allen Sitzungen, die von allen Benutzern für alle Knoten erstellt wurden. Diese

Berechtigung sollte nur einem Administrator gewährt werden, der vollständige Kontrolle über die Session Manager-Aktivitäten Ihrer Organisation benötigt.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "ssm:StartSession",
                "ssm:TerminateSession",
                "ssm:ResumeSession",
                "ssm:DescribeSessions",
                "ssm:GetConnectionStatus"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
       }
    ]
}
```

Schritt 4: Konfigurieren von Sitzungspräferenzen

Benutzer, denen in ihrer AWS Identity and Access Management (IAM)-Richtlinie administrative Berechtigungen gewährt wurden, können die Sitzungseinstellungen konfigurieren, einschließlich der folgenden:

- Aktivieren Sie den Run-As-Support für Linux-verwaltete Knoten. Dadurch ist es möglich, Sitzungen unter Verwendung der Anmeldeinformationen für den Benutzer einen bestimmtes Betriebssystems anstelle der Anmeldeinformationen eines vom System generierten `ssm-user`-Konto zu verwenden, das AWS Systems Manager Session Manager auf einem verwalteten Knoten erstellen kann.

- Konfigurieren Sie Session Manager, um mit AWS KMS key-Verschlüsselung eine zusätzliche Schutzschicht für die übermittelten Daten zwischen Client-Maschinen und verwalteten Knoten einzurichten.
- Konfigurieren Sie Session Manager so, dass es die Protokolle des Sitzungsverlaufs erstellt und an einen Amazon Simple Storage Service (Amazon S3)-Bucket oder an eine Amazon CloudWatch Logs-Protokollgruppe sendet. Die gespeicherten Protokolldaten können anschließend verwendet werden, um die Sitzungsverbindungen mit Ihren verwalteten Knoten und die auf diesen während der Sitzungen ausgeführten Befehle zu melden.
- Konfigurieren Sie Sitzungs-Timeouts. Mit dieser Einstellung können Sie festlegen, wann eine Sitzung nach einem Zeitraum der Inaktivität beendet werden soll.
- Konfigurieren Sie Session Manager so, dass es konfigurierbare Shell-Profile verwendet. Mit diesen anpassbaren Profilen können Sie Einstellungen in Sitzungen wie Shell-Einstellungen, Umgebungsvariablen, Arbeitsverzeichnissen und das Ausführen mehrerer Befehle definieren, wenn eine Sitzung gestartet wird.

Weitere Informationen zu den Berechtigungen, die zum Konfigurieren von Session Manager-Einstellungen erforderlich sind, finden Sie unter the section called “Gewähren oder Verweigern von Benutzerberechtigungen zum Aktualisieren von Session Manager-Einstellungen”.

Themen

- Gewähren oder Verweigern von Benutzerberechtigungen zum Aktualisieren von Session Manager-Einstellungen
- Angeben eines Zeitüberschreitungswerts für Leerlaufsitzungen
- Angeben der maximalen Sitzungsdauer
- Konfigurierbare Shell-Profile zulassen
- Run-As-Support für Linux- und macOS-verwaltete Knoten einschalten
- So aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole)
- Erstellen eines Dokuments mit Session Manager-Einstellungen (Befehlszeile)
- Aktualisieren von Session Manager-Einstellungen (Befehlszeile)

Weitere Informationen zur Verwendung der Systems Manager-Konsole zum Konfigurieren von Optionen für die Protokollierung von Sitzungsdaten finden Sie in den folgenden Themen:

- Protokollieren von Sitzungsdaten mithilfe von Amazon S3 (Konsole)

- Streaming-Sitzungsdaten mit Amazon CloudWatch Logs (Konsole)
- Protokollierung von Sitzungsdaten mit Amazon CloudWatch Logs (Konsole)

Gewähren oder Verweigern von Benutzerberechtigungen zum Aktualisieren von Session Manager-Einstellungen

Kontoeinstellungen werden jeweils AWS-Region als AWS Systems Manager (SSM-) Dokumente gespeichert. Bevor Benutzer die Kontoeinstellungen für Sitzungen in Ihrem Konto aktualisieren können, müssen ihnen die notwendigen Berechtigungen für den Zugriff auf die Art des SSM-Dokuments gewährt werden, in denen diese Einstellungen gespeichert werden. Diese Berechtigungen werden durch eine AWS Identity and Access Management (IAM-) Richtlinie gewährt.

Administratorrichtlinie, die das Erstellen und Aktualisieren von Richtlinien zulässt

Ein Administrator kann die folgende Richtlinie zum jederzeitigen Erstellen und Aktualisieren von Einstellungen besitzen. Die folgende Richtlinie gewährt die Berechtigung für Zugriff und Aktualisierung des Dokuments `SSM-SessionManagerRunShell` im Konto 123456789012 in der Region us-east-2.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "ssm:CreateDocument",
                "ssm:GetDocument",
                "ssm:UpdateDocument",
                "ssm:DeleteDocument"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
            ]
        }
    ]
}
```

Benutzerrichtlinie, die das Aktualisieren von Einstellungen verhindert

Mittels der folgenden Richtlinie verhindern Sie das Aktualisieren oder Überschreiben von Session Manager-Einstellungen durch Endbenutzer in Ihrem Konto.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "ssm:CreateDocument",
                "ssm:GetDocument",
                "ssm:UpdateDocument",
                "ssm:DeleteDocument"
            ],
            "Effect": "Deny",
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/SSM-
SessionManagerRunShell"
            ]
        }
    ]
}
```

Angeben eines Zeitüberschreitungswerts für Leerlaufsitzungen

Mit Session Manager, einem Tool in AWS Systems Manager, können Sie festlegen, wie lange Benutzer inaktiv sein können, bevor das System eine Sitzung beendet. Standardmäßig wird eine Sitzung nach 20 Minuten Inaktivität beendet. Sie können diese Einstellung ändern und eine Zeitüberschreitung zwischen 1 und 60 Minuten Inaktivität festlegen. Einige professionelle Agenturen für Computersicherheit empfehlen, Timeouts für inaktive Sitzungen auf maximal 15 Minuten festzulegen.

So lassen Sie Zeitüberschreitungen für Leerlaufsitzungen zu (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Präferenzen und anschließend Bearbeiten aus.
4. Geben Sie im Feld minutes unter Zeitüberschreitung bei Leerlaufsitzung an, wie lange ein Benutzer inaktiv sein kann, bevor eine Sitzung beendet wird.
5. Wählen Sie Speichern.

Angeben der maximalen Sitzungsdauer

Session Manager, ein Tool in AWS Systems Manager, ermöglicht es Ihnen, die maximale Dauer einer Sitzung anzugeben, bevor sie endet. Standardmäßig haben Sitzungen keine maximale Dauer. Der Wert, den Sie für die maximale Sitzungsdauer angeben, muss zwischen 1 und 1 440 Minuten liegen.

So geben Sie die maximale Sitzungsdauer an (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Präferenzen und anschließend Bearbeiten aus.
4. Aktivieren Sie das Kontrollkästchen neben Enable maximum session duration (Aktivieren der maximalen Sitzungsdauer).
5. Geben Sie die maximale Sitzungsdauer in dem Feld minutes (Minuten) unter Maximum session duration (Maximale Sitzungsdauer) an.
6. Wählen Sie Speichern.

Konfigurierbare Shell-Profile zulassen

Standardmäßig starten Sitzungen auf EC2-Instances für Linux, die Bourne-Shell (sh) zu verwenden. Sie könnten jedoch eine andere Shell wie bash vorziehen. Indem Sie konfigurierbare Shell-Profile zulassen, können Sie Einstellungen in Sitzungen wie Shell-Einstellungen, Umgebungsvariablen, Arbeitsverzeichnissen und das Ausführen mehrerer Befehle anpassen, wenn eine Sitzung gestartet wird.

Important

Systems Manager überprüft die Befehle oder Skripts in Ihrem Shell-Profil nicht, bevor sie ausgeführt werden, um zu sehen, welche Änderungen sie an einer Instance vornehmen würden. Um die Fähigkeit eines Benutzers, Befehle oder Skripte zu ändern, die in seinem Shell-Profil eingegeben wurden, einzuschränken, wird Folgendes empfohlen:

- Erstellen Sie ein angepasstes Sitzungsdokument für Ihre AWS Identity and Access Management (IAM)-Benutzer und -Rollen. Ändern Sie dann die IAM-Richtlinie für diese Benutzer und Rollen so, dass die `StartSession` API-Operation nur das Sitzungstyp-Dokument verwenden kann, das Sie für sie erstellt haben. Weitere Informationen finden Sie unter Erstellen eines Dokuments mit Session Manager-Einstellungen (Befehlszeile) und Kurzeinführung in Endbenutzerrichtlinien für Session Manager.
- Ändern Sie die IAM-Richtlinie für Ihre IAM-Benutzer und -Rollen, um den Zugriff auf die `UpdateDocument` API-Operation für die von Ihnen erstellte Sitzungstyp-Dokumentressource zu verweigern. Auf diese Weise können Benutzer und Rollen das von Ihnen erstellte Dokument für ihre Sitzungseinstellungen verwenden, ohne dass sie die Einstellungen ändern können.

So aktivieren Sie konfigurierbare Shell-Profile

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Präferenzen und anschließend Bearbeiten aus.
4. Geben Sie die Umgebungsvariablen, Shell-Einstellungen oder Befehle, die beim Start der Sitzung ausgeführt werden sollen, in den Feldern der entsprechenden Betriebssysteme an.
5. Wählen Sie Speichern.

Im Folgenden sehen Sie einige Beispielbefehle, die Ihrem Shell-Profil hinzugefügt werden können.

Wechseln Sie zur bash-Shell und wechseln Sie in das Verzeichnis /usr auf Linux-Instances.

```
exec /bin/bash
cd /usr
```

Geben Sie einen Zeitstempel und eine Begrüßungsnachricht zu Beginn einer Sitzung aus.

Linux & macOS

```
timestamp=$(date '+%Y-%m-%dT%H:%M:%SZ')
user=$(whoami)
echo $timestamp && echo "Welcome $user"'!'
echo "You have logged in to a production instance. Note that all session activity is
 being logged."
```

Windows

```
$timestamp = (Get-Date).ToString("yyyy-MM-ddTH:mm:ssZ")
$splitName = (whoami).Split("\")
$user = $splitName[1]
Write-Host $timestamp
Write-Host "Welcome $user!"
Write-Host "You have logged in to a production instance. Note that all session
 activity is being logged."
```

Zeigen Sie die dynamische Systemaktivität zu Beginn einer Sitzung an.

Linux & macOS

```
top
```

Windows

```
while ($true) { Get-Process | Sort-Object -Descending CPU | Select-Object -First 30;
 `
Start-Sleep -Seconds 2; cls
Write-Host "Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id ProcessName";
Write-Host "-------  ------    -----      ----- -----   ------     -- -----------"}
```

Run-As-Support für Linux- und macOS-verwaltete Knoten einschalten

Standardmäßig authentifiziert Session Manager Verbindungen mit den Anmeldeinformationen des vom System generierten `ssm-user`-Kontos, das auf einem verwalteten Knoten erstellt wird. (Auf Linux- und macOS-Maschinen wird das Konto zu `/etc/sudoers/` hinzugefügt.) Wenn Sie

möchten, können Sie Sitzungen stattdessen mit den Anmeldeinformationen eines Betriebssystem-Benutzerkontos (OS) oder eines Domainbenutzers für Instances authentifizieren, die einem Active Directory beigetreten sind. In diesem Fall überprüft Session Manager vor dem Starten der Sitzung, ob das von Ihnen angegebene Betriebssystemkonto auf dem Knoten oder in der Domain vorhanden ist. Wenn Sie versuchen, eine Sitzung mit einem Betriebssystemkonto zu starten, das auf dem Knoten oder in der Domain nicht vorhanden ist, schlägt die Verbindung fehl.

> **Note**
>
> Session Manager unterstützt nicht die Verwendung des `root`-Benutzerkontos eines Betriebssystems zur Authentifizierung von Verbindungen. Für Sitzungen, die mit einem Betriebssystem-Benutzerkonto authentifiziert werden, gelten die Betriebssystem- und Verzeichnisrichtlinien des Knotens, wie Anmeldeeinschränkungen oder Nutzungseinschränkungen für Systemressourcen, möglicherweise nicht.

Funktionsweise

Wenn Sie die Run As-Unterstützung für Sitzungen aktivieren, überprüft das System für Zugriffsberechtigungen wie folgt:

1. Wurde die IAM-Entität (Benutzer oder Rolle) des Benutzers, der die Sitzung startet, mit `SSMSessionRunAs = os user account name` gekennzeichnet?

   Falls ja, ist der Betriebssystem-Benutzername auf dem verwalteten Knoten vorhanden? Wenn dies der Fall ist, wird die Sitzung gestartet. Wenn dies nicht der Fall ist, wird das Starten der Sitzung verboten.

   Wenn die IAM-Entität nicht mit `SSMSessionRunAs = os user account name` gekennzeichnet wurde, fahren Sie mit Schritt 2 fort.
2. Wenn die IAM-Entität nicht mit `SSMSessionRunAs = os user account name` gekennzeichnet wurde, wurde in den Session Manager-Einstellungen des AWS-Kontos ein Betriebssystem-Benutzername angegeben?

   Falls ja, ist der Betriebssystem-Benutzername auf dem verwalteten Knoten vorhanden? Wenn dies der Fall ist, wird die Sitzung gestartet. Wenn dies nicht der Fall ist, wird das Starten der Sitzung verboten.

> **Note**
>
> Wenn Sie die Unterstützung „Ausführen als“ aktivieren, wird Session Manager daran gehindert, Sitzungen mit dem `ssm-user`-Konto auf einem verwalteten Knoten zu starten. Dies bedeutet, dass, wenn Session Manager die Verbindung nicht mithilfe des angegebenen Betriebssystem-Benutzerkontos herstellen kann, es nicht auf die Standardmethode zurückgreift.
> Wenn Sie „Ausführen als“ aktivieren, ohne ein Betriebssystemkonto anzugeben oder eine IAM-Entität zu markieren, und Sie in den Session Manager-Einstellungen kein Betriebssystemkonto angegeben haben, schlagen Sitzungsverbindungsversuche fehl.

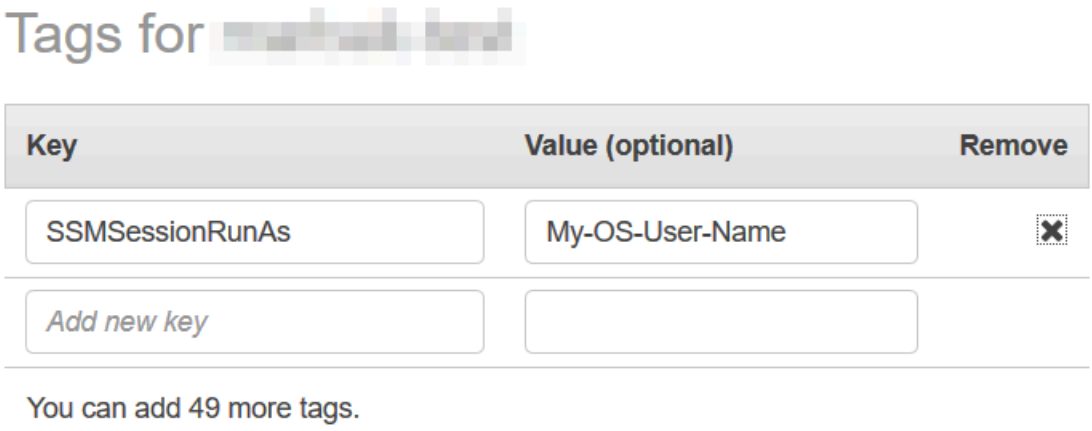
So aktivieren Sie den Run-As-Support für Linux- und macOS-verwaltete Knoten

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Präferenzen und anschließend Bearbeiten aus.
4. Aktivieren Sie das Kontrollkästchen neben Run As-Unterstützung für Linux-Instances aktivieren.
5. Führen Sie eine der folgenden Aktionen aus:

   - Option 1: Geben Sie im Feld Benutzername des Betriebssystems den Namen des Benutzerkontos des Betriebssystems auf dem verwalteten Zielknoten ein, den Sie zum Starten von Sitzungen verwenden möchten. Mit dieser Option werden alle Sitzungen von demselben Betriebssystem-Benutzer für alle Benutzer in Ihrem AWS-Konto ausgeführt, die eine Verbindung mithilfe von Session Manager herstellen.
   - Option 2 (Empfohlen): Wählen Sie den Link IAM-Konsole öffnen aus. Wählen Sie im Navigationsbereich eine der Optionen Users (Benutzer) oder Roles (Rollen). Wählen Sie die Entität (Benutzer oder Rolle) aus, der Sie Tags hinzufügen möchten, und wählen Sie dann die Registerkarte Tags. Geben Sie `SSMSessionRunAs` als Schlüsselname ein. Geben Sie den Namen eines Betriebssystem-Benutzerkontos als den Schlüsselwert ein. Wählen Sie Änderungen speichern aus.

     Mit dieser Option können Sie bei Bedarf eindeutige Betriebssystembenutzer für verschiedene IAM-Entitäten angeben. Weitere Informationen zum Markieren von IAM-Ressourcen (Benutzer oder Rollen) finden Sie unter Markieren von IAM-Ressourcen im IAM-Benutzerhandbuch.

Im Folgenden wird ein Beispiel gezeigt.



6. Wählen Sie Speichern.

So aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole)

Verwenden Sie AWS Key Management Service (AWS KMS) zum Erstellen und Verwalten von Verschlüsselungschlüsseln. Mit AWS KMS können Sie die Verwendung der Verschlüsselung in einer breiten Palette an AWS-Services und in Ihren Anwendungen steuern. Sie können festlegen, dass die Sitzungsdaten, die zwischen Ihren verwalteten Knoten und den lokalen Maschinen der Benutzer in Ihrem AWS-Konto übertragen werden, mithilfe der KMS-Schlüsselverschlüsselung verschlüsselt werden. (Dies wird zusätzlich zur TLS-1.2/1.3-Verschlüsselung angewendet, die AWS bereits standardmäßig zur Verfügung stellt.) Um Session Manager-Sitzungsdaten zu verschlüsseln, erstellen Sie einen symmetrischen KMS-Schlüssel mit AWS KMS.

AWS KMS-Verschlüsselung ist für `Standard_Stream`, `NonInteractiveCommands` und `InteractiveCommands`-Sitzungstypen verfügbar. Damit Sie die Option zum Verschlüsseln von Sitzungsdaten mit einem in AWS KMS erstellten Schlüssel verwenden können, muss Version 2.3.539.0 oder höher des AWS Systems Manager SSM Agent auf dem verwalteten Knoten installiert sein.

> **Note**
>
> Sie müssen die AWS KMS-Verschlüsselung erlauben, um Passwörter auf Ihren verwalteten Knoten über die AWS Systems Manager-Konsole zurückzusetzen. Weitere Informationen finden Sie unter Zurücksetzen eines Passworts auf einem verwalteten Knoten.

Sie können einen Schlüssel verwenden, den Sie in Ihrem AWS-Konto erstellt haben. Sie können jedoch auch einen Schlüssel verwenden, der in einem anderen AWS-Konto erstellt wurde. Der Ersteller des Schlüssel in einem anderen AWS-Konto muss Ihnen die erforderlichen Berechtigungen zur Nutzung des Schlüssels erteilen.

Nachdem Sie die KMS-Schlüsselverschlüsselung für Ihre Sitzungsdaten aktiviert haben, müssen sowohl die Benutzer, die Sitzungen starten, als auch die verwalteten Knoten, mit denen sie verbunden sind, über die Berechtigung zur Verwendung des Schlüssels verfügen. Sie erteilen die Berechtigung zur Verwendung des KMS-Schlüssels mit Session Manager anhand von AWS Identity and Access Management (IAM)-Richtlinien. Weitere Informationen finden Sie unter den folgenden Themen:

- Hinzufügen von AWS KMS-Berechtigungen für Benutzer in Ihrem Konto: Muster-IAM-Richtlinien für Session Manager.
- Hinzufügen von AWS KMS-Berechtigungen für verwalteten Knoten in Ihrem Konto: Schritt 2: Überprüfen oder Hinzufügen von Instance-Berechtigungen für Session Manager.

Weitere Informationen zum Erstellen und Verwalten von KMS-Schlüsseln finden Sie im AWS Key Management Service-Entwicklerhandbuch.

Weitere Informationen zur Verwendung der AWS CLI zur Aktivierung der KMS-Schlüsselverschlüsselung von Sitzungsdaten in Ihrem Konto finden Sie unter Erstellen eines Dokuments mit Session Manager-Einstellungen (Befehlszeile) oder Aktualisieren von Session Manager-Einstellungen (Befehlszeile).

> **Note**
>
> Es entstehen Kosten für die Verwendung von KMS-Schlüsseln. Weitere Informationen finden Sie unter AWS Key Management Service-Preise.

So aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Präferenzen und anschließend Bearbeiten aus.

4. Aktivieren Sie das Kontrollkästchen neben Enable KMS encryption (Aktivieren der KMS-Verschlüsselung).
5. Führen Sie eine der folgenden Aktionen aus:

   - Klicken Sie auf die Schaltfläche neben Select a KMS key in my current account (Einen KMS-Schlüssel in meinem aktuellen Konto auswählen) und wählen Sie anschließend einen Schlüssel aus der Liste aus.

     –oder–

     Wählen Sie die Schaltfläche neben Enter a KMS key alias or KMS key ARN (Einen KMS-Schlüssel-Alias oder KMS-Schlüssel-ARN eingeben) aus. Geben Sie manuell einen KMS-Schlüssel-Alias für einen Schlüssel ein, der in Ihrem aktuellen Konto erstellt wurde. Für einen Schlüssel in einem anderen Konto geben Sie den Amazon-Ressourcennamen (ARN) des Schlüssels ein. Im Folgenden sind einige Beispiele aufgeführt:

     - Schlüssel-Alias: `alias/my-kms-key-alias`
     - Schüssel-ARN: `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-12345EXAMPLE`

     –oder–

     Wählen Sie Create new key (Neuen Schlüssel erstellen), um einen neuen KMS-Schlüssel in Ihrem Konto zu erstellen. Nachdem Sie den neuen Schlüssel erstellt haben, kehren Sie zur Registerkarte Preferences (Einstellungen) zurück und wählen Sie den Schlüssel zum Verschlüsseln der Sitzungsdaten in Ihrem Konto aus.

   Weitere Informationen zum Freigeben von Schlüsseln finden Sie unter Externen AWS-Konten den Zugriff auf einen Schlüssel erlauben im AWS Key Management Service-Entwicklerhandbuch.
6. Wählen Sie Speichern.

Erstellen eines Dokuments mit Session Manager-Einstellungen (Befehlszeile)

Gehen Sie wie folgt vor, um SSM-Dokumente zu erstellen, die Ihre Einstellungen für AWS Systems Manager Session Manager Sitzungen definieren. Sie können das Dokument verwenden, um Sitzungsoptionen wie Datenverschlüsselung, Sitzungsdauer und Protokollierung zu konfigurieren. Sie können zum Beispiel angeben, ob Sitzungsprotokolldaten in einem Bucket von Amazon

Simple Storage Service (Amazon S3) oder in einer Protokollgruppe von Amazon CloudWatch Logs gespeichert werden sollen. Sie können Dokumente erstellen, die allgemeine Einstellungen für alle Sitzungen für AWS-Konto und AWS-Region festlegen oder die Einstellungen für einzelne Sitzungen definieren.

> **Note**
>
> Sie können die allgemeinen Sitzungseinstellungen auch über die Session-Manager-Konsole konfigurieren.

Dokumente, die zum Einstellen von Session-Manager-Einstellungen verwendet werden, müssen einen `sessionType` von `Standard_Stream` haben. Weitere Informationen zu Sitzungs-Dokumenten finden Sie unter the section called “Schema des Sitzungsdokuments”.

Weitere Informationen zur Verwendung der Befehlszeile zum Aktualisieren vorhandener Session Manager-Einstellungen finden Sie unter Aktualisieren von Session Manager-Einstellungen (Befehlszeile).

Ein Beispiel für die Erstellung von Sitzungseinstellungen mit CloudFormation finden Sie unter Erstellen eines Systems Manager-Dokuments für Session Manager-Einstellungen im AWS CloudFormation-Benutzerhandbuch.

> **Note**
>
> Diese Vorgehensweise beschreibt, wie Sie Dokumente für die Festlegung von Session Manager-Einstellungen auf der AWS-Konto-Ebene erstellen. Um Dokumente zu erstellen, die für die Festlegung von Einstellungen auf Sitzungsebene verwendet werden, geben Sie einen anderen Wert als `SSM-SessionManagerRunShell` für die dateibezogenen Befehlseingaben an.
>
> Wenn Sie Ihr Dokument verwenden möchten, um Einstellungen für Sitzungen festzulegen, die mit der AWS Command Line Interface (AWS CLI) gestartet wurden, geben Sie den Dokumentnamen als `--document-name`-Parameterwert an. Um Einstellungen für Sitzungen vorzunehmen, die von der Session-Manager-Konsole aus gestartet werden, können Sie den Namen Ihres Dokuments eingeben oder aus einer Liste auswählen.

So erstellen Sie Session Manager-Einstellungen (Befehlszeile)

1. Erstellen Sie eine JSON-Datei auf Ihrem lokalen Computer und geben Sie Ihr beispielsweise einen Namen wie `SessionManagerRunShell.json`. Fügen Sie der Datei anschließend den folgenden Inhalt ein.

```
{
    "schemaVersion": "1.0",
    "description": "Document to hold regional settings for Session Manager",
    "sessionType": "Standard_Stream",
    "inputs": {
        "s3BucketName": "",
        "s3KeyPrefix": "",
        "s3EncryptionEnabled": true,
        "cloudWatchLogGroupName": "",
        "cloudWatchEncryptionEnabled": true,
        "cloudWatchStreamingEnabled": false,
        "kmsKeyId": "",
        "runAsEnabled": false,
        "runAsDefaultUser": "",
        "idleSessionTimeout": "",
        "maxSessionDuration": "",
        "shellProfile": {
            "windows": "date",
            "linux": "pwd;ls"
        }
    }
}
```

Sie können Werte auch mithilfe von Parametern an Ihre Sitzungseinstellungen übergeben, anstatt die Werte fest zu kodieren, wie im folgenden Beispiel gezeigt.

```
{
   "schemaVersion":"1.0",
   "description":"Session Document Parameter Example JSON Template",
   "sessionType":"Standard_Stream",
   "parameters":{
      "s3BucketName":{
         "type":"String",
         "default":""
      },
      "s3KeyPrefix":{
```

```
         "type":"String",
         "default":""
      },
      "s3EncryptionEnabled":{
         "type":"Boolean",
         "default":"false"
      },
      "cloudWatchLogGroupName":{
         "type":"String",
         "default":""
      },
      "cloudWatchEncryptionEnabled":{
         "type":"Boolean",
         "default":"false"
      }
   },
   "inputs":{
      "s3BucketName":"{{s3BucketName}}",
      "s3KeyPrefix":"{{s3KeyPrefix}}",
      "s3EncryptionEnabled":"{{s3EncryptionEnabled}}",
      "cloudWatchLogGroupName":"{{cloudWatchLogGroupName}}",
      "cloudWatchEncryptionEnabled":"{{cloudWatchEncryptionEnabled}}",
      "kmsKeyId":""
   }
}
```

2. Legen Sie fest, wohin Sie die Sitzungsdaten senden möchten. Sie können einen S3-Bucketnamen (mit einem optionalen Präfix) oder eine CloudWatch Logs-Protokollgruppe angeben. Wenn Sie die Daten zwischen dem lokalen Client und den verwalteten Knoten weiter verschlüsseln möchten, geben Sie den KMS-Schlüssel ein, der für die Verschlüsselung verwendet werden soll. Im Folgenden wird ein Beispiel gezeigt.

```
{
  "schemaVersion": "1.0",
  "description": "Document to hold regional settings for Session Manager",
  "sessionType": "Standard_Stream",
  "inputs": {
    "s3BucketName": "amzn-s3-demo-bucket",
    "s3KeyPrefix": "MyS3Prefix",
    "s3EncryptionEnabled": true,
    "cloudWatchLogGroupName": "MyLogGroupName",
    "cloudWatchEncryptionEnabled": true,
    "cloudWatchStreamingEnabled": false,
```

```
    "kmsKeyId": "MyKMSKeyID",
    "runAsEnabled": true,
    "runAsDefaultUser": "MyDefaultRunAsUser",
    "idleSessionTimeout": "20",
    "maxSessionDuration": "60",
    "shellProfile": {
        "windows": "MyCommands",
        "linux": "MyCommands"
    }
  }
}
```

> **Note**
>
> Wenn Sie die Protokolldaten der Sitzung nicht verschlüsseln möchten, ändern Sie für `s3EncryptionEnabled` `true` in `false`.
> Wenn Sie die Protokolle weder an einen Amazon S3-Bucket noch eine CloudWatch Logs-Protokollgruppe senden bzw. die aktiven Sitzungsdaten nicht verschlüsseln möchten oder die Run As-Unterstützung für die Sitzungen an Ihrem Konto nicht aktivieren möchten, können Sie die Zeilen für diese Optionen löschen. Überprüfen Sie, dass die letzte Zeile im Abschnitt `inputs` nicht mit einem Komma endet.
> Wenn Sie eine KMS-Schlüssel-ID zum Verschlüsseln Ihrer Sitzungsdaten hinzufügen, müssen sowohl die Benutzer, die die Sitzungen starten, als auch die verwalteten Knoten, mit denen sie sich verbinden, über die Berechtigung zur Verwendung des Schlüssels verfügen. Sie erteilen die Berechtigung zur Verwendung des KMS-Schlüssels mit Session Manager anhand von IAM-Richtlinien. Weitere Informationen finden Sie unter den folgenden Themen:
>
> - Hinzufügen von AWS KMS-Berechtigungen für Benutzer in Ihrem Konto: Muster-IAM-Richtlinien für Session Manager.
> - Hinzufügen von AWS KMS-Berechtigungen für verwalteten Knoten in Ihrem Konto: Schritt 2: Überprüfen oder Hinzufügen von Instance-Berechtigungen für Session Manager

3. Speichern Sie die Datei.
4. Führen Sie in dem Verzeichnis, in dem Sie die JSON-Datei erstellt haben, den folgenden Befehl aus.

Linux & macOS

```
aws ssm create-document \
    --name SSM-SessionManagerRunShell \
    --content "file://SessionManagerRunShell.json" \
    --document-type "Session" \
    --document-format JSON
```

Windows

```
aws ssm create-document ^
    --name SSM-SessionManagerRunShell ^
    --content "file://SessionManagerRunShell.json" ^
    --document-type "Session" ^
    --document-format JSON
```

PowerShell

```
New-SSMDocument `
    -Name "SSM-SessionManagerRunShell" `
    -Content (Get-Content -Raw SessionManagerRunShell.json) `
    -DocumentType "Session" `
    -DocumentFormat JSON
```

Bei erfolgreicher Ausführung gibt der Befehl eine Ausgabe zurück, die in etwa wie folgt aussieht:

```
{
    "DocumentDescription": {
        "Status": "Creating",
        "Hash": "ce4fd0a2ab9b0fae759004ba603174c3ec2231f21a81db8690a33eb66EXAMPLE",
        "Name": "SSM-SessionManagerRunShell",
        "Tags": [],
        "DocumentType": "Session",
        "PlatformTypes": [
            "Windows",
            "Linux"
        ],
        "DocumentVersion": "1",
        "HashType": "Sha256",
        "CreatedDate": 1547750660.918,
```

```
        "Owner": "111122223333",
        "SchemaVersion": "1.0",
        "DefaultVersion": "1",
        "DocumentFormat": "JSON",
        "LatestVersion": "1"
    }
}
```

Aktualisieren von Session Manager-Einstellungen (Befehlszeile)

Im folgenden Verfahren wird beschrieben, wie Sie mit Ihrem bevorzugten Befehlszeilen-Tool Änderungen an den AWS Systems Manager Session Manager-Einstellungen für Ihr AWS-Konto in der ausgewählten AWS-Region vornehmen. Verwenden Sie Session Manager-Einstellungen, um die Optionen für die Protokollierung von Sitzungsdaten in einem Amazon Simple Storage Service (Amazon S3)-Bucket oder einer Amazon CloudWatch Logs-Protokollgruppe anzugeben. Mithilfe der Session Manager-Einstellungen können Sie zudem Ihre Sitzungsdaten verschlüsseln.

So aktualisieren Sie Session Manager-Einstellungen (Befehlszeile)

1. Erstellen Sie eine JSON-Datei auf Ihrem lokalen Computer und geben Sie Ihr beispielsweise einen Namen wie `SessionManagerRunShell.json`. Fügen Sie der Datei anschließend den folgenden Inhalt ein.

```
{
    "schemaVersion": "1.0",
    "description": "Document to hold regional settings for Session Manager",
    "sessionType": "Standard_Stream",
    "inputs": {
        "s3BucketName": "",
        "s3KeyPrefix": "",
        "s3EncryptionEnabled": true,
        "cloudWatchLogGroupName": "",
        "cloudWatchEncryptionEnabled": true,
        "cloudWatchStreamingEnabled": false,
        "kmsKeyId": "",
        "runAsEnabled": true,
        "runAsDefaultUser": "",
        "idleSessionTimeout": "",
        "maxSessionDuration": "",
        "shellProfile": {
            "windows": "date",
```

```
            "linux": "pwd;ls"
        }
    }
}
```

2. Legen Sie fest, wohin Sie die Sitzungsdaten senden möchten. Sie können einen S3-Bucketnamen (mit einem optionalen Präfix) oder eine CloudWatch Logs-Protokollgruppe angeben. Wenn Sie die Daten zwischen dem lokalen Client und den verwalteten Knoten weiter verschlüsseln möchten, stellen Sie den AWS KMS key bereit, der für die Verschlüsselung verwendet werden soll. Im Folgenden wird ein Beispiel gezeigt.

```
{
  "schemaVersion": "1.0",
  "description": "Document to hold regional settings for Session Manager",
  "sessionType": "Standard_Stream",
  "inputs": {
    "s3BucketName": "amzn-s3-demo-bucket",
    "s3KeyPrefix": "MyS3Prefix",
    "s3EncryptionEnabled": true,
    "cloudWatchLogGroupName": "MyLogGroupName",
    "cloudWatchEncryptionEnabled": true,
    "cloudWatchStreamingEnabled": false,
    "kmsKeyId": "MyKMSKeyID",
    "runAsEnabled": true,
    "runAsDefaultUser": "MyDefaultRunAsUser",
    "idleSessionTimeout": "20",
    "maxSessionDuration": "60",
    "shellProfile": {
        "windows": "MyCommands",
        "linux": "MyCommands"
    }
  }
}
```

> **Note**
>
> Wenn Sie die Protokolldaten der Sitzung nicht verschlüsseln möchten, ändern Sie für `s3EncryptionEnabled` `true` in `false`.
> Wenn Sie die Protokolle weder an einen Amazon S3-Bucket noch eine CloudWatch Logs-Protokollgruppe senden bzw. die aktiven Sitzungsdaten nicht verschlüsseln möchten oder die Run As-Unterstützung für die Sitzungen an Ihrem Konto nicht

aktivieren möchten, können Sie die Zeilen für diese Optionen löschen. Überprüfen Sie, dass die letzte Zeile im Abschnitt `inputs` nicht mit einem Komma endet.
Wenn Sie eine KMS-Schlüssel-ID zum Verschlüsseln Ihrer Sitzungsdaten hinzufügen, müssen sowohl die Benutzer, die die Sitzungen starten, als auch die verwalteten Knoten, mit denen sie sich verbinden, über die Berechtigung zur Verwendung des Schlüssels verfügen. Sie erteilen die Berechtigung zur Verwendung des KMS-Schlüssels mit Session Manager anhand von AWS Identity and Access Management (IAM)-Richtlinien. Weitere Informationen finden Sie unter den folgenden Themen:

- Hinzufügen von AWS KMS-Berechtigungen für Benutzer in Ihrem Konto: Muster-IAM-Richtlinien für Session Manager.
- Hinzufügen von AWS KMS-Berechtigungen für verwalteten Knoten in Ihrem Konto: Schritt 2: Überprüfen oder Hinzufügen von Instance-Berechtigungen für Session Manager.

3. Speichern Sie die Datei.
4. Führen Sie in dem Verzeichnis, in dem Sie die JSON-Datei erstellt haben, den folgenden Befehl aus.

   Linux & macOS

   ```
   aws ssm update-document \
       --name "SSM-SessionManagerRunShell" \
       --content "file://SessionManagerRunShell.json" \
       --document-version "\$LATEST"
   ```

   Windows

   ```
   aws ssm update-document ^
       --name "SSM-SessionManagerRunShell" ^
       --content "file://SessionManagerRunShell.json" ^
       --document-version "$LATEST"
   ```

   PowerShell

   ```
   Update-SSMDocument `
       -Name "SSM-SessionManagerRunShell" `
       -Content (Get-Content -Raw SessionManagerRunShell.json) `
   ```

```
    -DocumentVersion '$LATEST'
```

Bei erfolgreicher Ausführung gibt der Befehl eine Ausgabe zurück, die in etwa wie folgt aussieht:

```
{
    "DocumentDescription": {
        "Status": "Updating",
        "Hash": "ce4fd0a2ab9b0fae759004ba603174c3ec2231f21a81db8690a33eb66EXAMPLE",
        "Name": "SSM-SessionManagerRunShell",
        "Tags": [],
        "DocumentType": "Session",
        "PlatformTypes": [
            "Windows",
            "Linux"
        ],
        "DocumentVersion": "2",
        "HashType": "Sha256",
        "CreatedDate": 1537206341.565,
        "Owner": "111122223333",
        "SchemaVersion": "1.0",
        "DefaultVersion": "1",
        "DocumentFormat": "JSON",
        "LatestVersion": "2"
    }
}
```

Schritt 5: (Optional) Beschränken des Zugriffs auf Befehle in einer Sitzung

Sie können die Befehle einschränken, die ein Benutzer in einer AWS Systems Manager Session Manager Sitzung ausführen kann, indem Sie ein Dokument mit benutzerdefiniertem `Session` Typ AWS Systems Manager (SSM) verwenden. In dem Dokument definieren Sie den Befehl, der ausgeführt wird, wenn der Benutzer eine Sitzung startet, und die Parameter, die der Benutzer dem Befehl übergeben kann. Die `Session` des `schemaVersion`-Dokuments muss 1.0 und der `sessionType` des Dokuments muss `InteractiveCommands` lauten. Anschließend können Sie AWS Identity and Access Management (IAM)-Richtlinien erstellen, die es den Benutzern ermöglichen, nur auf die von Ihnen definierten `Session`-Dokumente zuzugreifen. Weitere Informationen zur Verwendung von IAM-Richtlinien zum Beschränken des Zugriffs auf Befehle in einer Sitzung finden Sie unter IAM-Richtlinienbeispiele für interaktive Befehle.

Dokumente mit dem Zeichen `sessionType` von `InteractiveCommands` werden nur für Sitzungen unterstützt, die mit AWS Command Line Interface (AWS CLI) gestartet wurden. Der Benutzer gibt den Namen des benutzerdefinierten Dokuments als `--document-name`-Parameterwert an und gibt alle Befehlsparameterwerte über die Option `--parameters` an. Weitere Informationen zur Ausführung interaktiver Befehle finden Sie unter Starten einer Sitzung (interaktive und nicht interaktive Befehle).

Gehen Sie wie folgt vor, um ein SSM-Dokument vom benutzerdefinierten Typ `Session` zu erstellen, das den Befehl definiert, den ein Benutzer ausführen darf.

Beschränken des Zugriffs auf Befehle in einer Sitzung (Konsole)

So beschränken Sie die Befehle, die ein Benutzer in einer Session Manager-Sitzung ausführen kann (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie Create command or session (Befehl oder Sitzung erstellen) aus.
4. Geben Sie unter Name einen aussagekräftigen Namen für das Dokument ein.
5. Wählen Sie für Document type (Dokumenttyp) die Option Session document (Sitzungsdokument) aus.
6. Geben Sie mithilfe von JSON oder YAML Ihren Dokumentinhalt ein, der den Befehl definiert, den ein Benutzer in einer Session Manager-Sitzung ausführen kann, wie im folgenden Beispiel gezeigt.

   YAML

```
---
schemaVersion: '1.0'
description: Document to view a log file on a Linux instance
sessionType: InteractiveCommands
parameters:
  logpath:
    type: String
    description: The log file path to read.
    default: "/var/log/amazon/ssm/amazon-ssm-agent.log"
    allowedPattern: "^[a-zA-Z0-9-_/]+(.log)$"
properties:
  linux:
```

```
    commands: "tail -f {{ logpath }}"
    runAsElevated: true
```

JSON

```
{
    "schemaVersion": "1.0",
    "description": "Document to view a log file on a Linux instance",
    "sessionType": "InteractiveCommands",
    "parameters": {
        "logpath": {
            "type": "String",
            "description": "The log file path to read.",
            "default": "/var/log/amazon/ssm/amazon-ssm-agent.log",
            "allowedPattern": "^[a-zA-Z0-9-_/]+(.log)$"
        }
    },
    "properties": {
        "linux": {
            "commands": "tail -f {{ logpath }}",
            "runAsElevated": true
        }
    }
}
```

7. Wählen Sie Create document (Dokument erstellen) aus.

Beschränken des Zugriffs auf Befehle in einer Sitzung (Befehlszeile)

Bevor Sie beginnen

Falls Sie es noch nicht getan haben, installieren und konfigurieren Sie die AWS Command Line Interface (AWS CLI) oder die AWS -Tools für PowerShell. Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.

So beschränken Sie die Befehle, die ein Benutzer in einer Session Manager-Sitzung ausführen kann (Befehlszeile)

1. Erstellen Sie eine JSON- oder YAML-Datei für Ihren Dokumentinhalt, der den Befehl definiert, den ein Benutzer in einer Session Manager-Sitzung ausführen kann, wie im folgenden Beispiel gezeigt.

YAML

```
---
schemaVersion: '1.0'
description: Document to view a log file on a Linux instance
sessionType: InteractiveCommands
parameters:
  logpath:
    type: String
    description: The log file path to read.
    default: "/var/log/amazon/ssm/amazon-ssm-agent.log"
    allowedPattern: "^[a-zA-Z0-9-_/]+(.log)$"
properties:
  linux:
    commands: "tail -f {{ logpath }}"
    runAsElevated: true
```

JSON

```
{
    "schemaVersion": "1.0",
    "description": "Document to view a log file on a Linux instance",
    "sessionType": "InteractiveCommands",
    "parameters": {
        "logpath": {
            "type": "String",
            "description": "The log file path to read.",
            "default": "/var/log/amazon/ssm/amazon-ssm-agent.log",
            "allowedPattern": "^[a-zA-Z0-9-_/]+(.log)$"
        }
    },
    "properties": {
        "linux": {
            "commands": "tail -f {{ logpath }}",
            "runAsElevated": true
        }
    }
}
```

2. Führen Sie die folgenden Befehle aus, um ein SSM-Dokument unter Verwendung Ihres Inhalts zu erstellen, der den Befehl definiert, den ein Benutzer in einer Session Manager-Sitzung ausführen kann.

Linux & macOS

```
aws ssm create-document \
    --content file://path/to/file/documentContent.json \
    --name "exampleAllowedSessionDocument" \
    --document-type "Session"
```

Windows

```
aws ssm create-document ^
    --content file://C:\path\to\file\documentContent.json ^
    --name "exampleAllowedSessionDocument" ^
    --document-type "Session"
```

PowerShell

```
$json = Get-Content -Path "C:\path\to\file\documentContent.json" | Out-String
New-SSMDocument `
    -Content $json `
    -Name "exampleAllowedSessionDocument" `
    -DocumentType "Session"
```

Interaktive Befehlsparameter und AWS CLI

Sie können interaktive Befehlsparameter bereitstellen, wenn Sie die AWS CLI verwenden. Je nach Betriebssystem (OS) Ihres Client-Computers, mit dem Sie eine Verbindung zu verwalteten Knoten herstellen, kann die Syntax AWS CLI, die Sie für Befehle angeben, die Sonder- oder Escape-Zeichen enthalten, unterschiedlich sein. Die folgenden Beispiele zeigen einige der verschiedenen Möglichkeiten, wie Sie Befehlsparameter angeben können, wenn Sie die verwenden AWS CLI, und wie mit Sonder- oder Escape-Zeichen umgegangen wird.

Auf Parameter, die in gespeichert sind, Parameter Store kann in den AWS CLI Befehlsparametern verwiesen werden, wie im folgenden Beispiel gezeigt.

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name MyInteractiveCommandDocument \
```

```
    --parameters '{"command":["{{ssm:mycommand}}"]}'
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name MyInteractiveCommandDocument ^
    --parameters '{"command":["{{ssm:mycommand}}"]}'
```

Das folgende Beispiel zeigt, wie Sie mit der AWS CLI eine Kurzschriftsyntax verwenden, um Parameter zu übergeben.

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name MyInteractiveCommandDocument \
    --parameters command="ifconfig"
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name MyInteractiveCommandDocument ^
    --parameters command="ipconfig"
```

Sie können auch optionale Parameter in JSON angeben, wie im folgenden Beispiel dargestellt.

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name MyInteractiveCommandDocument \
    --parameters '{"command":["ifconfig"]}'
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
```

```
    --document-name MyInteractiveCommandDocument ^
    --parameters '{"command":["ipconfig"]}'
```

Parameter können auch in einer JSON-Datei gespeichert und für die bereitgestellt werden, AWS CLI wie im folgenden Beispiel gezeigt. Weitere Informationen zur Verwendung von AWS CLI -Parametern aus einer Datei finden Sie unter Laden von AWS CLI -Parametern aus einer Datei im AWS Command Line Interface -Benutzerhandbuch.

```
{
    "command": [
        "my command"
    ]
}
```

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name MyInteractiveCommandDocument \
    --parameters file://complete/path/to/file/parameters.json
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name MyInteractiveCommandDocument ^
    --parameters file://complete/path/to/file/parameters.json
```

Sie können auch ein AWS CLI Skelett aus einer JSON-Eingabedatei generieren, wie im folgenden Beispiel gezeigt. Weitere Informationen zum Generieren von AWS CLI Skeletten aus JSON-Eingabedateien finden Sie im Benutzerhandbuch unter Generieren von AWS CLI Skeletten und Eingabeparametern aus einer JSON- oder YAML-Eingabedatei.AWS Command Line Interface

```
{
    "Target": "instance-id",
    "DocumentName": "MyInteractiveCommandDocument",
    "Parameters": {
        "command": [
            "my command"
```

```
        ]
    }
}
```

Linux & macOS

```
aws ssm start-session \
    --cli-input-json file://complete/path/to/file/parameters.json
```

Windows

```
aws ssm start-session ^
    --cli-input-json file://complete/path/to/file/parameters.json
```

Um Zeichen in Anführungszeichen zu maskieren, müssen Sie den Escapezeichen zusätzliche umgekehrte Schrägstriche hinzufügen, wie im folgenden Beispiel gezeigt.

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name MyInteractiveCommandDocument \
    --parameters '{"command":["printf \"abc\\\\tdef\""]}'
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name MyInteractiveCommandDocument ^
    --parameters '{"command":["printf \"abc\\\\tdef\""]}'
```

Informationen zum Verwenden von Anführungszeichen bei Befehlsparametern in AWS CLI finden Sie unter Verwenden von Anführungszeichen mit Zeichenfolgen in der AWS CLI im AWS Command Line Interface -Benutzerhandbuch.

IAM-Richtlinienbeispiele für interaktive Befehle

Sie können IAM-Richtlinien erstellen, mit denen Benutzer nur auf die von Ihnen definierten `Session`-Dokumente zugreifen können. Dadurch werden die Befehle, die ein Benutzer in einer Session

Manager-Sitzung ausführen kann, nur auf die Befehle beschränkt, die in Ihren benutzerdefinierten SSM-Dokumenten vom Typ `Session` definiert sind.

Einem Benutzer erlauben, einen interaktiven Befehl auf einem einzelnen verwalteten Knoten auszuführen

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ssm:StartSession",
         "Resource":[
            "arn:aws:ec2:us-east-1:444455556666:instance/i-02573cafcfEXAMPLE",
            "arn:aws:ssm:us-east-1:444455556666:document/allowed-session-
document"
         ]
      },
      {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
      }
   ]
}
```

Einem Benutzer erlauben, einen interaktiven Befehl auf allen verwalteten Knoten auszuführen

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ssm:StartSession",
         "Resource":[
            "arn:aws:ec2:us-east-1:444455556666:instance/*",
            "arn:aws:ssm:us-east-1:444455556666:document/allowed-session-
document"
         ]
```

```
      },
      {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
      }
   ]
}
```

Einem Benutzer erlauben, mehrere interaktive Befehle auf allen verwalteten Knoten auszuführen

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ssm:StartSession",
         "Resource":[
            "arn:aws:ec2:us-east-1:444455556666:instance/*",
            "arn:aws:ssm:us-east-1:444455556666:document/allowed-session-
document",
            "arn:aws:ssm:us-east-1:444455556666:document/allowed-session-
document-2"
         ]
      },
      {
         "Effect": "Allow",
         "Action": ["ssmmessages:OpenDataChannel"],
         "Resource": ["arn:aws:ssm:*:*:session/${aws:userid}-*"]
      }
   ]
}
```

Schritt 6: (Optional) Verwenden von AWS PrivateLink zum Einrichten eines VPC-Endpunkts für Session Manager

Sie können den Sicherheitsstatus Ihrer verwalteten Knoten weiter verbessern, indem Sie AWS Systems Manager zum Verwenden eines Virtual Private Cloud (VPC)-Schnittstellenendpunkts konfigurieren. Schnittstellenendpunkte werden mittels AWS PrivateLink realisiert, einer Technologie,

mit der Sie unter Verwendung privater IP-Adressen privat auf Amazon Elastic Compute Cloud (Amazon EC2)- und Systems Manager-APIs zugreifen können.

AWS PrivateLink schränkt den gesamten Netzwerkverkehr zwischen Ihren verwalteten Knoten, Systems Manager und Amazon EC2 auf das Amazon-Netzwerk ein. (Verwaltete Knoten haben keinen Zugriff auf das Internet.) Zudem benötigen Sie kein Internet-Gateway, kein NAT-Gerät und kein Virtual Private Gateway.

Informationen zum Erstellen eines VPC-Endpunkts finden Sie unter Die Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten verbessern für Systems Manager.

Die Alternative zur Verwendung eines VPC-Endpunkts ist das Erlauben von ausgehendem Internetzugriff auf Ihre verwalteten Knoten. In diesem Fall müssen die verwalteten Knoten auch ausgehenden HTTPS-Datenverkehr (Port 443) zu den folgenden Endpunkten erlauben:

- `ec2messages.region.amazonaws.com`
- `ssm.region.amazonaws.com`
- `ssmmessages.region.amazonaws.com`

Systems Manager verwendet `ssmmessages.region.amazonaws.com`, den letzten dieser Endpunkte, über den Sie Anrufe von SSM Agent auf den Session Manager-Service in der Cloud tätigen können.

Um optionale Funktionen wie AWS Key Management Service (AWS KMS- Verschlüsselung, Streaming von Protokollen an Amazon CloudWatch Logs (CloudWatch Logs) und das Senden von Protokollen an Amazon Simple Storage Service (Amazon S3) zu verwenden, müssen Sie ausgehenden HTTPS-Datenverkehr (Port 443) an die folgenden Endpunkte erlauben:

- `kms.region.amazonaws.com`
- `logs.region.amazonaws.com`
- `s3.region.amazonaws.com`

Weitere Informationen zu erforderlichen Endpunkten für Systems Manager finden Sie unter Referenz: ec2messages, ssmmessages und andere API-Operationen.

Schritt 7: (Optional) Deaktivieren oder Aktivieren der Administratorberechtigungen für das SSM-Benutzerkonto

Ab Version 2.3.50.0 von AWS Systems Manager SSM Agent, erstellt der Agent ein lokales Benutzerkonto namens `ssm-user` und fügt es `/etc/sudoers` (Linux und macOS) bzw. der Administratorengruppe (Windows) hinzu. Auf Agenten-Versionen vor 2.3.612.0 wird das Konto beim ersten Start bzw. Neustart des SSM Agent nach der Installation erstellt. Auf Version 2.3.612.0 und höher wird das `ssm-user`-Konto beim ersten Start einer Sitzung auf einem Knoten erstellt. Dieser `ssm-user` ist der Standardbenutzer des Betriebssystems (OS), wenn eine AWS Systems Manager Session Manager-Sitzung gestartet wird. SSM Agent Version 2.3.612.0 wurde am 8. Mai 2019 veröffentlicht.

Wenn Sie verhindern möchten, dass Session Manager-Benutzer administrative Befehle auf einem Knoten ausführen, können Sie ihre `ssm-user`-Kontoberechtigungen aktualisieren. Sie können diese Berechtigungen wiederherstellen, nachdem sie entfernt wurden.

Themen

- Verwalten von sudo-Berechtigungen für ssm-user-Konten in Linux und macOS
- Verwalten von Administratorberechtigungen für das Konto ssm-user in Windows Server

Verwalten von sudo-Berechtigungen für ssm-user-Konten in Linux und macOS

Mit den folgenden Verfahren können Sie die sudo-Berechtigungen von ssm-Benutzerkonten auf Linux- und macOS-verwalteten Knoten aktivieren oder deaktivieren.

Verwenden von Run Command zum Ändern von sudo-Berechtigungen für ssm-user (Konsole)

- Verwenden Sie die Prozedur in Ausführen von Befehlen über die Konsole mit den folgenden Werten:
  - Wählen Sie unter Command document (Befehlsdokument) die Option `AWS-RunShellScript` aus.
  - Um den sudo-Zugriff zu entfernen, fügen Sie im Bereich Command parameters (Befehlsparameter) Folgendes in das Feld Commands (Befehle) ein.

```
cd /etc/sudoers.d
echo "#User rules for ssm-user" > ssm-agent-users
```

–oder–

Um den sudo-Zugriff wiederherzustellen, fügen Sie im Bereich Command parameters (Befehlsparameter) Folgendes in das Feld Commands (Befehle) ein.

```
cd /etc/sudoers.d
echo "ssm-user ALL=(ALL) NOPASSWD:ALL" > ssm-agent-users
```

Verwenden der Befehlszeile zum Ändern von sudo-Berechtigungen für ssm-user (AWS CLI)

1. Stellen Sie eine Verbindung zum verwalteten Knoten her und führen Sie den folgenden Befehl aus.

   ```
   sudo -s
   ```

2. Ändern Sie den Arbeitsordner mit dem folgenden Befehl.

   ```
   cd /etc/sudoers.d
   ```

3. Öffnen Sie die Datei mit dem Namen `ssm-agent-users`, um sie zu bearbeiten.
4. Um den sudo-Zugriff zu entfernen, löschen Sie die folgende Zeile.

   ```
   ssm-user ALL=(ALL) NOPASSWD:ALL
   ```

   –oder–

   Um den sudo-Zugriff wiederherzustellen, fügen Sie die folgende Zeile hinzu.

   ```
   ssm-user ALL=(ALL) NOPASSWD:ALL
   ```

5. Speichern Sie die Datei.

Verwalten von Administratorberechtigungen für das Konto ssm-user in Windows Server

Mit den folgenden Verfahren können Sie die Administratorberechtigungen von ssm-user-Konten auf von Windows Server verwalteten Knoten aktivieren oder deaktivieren.

Verwenden von Run Command zum Ändern von Administratorberechtigungen (Konsole)

- Verwenden Sie die Prozedur in Ausführen von Befehlen über die Konsole mit den folgenden Werten:

  Wählen Sie unter Command document (Befehlsdokument) die Option `AWS-RunPowerShellScript` aus.

  Um den administrativen Zugriff zu entfernen, fügen Sie im Bereich Command parameters (Befehlsparameter) Folgendes in das Feld Commands (Befehle) ein.

  ```
  net localgroup "Administrators" "ssm-user" /delete
  ```

  –oder–

  Um den administrativen Zugriff wiederherzustellen, fügen Sie im Bereich Command parameters (Befehlsparameter) Folgendes in das Feld Commands (Befehle) ein.

  ```
  net localgroup "Administrators" "ssm-user" /add
  ```

Verwenden des PowerShell- oder Eingabeaufforderungs-Fensters zum Ändern von Administratorberechtigungen

1. Stellen Sie eine Verbindung mit dem verwalteten Knoten her und öffnen Sie das PowerShell- oder Eingabeaufforderungsfenster.
2. Um den administrativen Zugriff zu entfernen, führen Sie den folgenden Befehl aus.

   ```
   net localgroup "Administrators" "ssm-user" /delete
   ```

   –oder–

   Um den administrativen Zugriff wiederherzustellen, führen Sie den folgenden Befehl aus.

   ```
   net localgroup "Administrators" "ssm-user" /add
   ```

Verwenden Sie die Windows-Konsole, um die Berechtigungen für Administratoren zu ändern

1. Stellen Sie eine Verbindung mit dem verwalteten Knoten her und öffnen Sie das PowerShell- oder Eingabeaufforderungsfenster.
2. Führen Sie in der Befehlszeile `lusrmgr.msc` aus, um die Konsole Local Users and Groups (Lokale Benutzer und Gruppen) zu öffnen.
3. Öffnen Sie das Verzeichnis Benutzer und dann ssm-user.
4. Führen Sie auf der Registerkarte Member Of (Mitglied von) einen der folgenden Schritte aus:

   - Um den administrativen Zugriff zu entfernen, wählen Sie Administrators (Administratoren) und dann Remove (Entfernen) aus.

     –oder–

     Um den administrativen Zugriff wiederherzustellen, geben Sie **`Administrators`** in das Textfeld ein und klicken dann auf Add (Hinzufügen).
5. Wählen Sie OK aus.

Schritt 8: (Optional) Berechtigungen für SSH-Verbindungen über Session Manager zulassen und steuern

Sie können Benutzern in Ihrem Konto erlauben AWS-Konto , mithilfe von AWS Command Line Interface (AWS CLI) Secure Shell (SSH) -Verbindungen zu verwalteten Knoten herzustellen. AWS Systems Manager Session Manager Benutzer, die eine Verbindung über SSH herstellen, können auch mit dem Secure Copy Protocol (SCP) Dateien zwischen ihren lokalen Maschinen und verwalteten Knoten kopieren. Sie können diese Funktionalität verwenden, um eine Verbindung zu verwalteten Knoten herzustellen, ohne eingehende Ports öffnen oder Bastion-Hosts pflegen zu müssen.

Wenn Sie SSH-Verbindungen überSession Manager, AWS CLI und SSM Agent stellen Sie sichere WebSocket Verbindungen über TLS zu Session Manager Endpunkten her. Die SSH-Sitzung läuft innerhalb dieses verschlüsselten Tunnels und bietet so eine zusätzliche Sicherheitsebene, ohne dass eingehende Ports auf Ihren verwalteten Knoten geöffnet werden müssen.

Nachdem Sie SSH-Verbindungen zugelassen haben, können Sie AWS Identity and Access Management (IAM-) Richtlinien verwenden, um Benutzern, Gruppen oder Rollen das Herstellen von SSH-Verbindungen explizit zu gestatten oder zu verweigern. Session Manager

> **Note**
>
> Protokollieren ist für Session Manager-Sitzungen, die eine Verbindung über Port-Weiterleitung oder SSH herstellen, nicht verfügbar. Das liegt daran, dass SSH alle Sitzungsdaten innerhalb der sicheren TLS-Verbindung zwischen den Endpunkten AWS CLI und Session Manager Endpunkten verschlüsselt und Session Manager nur als Tunnel für SSH-Verbindungen dient.

Themen

- Zulassen von SSH-Verbindungen für Session Manager
- Steuern von Benutzerberechtigungen für SSH-Verbindungen über Session Manager

Zulassen von SSH-Verbindungen für Session Manager

Gehen Sie wie folgt vor, um über Session Manager SSH-Verbindungen für einen verwalteten Knoten zuzulassen.

Um SSH-Verbindungen für Session Manager zuzulassen

1. Gehen Sie auf dem verwalteten Knoten, zu dem Sie SSH-Verbindungen erlauben möchten, wie folgt vor:

   - Stellen Sie sicher, dass SSH auf dem verwalteten Knoten ausgeführt wird. (Sie können eingehende Ports für den Knoten schließen.)
   - Stellen Sie sicher, dass SSM Agent Version 2.3.672.0 oder höher auf dem verwalteten Knoten installiert ist.

     Weitere Informationen zum Installieren oder Aktualisieren von SSM Agent auf einem verwalteten Knoten finden Sie in den folgenden Themen:

     - Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Windows Server.
     - Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux
     - Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für macOS
     - So installieren Sie den SSM Agent in Hybrid-Windows-Knoten
     - So installieren Sie den SSM Agent in Hybrid-Linux-Knoten

> **Note**
>
> Für die Verwendung Session Manager mit lokalen Servern, Edge-Geräten und virtuellen Maschinen (VMs), die Sie als verwaltete Knoten aktiviert haben, müssen Sie die Stufe Advanced-Instances verwenden. Weitere Informationen über erweiterte Instances finden Sie unter Konfigurieren von Instance-Kontingenten.

2. Gehen Sie auf der lokalen Maschine, mit der Sie mit SSH eine Verbindung zu einem verwalteten Knoten herstellen möchten, wie folgt vor:

   - Stellen Sie sicher, dass Version 1.1.23.0 oder höher des Session Manager-Plugin installiert ist.

     Weitere Informationen zur Installation des Session Manager-Plugins finden Sie unter Installieren des Session Manager-Plugin für die AWS CLI.
   - Aktualisieren Sie die SSH-Konfigurationsdatei so, dass ein Proxy-Befehl ausgeführt wird, der eine Session Manager-Sitzung startet und alle Daten über die Verbindung überträgt.

     Linux und macOS

     > **Tip**
     >
     > Die SSH-Konfigurationsdatei befindet sich in der Regel unter `~/.ssh/config`.

     Fügen Sie der Konfigurationsdatei auf dem lokalen Computer den folgenden Code hinzu.

     ```
     # SSH over Session Manager
     Host i-* mi-*
         ProxyCommand sh -c "aws ssm start-session --target %h --document-name AWS-StartSSHSession --parameters 'portNumber=%p'"
         User ec2-user
     ```

     Windows

> **Tip**
>
> Die SSH-Konfigurationsdatei befindet sich in der Regel unter `C:\Users\<username>\.ssh\config`.

Fügen Sie der Konfigurationsdatei auf dem lokalen Computer den folgenden Code hinzu.

```
# SSH over Session Manager
Host i-* mi-*
    ProxyCommand C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe "aws
 ssm start-session --target %h --document-name AWS-StartSSHSession --parameters
 portNumber=%p"
```

- Erstellen ein Privacy-Enhanced-Mail-Zertifikat (eine PEM-Datei) oder mindestens einen öffentlichen Schlüssel, bzw. überprüfen Sie, ob sie darüber verfügen, die beim Herstellen von Verbindungen zu verwalteten Knoten verwendet werden sollen. Dies muss ein Schlüssel sein, der dem verwalteten Knoten bereits zugeordnet ist. Die Berechtigungen für Ihre private Schlüsseldatei müssen so festgelegt sein, dass nur Sie diese lesen können. Mit dem folgenden Befehl können Sie die Berechtigungen für Ihre private Schlüsseldatei so festlegen, dass nur Sie diese lesen können.

  ```
  chmod 400 <my-key-pair>.pem
  ```

  Zum Beispiel für eine Amazon Elastic Compute Cloud (Amazon EC2) -Instance die Schlüsselpaardatei, die Sie bei der Erstellung der Instance erstellt oder ausgewählt haben. (Sie geben den Pfad zum Zertifikat oder Schlüssel als Teil des Befehls zum Starten einer Sitzung an. Informationen zum Starten einer Sitzung mithilfe von SSH finden Sie unter Starten einer Sitzung (SSH).)

Steuern von Benutzerberechtigungen für SSH-Verbindungen über Session Manager

Nachdem Sie SSH-Verbindungen über Session Manager auf einem verwalteten Knoten aktiviert haben, können Sie IAM-Richtlinien verwenden, um Benutzern, Gruppen oder Rollen zu erlauben oder zu verweigern, SSH-Verbindungen über Session Manager herzustellen.

So verwenden Sie eine IAM-Richtlinie, um SSH-Verbindungen über Session Manager zu erlauben

- Wählen Sie eine der folgenden Optionen aus:
  - Option 1: Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.

    Wählen Sie im Navigationsbereich Policies (Richtlinien) aus und aktualisieren Sie dann die Berechtigungsrichtlinie für den Benutzer oder die Rolle, dem/der Sie die Berechtigung erteilen möchten, SSH-Verbindungen über Session Manager zu starten.

    Fügen Sie beispielsweise das folgende Element der Schnellstart-Richtlinie hinzu, die Sie in Kurzeinführung in Endbenutzerrichtlinien für Session Manager erstellt haben. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

    JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ssm:StartSession",
            "Resource": [
                "arn:aws:ec2:us-east-1:111122223333:instance/instance-id",
                "arn:aws:ssm:*:*:document/AWS-StartSSHSession"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ssmmessages:OpenDataChannel",
            "Resource": "arn:aws:ssm:*:*:session/${aws:userid}-*"
        }
    ]
}
```

  - Option 2: Hängen Sie mithilfe der, der oder der AWS-Managementkonsole AWS API eine Inline-Richtlinie an AWS CLI eine Benutzerrichtlinie an.

    Verwenden Sie die Methode Ihrer Wahl und fügen Sie die Richtlinienerklärung in Option 1 der Richtlinie für einen AWS Benutzer, eine Gruppe oder eine Rolle bei.

Informationen finden Sie im Abschnitt Hinzufügen und Entfernen von IAM-Identitätsberechtigungen im IAM-Benutzerhandbuch.

So verwenden Sie eine IAM-Richtlinie, um SSH-Verbindungen über Session Manager zu verweigern

- Wählen Sie eine der folgenden Optionen aus:

  - Option 1: Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/. Wählen Sie im Navigationsbereich Policies (Richtlinien) aus und aktualisieren Sie dann die Berechtigungsrichtlinie für den Benutzer oder die Rolle, die am Starten von Session Manager-Sitzungen gehindert werden soll.

    Fügen Sie beispielsweise das folgende Element der Schnellstart-Richtlinie hinzu, die Sie in Kurzeinführung in Endbenutzerrichtlinien für Session Manager erstellt haben.

    JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ssm:StartSession",
            "Resource": "arn:aws:ssm:*:*:document/AWS-StartSSHSession"
        },
        {
            "Effect": "Allow",
            "Action": "ssmmessages:OpenDataChannel",
            "Resource": "arn:aws:ssm:*:*:session/${aws:userid}-*"
        }
    ]
}
```

  - Option 2: Hängen Sie mithilfe der, der oder der AWS-Managementkonsole AWS API eine Inline-Richtlinie an AWS CLI eine Benutzerrichtlinie an.

    Verwenden Sie die Methode Ihrer Wahl und fügen Sie die Richtlinienerklärung in Option 1 der Richtlinie für einen AWS Benutzer, eine Gruppe oder eine Rolle bei.

    Informationen finden Sie im Abschnitt Hinzufügen und Entfernen von IAM-Identitätsberechtigungen im IAM-Benutzerhandbuch.

Arbeiten mit Session Manager

Sie können die AWS Systems Manager-Konsole, die Amazon Elastic Compute Cloud (Amazon EC2)-Konsole oder die AWS Command Line Interface (AWS CLI) verwenden, um Sitzungen zu starten, die eine Verbindung zu den verwalteten Knoten herstellen, auf die Ihr Systemadministrator Ihnen mit AWS Identity and Access Management (IAM)-Richtlinien Zugriff gewährt hat. Abhängig von Ihren Berechtigungen können Sie auch Informationen zu Sitzungen anzeigen, noch nicht abgelaufene inaktive Sitzungen fortsetzen und Sitzungen beenden. Nachdem eine Sitzung eingerichtet wurde, ist sie nicht von der Dauer der IAM-Rollensitzung betroffen. Informationen zur Begrenzung der Sitzungsdauer mit Session Manager, finden Sie unter Angeben eines Zeitüberschreitungswerts für Leerlaufsitzungen und Angeben der maximalen Sitzungsdauer.

Weitere Informationen zu Sitzungen finden Sie unter Was ist eine Sitzung?.

Themen

- Installieren des Session Manager-Plugin für die AWS CLI
- Starten einer Sitzung
- Eine Sitzung beenden
- Anzeigen des Sitzungsverlaufs

Installieren des Session Manager-Plugin für die AWS CLI

Um Session Manager-Sitzungen mit Ihren verwalteten Knoten mithilfe von AWS Command Line Interface (AWS CLI) zu initiieren, müssen Sie das Session Manager-Plugin auf Ihrer lokalen Maschine installieren. Sie können das Plugin in unterstützten Versionen von Microsoft Windows Server, macOS, Linux und Ubuntu Server installieren.

> **Note**
>
> Um das Session Manager-Plugin zu verwenden, müssen Sie AWS CLI-Version 1.16.12 oder höher auf Ihrer lokalen Maschine installiert haben. Weitere Informationen finden Sie unter Installieren oder Aktualisierung auf die neueste Version von AWS Command Line Interface.

Themen

- Aktuelle Version und Versionsverlauf des Session Manager-Plugins

- Installieren Sie das Session Manager-Plugin in Windows
- Installieren Sie das Session Manager-Plugin in macOS
- Installieren des Session Manager-Plugins unter Linux
- Verifizieren der Session Manager-Plugin-Installation
- Session Manager-Plugin in GitHub
- (Optional) Aktivieren Sie die Session Manager-Plug-In-Protokollierung

Aktuelle Version und Versionsverlauf des Session Manager-Plugins

Auf Ihrem lokalen Computer muss eine unterstützte Version des Session Manager-Plugins ausgeführt werden. Die aktuelle unterstützte Mindestversion ist 1.1.17.0. Wenn Sie eine ältere Version ausführen, werden Ihre Session Manager-Operationen möglicherweise nicht erfolgreich abgeschlossen.

Um zu prüfen, ob Sie die neueste Version ausführen, führen Sie den folgenden Befehl in der AWS CLI aus.

> **Note**
>
> Der Befehl gibt nur dann Ergebnisse zurück, wenn sich das Plugin im Standardinstallationsverzeichnis für Ihren Betriebssystemtypen befindet. Sie können die Version auch in der Datei `VERSION` überprüfen. Diese Datei befindet sich in dem Verzeichnis, in dem Sie das Plugin installiert haben.

```
session-manager-plugin --version
```

In der folgenden Tabelle finden Sie alle Versionen des Session Manager-Plugins sowie die in den einzelnen Versionen enthaltenen Features und Erweiterungen.

> **Important**
>
> Wir empfehlen Ihnen, immer die neueste Version zu verwenden. Die neueste Version enthält Verbesserungen, die die Benutzererfahrung mit dem Plugin verbessern.

Version	Datum der Veröffentlichung	Details
1,2,764,0	19. November 2025	Verbesserung: Unterstützung für OpenDataChannel Signatura nfragen hinzugefügt. Bugfix: Checkstyle-Probleme behoben, um die neuere Go-Version zu unterstützen.
1.2.707.0	6. Februar 2025	Verbesserung: Die Go-Version wurde im Dockerfile auf 1.23 aktualisiert. Der Schritt zur Versionskonfiguration in der README-Datei wurde aktualisiert.
1,2,694,0	20. November 2024	Fehlerbehebung: Die Änderung, durch die Anmeldein formationen zu Anfragen hinzugefügt wurden, wurde rückgängig gemacht. OpenDataChannel
1.2.688.0	6. November 2024	Diese Version ist am 20.11.2024 veraltet. Verbesserungen: • Zugangsdaten zu Anfragen hinzugefügt. OpenDataChannel • Die von `testify` und `objx` abhängigen Pakete wurden aktualisiert.
1.2.677.0	10. Oktober 2024	Verbesserung: Unterstützung für die Weitergabe der Plugin-Version mit Anfragen hinzugefügt. OpenDataChannel
1.2.650.0	02. Juli 2024	Verbesserung: Auf 1.54.10 aktualisiert aws-sdk-go. Bugfix: Kommentare für Gofmt Check wurden neu formatiert.
1.2.633.0	30. Mai 2024	Verbesserung: Das Dockerfile wurde aktualisiert, sodass es ein Amazon Elastic Container Registry (Amazon ECR)-Image verwendet.
1,2,553,0	10. Januar 2024	Verbesserung: Aktualisierte aws-sdk-go und abhängige Golang-Pakete.

Version	Datum der Veröffentlichung	Details
1.2.536.0	4. Dezember 2023	Verbesserung: Unterstützung für die Übergabe einer StartSessionAPI-Antwort als Umgebungsvariable an hinzugefügt. session-manager-plugin
1.2.497.0	1. August 2023	Verbesserung: Go SDK wurde auf v1.44.302 aktualisiert.
1,2,463,0	15. März 2023	Verbesserung: Mac with Apple silicon-Unterstützung für Apple Mac (M1) im macOS-Bundle-Installer und im signierten Installer hinzugefügt.
1,2,398,0	14. Oktober 2022	Verbesserung: Unterstützung für Golang-Version 1.17. Aktualisieren Sie den session-manager-plugin Standard-Runner für macOS, um Python3 zu verwenden. Aktualisieren Sie den Importpfad von SSMCLI auf. session-manager-plugin
1.2.339.0	16. Juni 2022	Fehlerbehebung: Behebt die Zeitbeschränkung der Leerlaufs itzung für Portsitzungen.
1.2.331,0	27. Mai 2022	Fehlerbehebung: Behebt Portsitzungen, die vorzeitig geschlossen werden, wenn der lokale Server vor der Zeitbeschränkung keine Verbindung herstellt.
1.2.323,0	19. Mai 2022	Fehlerbehebung: Deaktivieren Sie Smux Keep Alive, um das Timeout-Feature im Leerlauf zu verwenden.
1.2.312,0	31. März 2022	Verbesserung: Unterstützt mehr Payload-Typen für Ausgabenachrichten.
1.2.295,0	12. Januar 2022	Fehlerbehebung: Aufgehängte Sitzungen durch das erneute Senden von Stream-Daten des Clients, wenn der Agent inaktiv wird, und falsche Protokolle für die Nachrichten `start_publication` und `pause_publication` .
1.2.279,0	27. Oktober 2021	Verbesserung: ZIP-Paketierung für Windows-Plattform.

Version	Datum der Veröffentlichung	Details
1.2.245,0	19. August 2021	Verbesserung: Aktualisieren von `aws-sdk-go` auf die neueste Version (v1.40.17), um AWS IAM Identity Center zu unterstützen.
1,2,234,0	26. Juli 2021	Fehlerbehebung: Abrupt abgebrochenes Szenario im interakti ven Sitzungstyp behandeln.
1.2.205,0	10. Juni 2021	Verbesserung: Unterstützung für signiertes macOS-Ins tallationsprogramm.
1,2,54,0	29. Januar 2021	Verbesserung: Unterstützung für das Ausführen von Sitzungen im NonInteractiveCommands Ausführungsmodus hinzugefügt.
1.2.30.0	24. November 2020	Verbesserung: (Nur Port-Weiterleitungssitzungen) Verbesser te Gesamtleistung.
1.2.7.0	15. Oktober 2020	Verbesserung: (Nur Port-Weiterleitungssitzungen) Verringerte Latenz und verbesserte Gesamtleistung.
1.1.61.0	17. April 2020	Erweiterung: ARM-Unterstützung für Linux und Ubuntu Server hinzugefügt.
1.1.54.0	6. Januar 2020	Fehlerbehebung: Verarbeitung des Race-Bedingungssze narios von Paketen, die gelöscht werden, wenn das Session Manager-Plugin nicht bereit ist.
1.1.50.0	19. November 2019	Erweiterung: Unterstützung für die Weiterleitung eines Ports an einen lokalen Unix-Socket hinzugefügt.
1.1.35.0	7. November 2019	Verbesserung: (Nur Portweiterleitungssitzungen) Sendet einen TerminateSession Befehl an, SSM Agent wenn der lokale Benutzer drückt. `Ctrl+C`

Version	Datum der Veröffentlichung	Details
1.1.33.0	26. September 2019	Erweiterung: (Nur Port-Weiterleitungssitzungen) Senden Sie ein Trennsignal an den Server, wenn der Client die TCP-Verbindung trennt.
1.1.31.0	6. September 2019	Erweiterung: Aktualisieren Sie, um die Port-Weiterleitungssitzung offen zu halten, bis der Remote-Server die Verbindung schließt.
1.1.26.0	30. Juli 2019	Erweiterung: Begrenzung der Datenübertragungsrate während einer Sitzung aktualisiert.
1.1.23.0	9. Juli 2019	Verbesserung: Unterstützung für die Ausführung von SSH-Sitzungen mit Session Manager hinzugefügt.
1.1.17.0	4. April 2019	Erweiterung: Unterstützung für die zusätzliche Verschlüsselung von Sitzungsdaten mit AWS Key Management Service (AWS KMS) hinzugefügt.
1.0.37.0	20. September 2018	Verbesserung: Fehlerbehebung für Windows-Version.
1.0.0.0	11. September 2018	Erstveröffentlichung des Session Manager-Plugins.

Installieren Sie das Session Manager-Plugin in Windows

Sie können das Session Manager-Plug-In auf Windows Vista oder höher mit dem eigenständigen Installationsprogramm installieren.

Wenn Aktualisierungen veröffentlicht werden, müssen Sie die Installation wiederholen, um die aktuelle Version des Session Manager-Plugins zu erhalten.

> **Note**
>
> Notieren Sie die folgenden Informationen:

- Das Session Manager-Plugin-Installationsprogramm benötigt Administratorrechte, um das Plugin zu installieren.
- Um optimale Ergebnisse zu erzielen, sollten Sie Sitzungen auf Windows-Clients mittels Windows PowerShell-Version 5 oder höher starten. Alternativ hierzu können Sie auch die Befehls-Shell in Windows 10 verwenden. Das Session Manager-Plug-In unterstützt nur PowerShell und die Befehls-Shell. Die Befehlszeilentools von Drittanbietern sind möglicherweise nicht mit dem Plug-In kompatibel.

So installieren Sie das Session Manager-Plug-In mithilfe des EXE-Installationsprogramms

1. Laden Sie das Installationsprogramm über die folgende URL herunter.

   ```
   https://s3.amazonaws.com/session-manager-downloads/plugin/latest/windows/
   SessionManagerPluginSetup.exe
   ```

   Alternativ können Sie eine gezippte Version des Installationsprogramms mit der folgenden URL herunterladen.

   ```
   https://s3.amazonaws.com/session-manager-downloads/plugin/latest/windows/
   SessionManagerPlugin.zip
   ```

2. Führen Sie das heruntergeladene Installationsprogramm aus und folgen Sie den Anweisungen auf dem Bildschirm. Wenn Sie die gezippte Version des Installationsprogramms heruntergeladen haben, müssen Sie zuerst das Installationsprogramm entpacken.

   Lassen Sie das Feld „Installationsspeicherort“ leer, um das Plug-In im Standardverzeichnis zu installieren.

   - `%PROGRAMFILES%\Amazon\SessionManagerPlugin\bin\`

3. Überprüfen Sie, ob die Installation erfolgreich war. Weitere Informationen finden Sie unter Verifizieren der Session Manager-Plugin-Installation.

   > **Note**
   >
   > Wenn Windows die ausführbare Datei nicht finden kann, müssen Sie die Eingabeaufforderung möglicherweise erneut öffnen oder das Installationsverzeichnis der Umgebungsvariablen PATH manuell hinzufügen. Informationen finden Sie

im Fehlerbehebungsthema Session Manager-Plugin wurde nicht automatisch zumBefehlszeilenpfad hinzugefügt (Windows).

Installieren Sie das Session Manager-Plugin in macOS

Wählen Sie eines der folgenden Themen aus, unter dem das Session Manager-Plugin auf macOS installiert werden soll.

> **Note**
>
> Das signierte Installationsprogramm ist eine signierte `.pkg`-Datei. Das mitgelieferte Installationsprogramm verwendet eine `.zip`-Datei. Wenn Sie die Datei entpackt haben, können Sie das Plugin mithilfe der Binärdatei installieren.

Installieren des Session Manager-Plugins in macOS mittels des signierten Installationsprogramms

In diesem Abschnitt wird beschrieben, wie Sie das Session Manager-Plugin mithilfe des signierten Installers auf macOS installieren.

So installieren Sie das Session Manager-Plugin mittels des signierten Installationsprogramms (macOS)

1. Laden Sie das signierte Installationsprogramm herunter.

   x86_64

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/mac/
   session-manager-plugin.pkg" -o "session-manager-plugin.pkg"
   ```

   Mac mit Apple-Halbleiter

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/
   mac_arm64/session-manager-plugin.pkg" -o "session-manager-plugin.pkg"
   ```

2. Führen Sie die Installationsbefehle aus. Wenn der Befehl fehlschlägt, überprüfen Sie, ob der Ordner `/usr/local/bin` vorhanden ist. Ist dies nicht der Fall, erstellen Sie ihn und führen Sie den Befehl erneut aus.

```
sudo installer -pkg session-manager-plugin.pkg -target /
sudo ln -s /usr/local/sessionmanagerplugin/bin/session-manager-plugin /usr/local/
bin/session-manager-plugin
```

3. Überprüfen Sie, ob die Installation erfolgreich war. Weitere Informationen finden Sie unter Verifizieren der Session Manager-Plugin-Installation.

Installieren Sie das Session Manager-Plugin in macOS

In diesem Abschnitt wird beschrieben, wie Sie das Session Manager-Plugin mithilfe des mitgelieferten Installers auf macOS installieren.

> **Important**
>
> Notieren Sie die folgenden wichtigen Informationen.
>
> - Standardmäßig benötigt das Installationsprogramm Sudo-Zugriff, um ausgeführt zu werden, da das Skript das Plugin im `/usr/local/sessionmanagerplugin`-Systemverzeichnis installiert. Wenn Sie das Plugin nicht mit Sudo installieren möchten, aktualisieren Sie das Installationsskript manuell, um das Plugin in einem Verzeichnis zu installieren, für das kein Sudo-Zugriff erforderlich ist.
> - Das gebündelte Installationsprogramm unterstützt keine Installation in Pfaden, die Leerzeichen enthalten.

Sie installieren Sie das Session Manager-Plugin mittels des Paketinstallationsprogramms (macOS)

1. Laden Sie das Paketinstallationsprogramm herunter.

   x86_64

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/mac/
   sessionmanager-bundle.zip" -o "sessionmanager-bundle.zip"
   ```

   Mac mit Apple-Halbleiter

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/
   mac_arm64/sessionmanager-bundle.zip" -o "sessionmanager-bundle.zip"
   ```

2. Entpacken Sie das Paket.

```
unzip sessionmanager-bundle.zip
```

3. Führen Sie den Installationsbefehl aus.

```
sudo ./sessionmanager-bundle/install -i /usr/local/sessionmanagerplugin -b /usr/
local/bin/session-manager-plugin
```

> **Note**
>
> Das Plugin benötigt Python 3.10 oder höher. Standardmäßig wird das Installationsskript unter der Standard-Systemversion von Python ausgeführt. Wenn Sie eine alternative Version von Python installiert haben und diese zum Installieren des Session Manager-Plugins verwenden möchten, führen Sie das Installationsskript mit dieser Version aus, indem Sie den absoluten Pfad der ausführbaren Python-Datei verwenden. Im Folgenden wird ein -Beispiel gezeigt.
>
> ```
> sudo /usr/local/bin/python3.11 sessionmanager-bundle/install -i /usr/local/
> sessionmanagerplugin -b /usr/local/bin/session-manager-plugin
> ```

Das Installationsprogramm installiert das Session Manager-Plugin in `/usr/local/sessionmanagerplugin` und erstellt den Symlink `session-manager-plugin` im Verzeichnis `/usr/local/bin`. Dies beseitigt die Notwendigkeit, das Installationsverzeichnis in der `$PATH`-Variablen des Benutzers anzugeben.

Eine Erklärung der Optionen `-i` und `-b` sehen Sie, indem Sie die Option `-h` verwenden.

```
./sessionmanager-bundle/install -h
```

4. Überprüfen Sie, ob die Installation erfolgreich war. Weitere Informationen finden Sie unter Verifizieren der Session Manager-Plugin-Installation.

> **Note**
>
> Um das Plugin zu deinstallieren, führen Sie die folgenden beiden Befehle in der angegebenen Reihenfolge aus.

```
sudo rm -rf /usr/local/sessionmanagerplugin
```

```
sudo rm /usr/local/bin/session-manager-plugin
```

Installieren des Session Manager-Plugins unter Linux

Dieser Abschnitt enthält Informationen zur Überprüfung der Signatur des Session Manager-Plugin-Installationspakets und zur Installation des Plugins in den folgenden Linux-Distributionen:

- Amazon Linux 2
- AL2023
- RHEL
- Debian Server
- Ubuntu Server

Themen

- Verifizieren der Signatur des Session Manager-Plugins
- Installieren Sie das Session Manager-Plugin auf Amazon Linux 2, Amazon Linux 2023 und Red Hat Enterprise Linux-Distributionen
- Das Session Manager-Plugin in Debian Server und Ubuntu Server installieren

Verifizieren der Signatur des Session Manager-Plugins

Die RPM- und Debian-Installationspakete des Session Manager-Plugins für Linux-Instances sind kryptografisch signiert. Sie können einen öffentlichen Schlüssel verwenden, um zu verifizieren, dass die Binärdatei und das Paket des Plugins original und unverändert sind. Wenn die Dateien beschädigt sind oder verändert wurden, schlägt die Überprüfung fehl. Sie können die Signatur des Installationspakets mithilfe des GNU Privacy Guard (GPG)-Tools überprüfen. Die folgenden Informationen sind für Session Manager-Plugin-Versionen 1.2.707.0 oder höher.

Führen Sie die folgenden Schritte aus, um die Signatur des Session Manager-Plugin-Installationspakets zu überprüfen.

Themen

- Schritt 1: Session Manager-Plugin-Installationspaket herunterladen
- Schritt 2: Zugehörige Signaturdatei herunterladen
- Schritt 3: GPG-Tool installieren
- Schritt 4: Session Manager-Plugin-Installationspaket auf einem Linux-Server überprüfen

Schritt 1: Session Manager-Plugin-Installationspaket herunterladen

Laden Sie das zu prüfende Session Manager-Plugin-Installationspaket herunter.

Amazon Linux 2-, AL2 023- und RHEL RPM-Pakete

x86_64

```
curl -o "session-manager-plugin.rpm" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/linux_64bit/session-manager-plugin.rpm"
```

ARM64

```
curl -o "session-manager-plugin.rpm" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/linux_arm64/session-manager-plugin.rpm"
```

Debian Server- und Ubuntu Server-Deb-Pakete

x86_64

```
curl -o "session-manager-plugin.deb" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_64bit/session-manager-plugin.deb"
```

ARM64

```
curl -o "session-manager-plugin.deb" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_arm64/session-manager-plugin.deb"
```

Schritt 2: Zugehörige Signaturdatei herunterladen

Nachdem Sie das Installationspaket heruntergeladen haben, laden Sie die zugehörige Signaturdatei für die Paketüberprüfung herunter. Um einen zusätzlichen Schutz vor unbefugtem Kopieren

oder Verwenden der session-manager-plugin Binärdatei im Paket zu bieten, bieten wir auch Binärsignaturen an, mit denen Sie einzelne Binärdateien validieren können. Sie können diese binären Signaturen je nach Ihren Sicherheitsanforderungen verwenden.

Amazon Linux 2, AL2 023 und RHEL Signaturpakete

x86_64

Paket:

```
curl -o "session-manager-plugin.rpm.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/linux_64bit/session-manager-plugin.rpm.sig"
```

Binärdatei:

```
curl -o "session-manager-plugin.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/linux_64bit/session-manager-plugin.sig"
```

ARM64

Paket:

```
curl -o "session-manager-plugin.rpm.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/linux_arm64/session-manager-plugin.rpm.sig"
```

Binärdatei:

```
curl -o "session-manager-plugin.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/linux_arm64/session-manager-plugin.sig"
```

Deb-Signaturpakete für Debian Server und Ubuntu Server

x86_64

Paket:

```
curl -o "session-manager-plugin.deb.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_64bit/session-manager-plugin.deb.sig"
```

Binärdatei:

```
curl -o "session-manager-plugin.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_64bit/session-manager-plugin.sig"
```

ARM64

Paket:

```
curl -o "session-manager-plugin.deb.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_arm64/session-manager-plugin.deb.sig"
```

Binärdatei:

```
curl -o "session-manager-plugin.sig" "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_arm64/session-manager-plugin.sig"
```

Schritt 3: GPG-Tool installieren

Um die Signatur des Session Manager-Plugins zu überprüfen, muss das GNU Privacy Guard (GPG)-Tool auf Ihrem System installiert sein. Der Überprüfungsprozess erfordert GPG-Version 2.1 oder höher. Sie können Ihre GPG-Version mit dem folgenden Befehl ermitteln:

```
gpg --version
```

Wenn Ihre Version älter als 2.1 ist, müssen Sie sie aktualisieren, bevor Sie fortfahren können. Bei den meisten Systemen können Sie das GPG-Tool über Ihren Paketmanager aktualisieren. In unterstützten Amazon-Linux- und RHEL-Versionen können Sie zum Beispiel die folgenden Befehle verwenden:

```
sudo yum update
sudo yum install gnupg2
```

Auf unterstützten Ubuntu Server- und Debian Server-Systemen können Sie die folgenden Befehle verwenden:

```
sudo apt-get update
sudo apt-get install gnupg2
```

Sie müssen die erforderliche GPG-Version haben, bevor Sie mit dem Überprüfungsprozess fortfahren.

Schritt 4: Session Manager-Plugin-Installationspaket auf einem Linux-Server überprüfen

Gehen Sie wie folgt vor, um das Session Manager-Plugin-Installationspaket auf einem Linux-Server zu überprüfen.

> **Note**
>
> Amazon Linux 2 unterstützt die GPG-Version 2.1 oder höher nicht. Wenn das folgende Verfahren auf Ihren Amazon-Linux-2-Instances nicht funktioniert, überprüfen Sie die Signatur auf einer anderen Plattform, bevor Sie sie auf Ihren Amazon-Linux-2-Instances installieren.

1. Kopieren Sie den folgenden öffentlichen Schlüssel und speichern Sie ihn in einer Datei namens session-manager-plugin .gpg.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mFIEZ5ERQxMIKoZIzj0DAQcCAwQjuZy+IjFoYg57sLTGhF3aZLBaGpzB+gY6j7Ix
P7NqbpXyjVj8a+dy79gSd64OEaMxUb7vw/jug+CfRXwVGRMNtIBBV1MgU1NNIFNl
c3Npb24gTWFuYWdlciA8c2Vzc2lvbi1tYW5hZ2VyLXBsdWdpbi1zaWduZXJAYW1h
em9uLmNvbT4gKEFXUyBTeXN0ZW1zIE1hbmFnZXIgU2Vzc2lvbiBNYW5hZ2VyIFBs
dWdpbiBMaW51eCBTaWduZXIgS2V5KYkBAAQQEwgAqAUCZ5ERQ4EcQVdTIFNTTSBT
ZXNzaW9uIE1hbmFnZXIgPHNlc3Npb24tbWFuYWdlci1wbHVnaW4tc2lnbmVyQGFt
YXpvbi5jb20+IChBV1MgU3lzdGVtcyBNYW5hZ2VyIFNlc3Npb24gTWFuYWdlciBQ
bHVnaW4gTGludXggU2lnbmVyIEtleSkWIQR5WWNxJM4JOtUB1HosTUr/b2dX7gIe
AwIbAwIVCAAKCRAsTUr/b2dX7rO1AQCa1kig3lQ78W/QHGU76uHx3XAyv0tfpE9U
oQBCIwFLSgEA3PDHt3lZ+s6m9JLGJsy+Cp5ZFzpiF6RgluR/2gA861M=
=2DQm
-----END PGP PUBLIC KEY BLOCK-----
```

2. Importieren Sie den öffentlichen Schlüssel in Ihren Schlüsselbund. Der zurückgegebene Schlüsselwert sollte 2C4D4AFF6F6757EE lauten.

```
$ gpg --import session-manager-plugin.gpg
gpg: key 2C4D4AFF6F6757EE: public key "AWS SSM Session Manager <session-manager-
plugin-signer@amazon.com> (AWS Systems Manager Session Manager Plugin Linux Signer
 Key)" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

3. Führen Sie den folgenden Befehl aus, um den Fingerabdruck zu verifizieren.

```
gpg --fingerprint 2C4D4AFF6F6757EE
```

Der Fingerabdruck für die Befehlsausgabe sollte wie folgt aussehen.

```
7959 6371 24CE 093A D501 D47A 2C4D 4AFF 6F67 57EE
```

```
pub   nistp256 2025-01-22 [SC]
      7959 6371 24CE 093A D501  D47A 2C4D 4AFF 6F67 57EE
uid           [ unknown] AWS SSM Session Manager <session-manager-plugin-
signer@amazon.com> (AWS Systems Manager Session Manager Plugin Linux Signer Key)
```

Wenn keine Übereinstimmung vorliegt, installieren Sie das Plugin nicht. Kontakt. AWS -Support

4. Überprüfen Sie die Installer-Paketsignatur. Ersetzen Sie das *signature-filename* und *downloaded-plugin-filename* durch die Werte, die Sie beim Herunterladen der Signaturdatei angegeben haben session-manager-plugin, und, wie in der Tabelle weiter oben in diesem Thema aufgeführt.

   ```
   gpg --verify signature-filename downloaded-plugin-filename
   ```

   Für die x86_64-Architektur unter Amazon Linux 2 lautet der Befehl wie folgt:

   ```
   gpg --verify session-manager-plugin.rpm.sig session-manager-plugin.rpm
   ```

   Dieser Befehl gibt etwa die folgende Ausgabe zurück.

   ```
   gpg: Signature made Mon Feb 3 20:08:32 2025 UTC gpg: using ECDSA key
    2C4D4AFF6F6757EE
   gpg: Good signature from "AWS Systems Manager Session Manager <session-manager-
   plugin-signer@amazon.com> (AWS Systems Manager Session Manager Plugin Linux Signer
    Key)" [unknown]
   gpg: WARNING: This key is not certified with a trusted signature!
   gpg: There is no indication that the signature belongs to the owner.
   Primary key fingerprint: 7959 6371 24CE 093A D501 D47A 2C4D 4AFF 6F67 57EE
   ```

Wenn die Ausgabe die Bezeichnung `BAD signature`enthält, überprüfen Sie, ob Sie das Verfahren korrekt durchgeführt haben. Wenn Sie weiterhin diese Antwort erhalten, wenden Sie sich an das

Paket AWS -Support und installieren Sie es nicht. Die Vertrauens-Warnmeldung bedeutet nicht, dass die Signatur ungültig ist, sondern nur, dass Sie den öffentlichen Schlüssel nicht überprüft haben. Beachten Sie die Warnung zu vertrauenswürdigen Inhalten. Wenn die Ausgabe die Phrase `Can't check signature: No public key` enthält, überprüfen Sie, ob Sie Session Manager-Plugin-Version 1.2.707.0 oder höher heruntergeladen haben.

Installieren Sie das Session Manager-Plugin auf Amazon Linux 2, Amazon Linux 2023 und Red Hat Enterprise Linux-Distributionen

Gehen Sie wie folgt vor, um das Session Manager Plugin auf Amazon Linux 2, Amazon Linux 2023 (AL2023) und RHEL Distributionen zu installieren.

1. Laden Sie das Session Manager-Plugin-RPM-Paket herunter und installieren Sie es.

   x86_64

   Führen Sie unter Amazon Linux 2 und RHEL 7 den folgenden Befehl aus:

   ```
   sudo yum install -y https://s3.amazonaws.com/session-manager-downloads/plugin/
   latest/linux_64bit/session-manager-plugin.rpm
   ```

   Führen Sie auf AL2 023, RHEL 8 und 9 den folgenden Befehl aus:

   ```
   sudo dnf install -y https://s3.amazonaws.com/session-manager-downloads/plugin/
   latest/linux_64bit/session-manager-plugin.rpm
   ```

   ARM64

   Führen Sie unter Amazon Linux 2 und RHEL 7 den folgenden Befehl aus:

   ```
   sudo yum install -y https://s3.amazonaws.com/session-manager-downloads/plugin/
   latest/linux_arm64/session-manager-plugin.rpm
   ```

   Führen Sie auf AL2 023 und RHEL 8 und 9 den folgenden Befehl aus:

   ```
   sudo dnf install -y https://s3.amazonaws.com/session-manager-downloads/plugin/
   latest/linux_arm64/session-manager-plugin.rpm
   ```

2. Überprüfen Sie, ob die Installation erfolgreich war. Weitere Informationen finden Sie unter Verifizieren der Session Manager-Plugin-Installation.

> **Note**
>
> Wenn Sie das Plugin deinstallieren möchten, führen Sie `sudo yum erase session-manager-plugin -y` aus.

Das Session Manager-Plugin in Debian Server und Ubuntu Server installieren

1. Laden Sie das Session Manager-Plug-In-deb-Paket herunter.

   x86_64

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_64bit/session-manager-plugin.deb" -o "session-manager-plugin.deb"
   ```

   ARM64

   ```
   curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_arm64/session-manager-plugin.deb" -o "session-manager-plugin.deb"
   ```

2. Führen Sie den Installationsbefehl aus.

   ```
   sudo dpkg -i session-manager-plugin.deb
   ```

3. Überprüfen Sie, ob die Installation erfolgreich war. Weitere Informationen finden Sie unter Verifizieren der Session Manager-Plugin-Installation.

> **Note**
>
> Wenn Sie das Plugin jemals deinstallieren möchten, führen Sie `sudo dpkg -r session-manager-plugin` aus.

Verifizieren der Session Manager-Plugin-Installation

Führen Sie die folgenden Befehle aus, um zu überprüfen, ob das Session Manager-Plugin erfolgreich installiert wurde.

```
session-manager-plugin
```

Wenn die Installation erfolgreich war, wird die folgende Meldung zurückgegeben.

```
The Session Manager plugin is installed successfully. Use the AWS CLI to start a
 session.
```

Sie können die Installation auch testen, indem Sie den start-session-Befehl in der AWS Command Line Interface (AWS CLI) ausführen. Ersetzen Sie im folgenden Befehl *instance-id* durch Ihre eigenen Informationen.

```
aws ssm start-session --target instance-id
```

Dieser Befehl funktioniert nur AWS CLI, wenn Sie den installiert und konfiguriert haben und Ihr Session Manager Administrator Ihnen die erforderlichen IAM-Berechtigungen für den Zugriff auf den verwalteten Zielknoten erteilt hat. Session Manager

Session Manager-Plugin in GitHub

Der Quellcode für das Session Manager-Plugin ist auf GitHub verfügbar, damit Sie das Plugin entsprechend Ihren Anforderungen anpassen können. Wir möchten Sie bitten, uns eventuelle Änderungswünsche mitzuteilen. Amazon Web Services bietet jedoch keine Unterstützung für die Ausführung modifizierter Kopien dieser Software.

(Optional) Aktivieren Sie die Session Manager-Plug-In-Protokollierung

Das Session Manager-Plug-In enthält eine Option zum Aktivieren der Protokollierung für von Ihnen ausgeführte Sitzungen. Standardmäßig ist die Protokollierung deaktiviert.

Wenn Sie die Protokollierung aktivieren, erstellt das Session Manager-Plug-In Protokolldateien sowohl für Anwendungsaktivitäten (`session-manager-plugin.log`) als auch für Fehler (`errors.log`) auf Ihrem lokalen Computer.

Themen

- Aktivieren der Protokollierung für das Session Manager-Plug-In (Windows)
- Aktivieren der Protokollierung für das Session Manager-Plug-In (Linux und macOS)

Aktivieren der Protokollierung für das Session Manager-Plug-In (Windows)

1. Suchen Sie die Datei `seelog.xml.template` für das Plug-In.

Der Standardspeicherort ist `C:\Program Files\Amazon\SessionManagerPlugin\seelog.xml.template`.

2. Ändern Sie den Namen der Datei in `seelog.xml`.
3. Öffnen Sie die Datei und ändern Sie `minlevel="off"` in `minlevel="info"` oder `minlevel="debug"`.

> **Note**
>
> Standardmäßig werden Protokolleinträge zum Öffnen eines Datenkanals und Neuverbinden von Sitzungen auf INFO-Ebene aufgezeichnet. Datenflusseinträge (Pakete und Bestätigung) werden auf DEBUG-Ebene aufgezeichnet.

4. Ändern Sie andere Konfigurationsoptionen, die Sie ändern möchten. Optionen, die Sie ändern können, sind:

   - Debug-Ebene: Sie können die Debug-Ebene von `formatid="fmtinfo"` in `formatid="fmtdebug"` ändern.
   - DieProtokolldateioptionen: Sie können die Protokolldateioptionen einschließlich des Speicherorts der Protokolle ändern, jedoch nicht die Namen von Protokolldateien.

   > **Important**
   >
   > Sie dürfen die Dateinamen nicht ändern. Wenn Sie dies tun, funktioniert die Protokollierung nicht korrekt.

   ```
   <rollingfile type="size" filename="C:\Program Files\Amazon\SessionManagerPlugin
   \Logs\session-manager-plugin.log" maxsize="30000000" maxrolls="5"/>
   <filter levels="error,critical" formatid="fmterror">
   <rollingfile type="size" filename="C:\Program Files\Amazon\SessionManagerPlugin
   \Logs\errors.log" maxsize="10000000" maxrolls="5"/>
   ```

5. Speichern Sie die Datei.

Aktivieren der Protokollierung für das Session Manager-Plug-In (Linux und macOS)

1. Suchen Sie die Datei `seelog.xml.template` für das Plug-In.

Der Standardspeicherort ist `/usr/local/sessionmanagerplugin/seelog.xml.template`.

2. Ändern Sie den Namen der Datei in `seelog.xml`.
3. Öffnen Sie die Datei und ändern Sie `minlevel="off"` in `minlevel="info"` oder `minlevel="debug"`.

> **Note**
>
> Standardmäßig werden Protokolleinträge zum Öffnen von Datenkanälen und Neuverbinden von Sitzungen auf INFO-Ebene aufgezeichnet. Datenflusseinträge (Pakete und Bestätigung) werden auf DEBUG-Ebene aufgezeichnet.

4. Ändern Sie andere Konfigurationsoptionen, die Sie ändern möchten. Optionen, die Sie ändern können, sind:

   - Debug-Ebene: Sie können die Debug-Ebene von `formatid="fmtinfo"` in `outputs formatid="fmtdebug"` ändern.
   - DieProtokolldateioptionen: Sie können die Protokolldateioptionen einschließlich des Speicherorts der Protokolle ändern, jedoch nicht die Namen von Protokolldateien.

   > **Important**
   >
   > Sie dürfen die Dateinamen nicht ändern. Wenn Sie dies tun, funktioniert die Protokollierung nicht korrekt.

   ```
   <rollingfile type="size" filename="/usr/local/sessionmanagerplugin/logs/session-manager-plugin.log" maxsize="30000000" maxrolls="5"/>
   <filter levels="error,critical" formatid="fmterror">
   <rollingfile type="size" filename="/usr/local/sessionmanagerplugin/logs/errors.log" maxsize="10000000" maxrolls="5"/>
   ```

   > **Important**
   >
   > Wenn Sie das angegebene Standardverzeichnis für das Speichern von Protokollen verwenden, müssen Sie Sitzungsbefehle entweder über sudo ausführen oder dem Verzeichnis, in dem das Plug-In installiert ist, vollständige Lese- und

Schreibberechtigungen erteilen. Um diese Einschränkungen zu umgehen, ändern Sie den Speicherort, an dem die Protokolle gespeichert werden.

5. Speichern Sie die Datei.

Starten einer Sitzung

Sie können die AWS Systems Manager Konsole, die Amazon Elastic Compute Cloud (Amazon EC2) -Konsole, die AWS Command Line Interface (AWS CLI) oder SSH verwenden, um eine Sitzung zu starten.

Themen

- Starten einer Sitzung (Systems Manager-Konsole)
- Eine Sitzung starten (EC2Amazon-Konsole)
- Starten einer Sitzung (AWS CLI)
- Starten einer Sitzung (SSH)
- Starten einer Sitzung (Port-Weiterleitung)
- Starten einer Sitzung (Port-Weiterleitung an entfernten Host)
- Starten einer Sitzung (interaktive und nicht interaktive Befehle)

Starten einer Sitzung (Systems Manager-Konsole)

Sie können die AWS Systems Manager Konsole verwenden, um eine Sitzung mit einem verwalteten Knoten in Ihrem Konto zu starten.

> **Note**
>
> Bevor Sie eine Sitzung beginnen, stellen Sie sicher, dass Sie die Einrichtungsschritte für Session Manager ausgeführt haben. Weitere Informationen finden Sie unter Einrichten von Session Manager.

Starten einer Sitzung (Systems-Manager-Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie Sitzung starten aus.
4. (Optional) Geben Sie eine Beschreibung für die Sitzung im Feld Grund für die Sitzung ein.
5. Aktivieren Sie unter Ziel-Instances das Optionsfeld links neben dem verwalteten Knoten aus, mit dem Sie eine Verbindung herstellen möchten.

   Wenn der gewünschte Knoten nicht in der Liste enthalten ist oder wenn Sie einen Knoten auswählen und einen Konfigurationsfehler erhalten, lesen Sie die Schritte zur Fehlerbehebung unter Verwalteter Knoten ist nicht verfügbar oder nicht für Session Manager konfiguriert.
6. Wählen Sie Sitzung starten, um die Sitzung sofort zu starten.

   –oder–

   Wählen Sie Weiter für die Sitzungsoptionen.
7. (Optional) Wählen Sie unter Sitzungsdokument das Dokument aus, das Sie zu Beginn der Sitzung ausführen möchten. Wenn Ihr Dokument Laufzeitparameter unterstützt, können Sie in jedes Parameterfeld einen oder mehrere durch Komma getrennte Werte eingeben.
8. Wählen Sie Weiter aus.
9. Wählen Sie Sitzung starten aus.

Nach der Herstellung der Verbindung können Sie Bash-Befehle (Linux und macOS) oder PowerShell-Befehle (Windows) wie für jeden anderen Verbindungstyp ausführen.

> **Important**
>
> Wenn Sie Benutzern die Möglichkeit geben möchten, beim Starten von Sitzungen in der Session-Manager-Konsole ein Dokument anzugeben, beachten Sie Folgendes:
>
> - Sie müssen Benutzern die in ihrer IAM-Richtlinie festgelegten Berechtigungen `ssm:GetDocument` und `ssm:ListDocuments` gewähren. Weitere Informationen finden Sie unter Zugriff auf benutzerdefinierte Sitzungsdokumente in der Konsole gewähren.
> - Die Konsole unterstützt nur Sitzungsdokumente, für die der `sessionType` als `Standard_Stream` definiert ist. Weitere Informationen finden Sie unter Schema des Sitzungsdokuments.

Eine Sitzung starten (EC2Amazon-Konsole)

Sie können die Amazon Elastic Compute Cloud (Amazon EC2) -Konsole verwenden, um eine Sitzung mit einer Instance in Ihrem Konto zu starten.

> **Note**
>
> Wenn Sie den Fehler erhalten, dass Sie nicht berechtigt sind, eine oder mehrere Systems Manager (`ssm:command-name`)-Aktionen auszuführen, müssen Sie sich an Ihren Administrator wenden. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt. Bitten Sie diese Person, Ihre Richtlinien zu aktualisieren, damit Sie Sitzungen von der EC2 Amazon-Konsole aus starten können. Wenn Sie ein Administrator sind, finden Sie weitere Informationen unter Muster-IAM-Richtlinien für Session Manager.

Um eine Sitzung zu starten (EC2 Amazon-Konsole)

1. Öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie die Instance und Connect (Verbinden) aus.
4. Für Connection method (Verbindungsmethode) wählen Sie Session Manager.
5. Wählen Sie Connect aus.

Nach der Herstellung der Verbindung können Sie Bash-Befehle (Linux und macOS) oder PowerShell-Befehle (Windows) wie für jeden anderen Verbindungstyp ausführen.

Starten einer Sitzung (AWS CLI)

Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

Bevor Sie eine Sitzung beginnen, stellen Sie sicher, dass Sie die Einrichtungsschritte für Session Manager ausgeführt haben. Weitere Informationen finden Sie unter Einrichten von Session Manager.

Um die Befehle AWS CLI to run session verwenden zu können, muss das Session Manager Plugin auch auf Ihrem lokalen Computer installiert sein. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugin für die AWS CLI.

Um eine Sitzung mit dem zu starten AWS CLI, führen Sie den folgenden Befehl aus und *instance-id* ersetzen Sie ihn durch Ihre eigenen Informationen.

```
aws ssm start-session \
    --target instance-id
```

Informationen zu anderen Optionen, die Sie mit dem start-session Befehl verwenden können, finden Sie start-sessionim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Starten einer Sitzung (SSH)

Um eine Session Manager-SSH-Sitzung zu starten, muss Version 2.3.672.0 oder höher von SSM Agent auf dem verwalteten Knoten installiert sein.

Anforderungen für SSH-Verbindungen

Beachten Sie die folgenden Anforderungen und Einschränkungen für Sitzungsverbindungen über Session Manager SSH:

- Ihr anvisierter verwalteter Knoten muss so konfiguriert sein, dass er SSH-Verbindungen unterstützt. Weitere Informationen finden Sie unter (Optional) Berechtigungen für SSH-Verbindungen über Session Manager aktivieren und steuern.
- Sie müssen mithilfe des Kontos des verwalteten Knotens verbinden, der dem Privacy Enhanced Mail (PEM)-Zertifikat zugeordnet ist, und nicht dem `ssm-user`-Konto, das für andere Arten von Sitzungsverbindungen verwendet wird. Bei EC2 Instanzen für Linux und macOS ist `ec2-user` der Standardbenutzer beispielsweise. Informationen zur Identifizierung des Standardbenutzers für jeden Instance-Typ finden Sie unter Get Information About Your Instance im EC2 Amazon-Benutzerhandbuch.
- Protokollieren ist für Session Manager-Sitzungen, die eine Verbindung über Port-Weiterleitung oder SSH herstellen, nicht verfügbar. Das liegt daran, dass SSH alle Sitzungsdaten innerhalb der sicheren TLS-Verbindung zwischen den Endpunkten AWS CLI und den Session Manager Endpunkten verschlüsselt und Session Manager nur als Tunnel für SSH-Verbindungen dient.

> **Note**
>
> Bevor Sie eine Sitzung beginnen, stellen Sie sicher, dass Sie die Einrichtungsschritte für Session Manager ausgeführt haben. Weitere Informationen finden Sie unter Einrichten von Session Manager.

Um eine Sitzung über SSH zu starten, führen Sie folgenden Befehl aus. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

```
ssh -i /path/my-key-pair.pem username@instance-id
```

> **Tip**
>
> Wenn Sie eine Sitzung mit SSH starten, können Sie mit dem folgenden Befehl lokale Dateien auf den anvisierten verwalteten Knoten kopieren.
>
> ```
> scp -i /path/my-key-pair.pem /path/ExampleFile.txt username@instance-id:~
> ```

Informationen zu anderen Optionen, die Sie mit dem start-session Befehl verwenden können, finden Sie start-sessionim AWS Systems Manager Abschnitt der Befehlsreferenz. AWS CLI

Starten einer Sitzung (Port-Weiterleitung)

Um eine Session Manager-Port-Weiterleitungs-Sitzung zu starten, muss Version 2.3.672.0 oder höher des SSM Agent auf dem verwalteten Knoten installiert sein.

> **Note**
>
> Bevor Sie eine Sitzung beginnen, stellen Sie sicher, dass Sie die Einrichtungsschritte für Session Manager ausgeführt haben. Weitere Informationen finden Sie unter Einrichten von Session Manager.
> Um die Befehle AWS CLI to run session verwenden zu können, müssen Sie das Session Manager Plug-in auf Ihrem lokalen Computer installieren. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugin für die AWS CLI.

Je nach Betriebssystem und Befehlszeilentool kann die Platzierung von Anführungszeichen unterschiedlich sein, und möglicherweise sind Escapezeichen erforderlich.

Um eine Port-Weiterleitungssitzung zu starten, führen Sie den folgenden Befehl in der CLI aus. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name AWS-StartPortForwardingSession \
    --parameters '{"portNumber":["80"], "localPortNumber":["56789"]}'
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name AWS-StartPortForwardingSession ^
    --parameters portNumber="3389",localPortNumber="56789"
```

`portNumber` ist der Remote-Port auf dem verwalteten Knoten, an den der Sitzungsverkehr umgeleitet werden soll. Sie können beispielsweise Port `3389` für die Verbindung zu einem Windows-Knoten mithilfe des Remote Desktop Protocol (RDP) angeben. Wenn Sie den `portNumber`-Parameter nicht angeben, verwendet Session Manager die `80`-Standardschulungsparameter.

`localPortNumber` ist der Port auf Ihrem lokalen Computer, an dem der Verkehr beginnt, z. B. `56789` Dieser Wert ist, was Sie eingeben, wenn Sie eine Verbindung mit einem verwalteten Knoten über einen Client herstellen. Beispiel, **`localhost:56789`**.

Informationen zu anderen Optionen, die Sie mit dem start-session Befehl verwenden können, finden Sie start-sessionim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Weitere Informationen zu Port-Weiterleitungssitzungen finden Sie unter Port-Weiterleitung unter Verwendung von AWS Systems ManagerSession Manager im AWS News Blog.

Starten einer Sitzung (Port-Weiterleitung an entfernten Host)

Um eine Session Manager-Port-Weiterleitungs-Sitzung zu einem entfernten Host zu starten, muss Version 3.1.1374.0 oder höher des SSM Agent auf dem verwalteten Knoten installiert sein. Der Remote-Host muss nicht von Systems Manager verwaltet werden.

> **Note**
>
> Bevor Sie eine Sitzung beginnen, stellen Sie sicher, dass Sie die Einrichtungsschritte für Session Manager ausgeführt haben. Weitere Informationen finden Sie unter Einrichten von Session Manager.
> Um die Befehle AWS CLI to run session verwenden zu können, müssen Sie das Session Manager Plug-in auf Ihrem lokalen Computer installieren. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugin für die AWS CLI.
> Je nach Betriebssystem und Befehlszeilentool kann die Platzierung von Anführungszeichen unterschiedlich sein, und möglicherweise sind Escapezeichen erforderlich.

Um eine Port-Weiterleitungssitzung zu starten, führen Sie den folgenden Befehl in der AWS CLI aus. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name AWS-StartPortForwardingSessionToRemoteHost \
    --parameters '{"host":["mydb.example.us-east-2.rds.amazonaws.com"],"portNumber":
["3306"], "localPortNumber":["3306"]}'
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name AWS-StartPortForwardingSessionToRemoteHost ^
    --parameters host="mydb.example.us-
east-2.rds.amazonaws.com",portNumber="3306",localPortNumber="3306"
```

Der `host`-Wert stellt den Hostnamen oder die IP-Adresse des Remote-Hosts dar, zu dem Sie eine Verbindung herstellen möchten. Es gelten weiterhin allgemeine Anforderungen an Konnektivität und Namensauflösung zwischen dem verwalteten Knoten und dem Remote-Host.

`portNumber` ist der Remote-Port auf dem verwalteten Knoten, an den der Sitzungsverkehr umgeleitet werden soll. Sie können beispielsweise Port `3389` für die Verbindung zu einem Windows-Knoten mithilfe des Remote Desktop Protocol (RDP) angeben. Wenn Sie den `portNumber`-Parameter nicht angeben, verwendet Session Manager die `80`-Standardschulungsparameter.

`localPortNumber` ist der Port auf Ihrem lokalen Computer, an dem der Verkehr beginnt, z. B. `56789` Dieser Wert ist, was Sie eingeben, wenn Sie eine Verbindung mit einem verwalteten Knoten über einen Client herstellen. Beispiel, **`localhost:56789`**.

Informationen zu anderen Optionen, die Sie mit dem start-session Befehl verwenden können, finden Sie start-sessionim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Starten einer Sitzung mit einer Amazon-ECS-Aufgabe

Session Manager unterstützt das Starten einer Portweiterleitungssitzung mit einer Aufgabe in einem Amazon Elastic Container Service (Amazon ECS)-Cluster. Aktivieren Sie dazu ECS Exec. Weitere Informationen finden Sie unter Überwachen von Amazon-ECS-Containern mit ECS Exec im Entwicklerhandbuch zum Amazon Elastic Container Service.

Sie müssen ebenso die Aufgabenrolle in IAM aktualisieren, sodass sie die folgenden Berechtigungen enthält:

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
       {
       "Effect": "Allow",
       "Action": [
            "ssmmessages:CreateControlChannel",
            "ssmmessages:CreateDataChannel",
            "ssmmessages:OpenControlChannel",
            "ssmmessages:OpenDataChannel"
       ],
      "Resource": "*"
      }
```

```
    ]
}
```

Um eine Port-Weiterleitungssitzung mit einer Amazon-ECS-Aufgabe zu starten, führen Sie den folgenden Befehl in der AWS CLI aus. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

> **Note**
>
> Entfernen Sie die <- und >-Symbole aus dem `target`-Parameter. Diese Symbole dienen nur zur Verdeutlichung des Lesers.

Linux & macOS

```
aws ssm start-session \
    --target ecs:<ECS_cluster_name>_<ECS_container_ID>_<container_runtime_ID> \
    --document-name AWS-StartPortForwardingSessionToRemoteHost \
    --parameters '{"host":["URL"],"portNumber":["port_number"], "localPortNumber":
["port_number"]}'
```

Windows

```
aws ssm start-session ^
    --target ecs:<ECS_cluster_name>_<ECS_container_ID>_<container_runtime_ID> ^
    --document-name AWS-StartPortForwardingSessionToRemoteHost ^
    --parameters host="URL",portNumber="port_number",localPortNumber="port_number"
```

Starten einer Sitzung (interaktive und nicht interaktive Befehle)

Bevor Sie eine Sitzung beginnen, stellen Sie sicher, dass Sie die Einrichtungsschritte für Session Manager ausgeführt haben. Weitere Informationen finden Sie unter Einrichten von Session Manager.

Um die Befehle AWS CLI to run session verwenden zu können, muss das Session Manager Plugin auch auf Ihrem lokalen Computer installiert sein. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugin für die AWS CLI.

Um eine Interactive Befehls-Sitzung zu starten, führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

Linux & macOS

```
aws ssm start-session \
    --target instance-id \
    --document-name CustomCommandSessionDocument \
    --parameters '{"logpath":["/var/log/amazon/ssm/amazon-ssm-agent.log"]}'
```

Windows

```
aws ssm start-session ^
    --target instance-id ^
    --document-name CustomCommandSessionDocument ^
    --parameters logpath="/var/log/amazon/ssm/amazon-ssm-agent.log"
```

Informationen zu anderen Optionen, die Sie mit dem start-session Befehl verwenden können, finden Sie start-sessionim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

Weitere Informationen

- Verwenden Sie Port-Forwarding in AWS Systems ManagerSession Manager, um eine Verbindung zu Remote-Hosts herzustellen
- EC2 Amazon-Instance-Portweiterleitung mit AWS Systems Manager
- AWS Verwaltete Microsoft AD-Ressourcen mit Session Manager Portweiterleitung verwalten
- Port-Weiterleitung mit AWS Systems ManagerSession Manager im AWS News Blog.

Eine Sitzung beenden

Sie können mithilfe der AWS Systems Manager-Konsole oder der AWS Command Line Interface (AWS CLI) eine Sitzung beenden, die Sie in Ihrem Konto gestartet haben. Wenn Sie in der Konsole auf die Schaltfläche Beenden für eine Sitzung klicken oder die API-Aktion TerminateSession mithilfe von AWS CLI aufrufen, wird Session Manager die Sitzung dauerhaft beenden und die Datenverbindung zwischen dem Session Manager-Client und SSM Agent auf dem verwalteten Knoten wird geschlossen. Sie können eine beendete Sitzung nicht fortsetzen.

Wenn in einer geöffneten Sitzung 20 Minuten lang keine Benutzeraktivität stattfindet, löst der Ruhezustand ein Timeout aus. Session Manager ruft `TerminateSession` nicht auf, schließt aber den zugrunde liegenden Kanal. Sie können eine Sitzung, die aufgrund eines Timeouts im Leerlauf geschlossen wurde, nicht fortsetzen.

Wir empfehlen, eine Sitzung immer explizit zu beenden, indem Sie den `terminate-session`-Befehl verwenden, wenn Sie die AWS CLI verwenden, oder die Schaltfläche Beenden, wenn Sie die Konsole verwenden. (Die Schaltflächen zum Beenden befinden sich sowohl im Sitzungsfenster als auch auf der Hauptseite der Session Manager-Konsole.) Wenn Sie nur ein Browser- oder Befehlsfenster schließen, wird die Sitzung in der Konsole 30 Tage lang als Aktiv aufgeführt. Wenn Sie eine Sitzung nicht explizit beenden oder wenn das Zeitlimit für eine Sitzung überschritten wird, werden alle Prozesse, die zu diesem Zeitpunkt auf dem verwalteten Knoten ausgeführt wurden, weiterhin ausgeführt.

Themen

- So beenden Sie eine Sitzung (Konsole)
- Beenden einer Sitzung (AWS CLI)

So beenden Sie eine Sitzung (Konsole)

Sie können mithilfe der AWS Systems Manager-Konsole eine Sitzung in Ihrem Konto beenden.

So beenden Sie eine Sitzung (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie unter Sessions (Sitzungen) die Optionsschaltfläche links neben der Sitzung aus, die Sie beenden möchten.
4. Wähen Sie Beenden.

Beenden einer Sitzung (AWS CLI)

Um eine Sitzung über die AWS CLI zu beenden, führen Sie folgenden Befehl aus. Ersetzen Sie *`session-id`* mit Ihren eigenen Informationen.

```
aws ssm terminate-session \
    --session-id session-id
```

Weitere Informationen zum Befehl terminate-session finden Sie unter terminate-session im Abschnitt AWS Systems Manager der Befehlsreferenz AWS CLI.

Anzeigen des Sitzungsverlaufs

Sie können die AWS Systems Manager-Konsole oder die AWS Command Line Interface (AWS CLI) verwenden, um Informationen zu Sitzungen in Ihrem Konto anzuzeigen. In der Konsole können Sie Sitzungsdetails wie die folgenden anzeigen:

- Die ID der Sitzung
- Welche Benutzer über eine Sitzung eine Verbindung mit einem verwalteten Knoten hergestellt haben
- Die ID des verwalteten Knotens
- Wann die Sitzung gestartet und beendet wurde
- Den Status der Sitzung
- Den für das Speichern von Sitzungsprotokollen angegebenen Speicherort (wenn aktiviert)

Über die AWS CLI können Sie eine Liste der Sessions in Ihrem Konto anzeigen, jedoch nicht die zusätzlichen Details, die in der Konsole verfügbar sind.

Informationen zur Protokollierung des Sitzungsverlaufs finden Sie unter Protokollierung von Sitzungen aktivieren und deaktivieren.

Themen

- Anzeigen des Sitzungsverlaufs (Konsole)
- Anzeigen des Sitzungsverlaufs (AWS CLI)

Anzeigen des Sitzungsverlaufs (Konsole)

Sie können die AWS Systems Manager-Konsole verwenden, um Details zu den Sitzungen in Ihrem Konto anzuzeigen.

So zeigen Sie den Sitzungsverlauf an (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Session history (Sitzungsverlauf) aus.

   –oder–

Wenn die Session Manager-Startseite zum ersten Mal geöffnet wird, wählen Sie Einstellungen konfigurieren und dann die Registerkarte Sitzungsverlauf aus.

Anzeigen des Sitzungsverlaufs (AWS CLI)

Um eine Liste der Sitzungen in Ihrem Konto über die AWS CLI anzuzeigen, führen Sie folgenden Befehl aus.

```
aws ssm describe-sessions \
    --state History
```

> **Note**
>
> Dieser Befehl gibt nur Ergebnisse für Verbindungen zu Zielen zurück, die mit Session Manager initiiert wurden. Es werden keine Verbindungen aufgeführt, die auf andere Weise hergestellt wurden, z. B. Remote Desktop Protocol (RDP) oder Secure Shell Protocol (SSH).

Informationen zu anderen Optionen, die Sie mit dem Befehl describe-sessions verwenden können, finden Sie unter describe-sessions im Abschnitt AWS Systems Manager der AWS CLI-Command Reference.

Protokollieren von Sitzungsaktivitäten

Zusätzlich zur Bereitstellung von Informationen zu den aktuellen und abgeschlossenen Sitzungen in der Systems-Manager-Konsole stellt Session Manager Ihnen Optionen für das Protokoll von Sitzungsaktivitäten in Ihrem AWS-Konto mit AWS CloudTrail bereit.

CloudTrail erfasst Sitzungs-API-Aufrufe, die über die Systems Manager-Konsole erfolgt sind, die AWS Command Line Interface (AWS CLI) und das Systems Manager SDK. Sie können die Informationen in der CloudTrail-Konsole anzeigen oder sie in einem angegebenen Amazon Simple Storage Service (Amazon S3)-Bucket speichern. Für alle CloudTrail-Protokolle in Ihrem Konto wird nur ein Amazon S3-Bucket benötigt. Weitere Informationen finden Sie unter Protokollierung von AWS Systems Manager-API-Aufrufen mit AWS CloudTrail.

> **Note**
>
> Für wiederkehrende, historische, analytische Analysen Ihrer Protokolldateien sollten Sie erwägen, CloudTrail-Protokolle mithilfe von CloudTrail Lake oder einer von Ihnen verwalteten Tabelle abzufragen. Weitere Informationen finden Sie unter Abfragen von AWS CloudTrail-Protokollen im AWS CloudTrail-Benutzerhandbuch.

Überwachen der Sitzungsaktivität mithilfe von Amazon EventBridge (Konsole)

EventBridge ermöglicht Ihnen das Einrichten von Regeln, um zu erkennen, wenn Änderungen bei den AWS-Ressourcen eintreten. Sie können eine Regel erstellen, um zu erkennen, wenn ein Benutzer in Ihrer Organisation eine Sitzung startet oder beendet, und dann z. B. über Amazon SNS eine Benachrichtigung bezüglich des Ereignisses erhalten.

Unterstützung von EventBridge für Session Manager basiert auf Datensätzen von API-Operationen, die von CloudTrail aufgezeichnet wurden. (Sie können mittels der CloudTrail-Integration mit EventBridge auf die meisten AWS Systems Manager-Ereignisse reagieren.) Aktionen, die innerhalb einer Sitzung stattfinden, z. B. ein `exit`-Befehl, die keinen API-Aufruf ausführen, werden von EventBridge nicht erkannt.

Die folgenden Schritte zeigen, wie Sie Benachrichtigungen über Amazon Simple Notification Service (Amazon SNS) initiieren, wenn ein Session ManagerAPI-Ereignis eintritt, z. B. StartSession.

So überwachen Sie die Sitzungsaktivität mithilfe von Amazon EventBridge (Konsole)

1. Erstellen Sie ein Amazon SNS-Thema zum Senden von Benachrichtigungen, wenn das Session Manager-Ereignis eintritt, die Sie überwachen möchten.

   Weitere Informationen finden Sie unter Erstellen eines Themas im Amazon Simple Notification Service-Entwicklerhandbuch.
2. Erstellen Sie eine EventBridge-Regel zum Aufrufen des Amazon SNS-Ziels für den Typ des Session Manager-Ereignisses, die Sie nachverfolgen möchten.

   Weitere Informationen finden Sie unter Erstellen von Amazon-EventBridge-Regeln, die auf Ereignisse reagieren im Amazon-EventBridge-Benutzerhandbuch.

   Wählen Sie während der Erstellung der Regel die folgenden Optionen aus:

   - Wählen Sie für AWS service (-Service), die Option Systems Manager aus.

- Wählen Sie für Event Type (Ereignistyp) die Option AWS API Call through CloudTrail (-API-Aufruf via CloudTrail) aus.
- Wählen Sie Specific operation (s) (Spezifische Operation(en)) aus und geben Sie dann nacheinander die Session Manager-Befehle ein, für die Sie Benachrichtigungen erhalten möchten. Sie können StartSession, ResumeSession und TerminateSession auswählen. (EventBridge unterstützt nicht die Befehle `Get*`, `List*` und `Describe*`.)
- Für Select a target (Wählen Sie ein Ziel aus), wählen Sie SNS-Thema aus. Wählen Sie unter Topic (Thema) den Namen des von Ihnen in Schritt 1 erstellten Amazon SNS-Themas aus.

Weitere Informationen finden Sie unter Amazon EventBridge-Benutzerhandbuch im Handbuch „Erste Schritte“ für Amazon Simple Notification Service.

Protokollierung von Sitzungen aktivieren und deaktivieren

Die Sitzungsprotokollierung zeichnet Informationen über aktuelle und abgeschlossene Sitzungen in der Systems-Manager-Konsole auf. Sie können auch Details zu Befehlen protokollieren, die während Sitzungen in Ihrem AWS-Konto ausgeführt werden. Mithilfe der Sitzungsprotokollierung können Sie Folgendes tun:

- Erstellen und Speichern von Sitzungsprotokollen zu Archivierungszwecken.
- Generieren eines Berichts mit Details zu jeder Verbindung, die mit Ihren verwalteten Knoten über Session Manager in den letzten 30 Tagen hergestellt wurde.
- Generieren Sie Benachrichtigungen für die Sitzungsprotokollierung in Ihren Benachrichtigungen AWS-Konto, z. B. Amazon Simple Notification Service (Amazon SNS).
- Initiieren Sie automatisch eine weitere Aktion für eine AWS Ressource als Ergebnis von Aktionen, die während einer Sitzung ausgeführt wurden, z. B. das Ausführen einer AWS Lambda Funktion, das Starten einer AWS CodePipeline Pipeline oder das Ausführen eines AWS Systems Manager Run Command Dokuments.

> **Important**
>
> Beachten Sie die folgenden Anforderungen und Einschränkungen für Session Manager:
>
> - Session Manager protokolliert die von Ihnen eingegebenen Befehle und deren Ausgabe während einer Sitzung abhängig von Ihren Sitzungseinstellungen. Um zu verhindern, dass vertrauliche Daten wie Passwörter in Ihren Sitzungsprotokollen angezeigt werden,

empfehlen wir die folgenden Befehle, wenn Sie während einer Sitzung vertrauliche Daten eingeben.

Linux & macOS

```
stty -echo; read passwd; stty echo;
```

Windows

```
$Passwd = Read-Host -AsSecureString
```

- Wenn Sie Windows Server 2012 oder früher verwenden, werden die Daten in Ihren Protokollen möglicherweise nicht optimal formatiert. Wir empfehlen die Verwendung von Windows Server 2012 R2 und höher, um optimal formatierte Protokolle zu erhalten.
- Wenn Sie Linux- oder macOS-verwaltete Knoten verwenden, muss das Screen-Serviceprogramm installiert sein. Wenn dies nicht der Fall ist, werden die Protokolldaten möglicherweise abgeschnitten. Unter Amazon Linux AL2 2.023 und ist Ubuntu Server das Screen Utility standardmäßig installiert. Um Screen manuell zu installieren, müssen Sie abhängig von Ihrer Linux-Version entweder `sudo yum install screen` oder `sudo apt-get install screen` ausführen.
- Protokollieren ist für Session Manager-Sitzungen, die eine Verbindung über Port-Weiterleitung oder SSH herstellen, nicht verfügbar. Das liegt daran, dass SSH alle Sitzungsdaten innerhalb der sicheren TLS-Verbindung zwischen den Session Manager Endpunkten verschlüsselt AWS CLI und Session Manager nur als Tunnel für SSH-Verbindungen dient.

Weitere Informationen zu den Berechtigungen, die für die Verwendung von Amazon S3 oder Amazon CloudWatch Logs für die Protokollierung von Sitzungsdaten erforderlich sind, finden Sie unterErstellen einer IAM-Rolle mit Berechtigungen für Amazon S3 Session Manager und CloudWatch Logs (Konsole).

Weitere Informationen zu Protokollierungsoptionen für Session Manager finden Sie in den folgenden Themen.

Themen

- Streaming-Sitzungsdaten mit Amazon CloudWatch Logs (Konsole)
- Protokollieren von Sitzungsdaten mithilfe von Amazon S3 (Konsole)

- Protokollierung von Sitzungsdaten mit Amazon CloudWatch Logs (Konsole)
- Konfigurieren der Sitzungsprotokollierung auf Festplatte
- Anpassen, wie lange die Session Manager temporäre Protokolldatei auf der Festplatte gespeichert wird
- Deaktivierung der Session Manager Protokollierung in CloudWatch Logs und Amazon S3

Streaming-Sitzungsdaten mit Amazon CloudWatch Logs (Konsole)

Sie können einen kontinuierlichen Stream von Sitzungsdatenprotokollen an Amazon CloudWatch Logs senden. Wichtige Details, wie die Befehle, die ein Benutzer in einer Sitzung ausgeführt hat, die ID des Benutzers, der die Befehle ausgeführt hat, und Zeitstempel für den Zeitpunkt, zu dem die Sitzungsdaten in CloudWatch Logs gestreamt werden, sind beim Streaming von Sitzungsdaten enthalten. Beim Streamen von Sitzungsdaten werden die Protokolle JSON-formatiert, um Ihnen bei der Integration in Ihre vorhandenen Protokollierungslösungen zu helfen. Streaming-Sitzungsdaten werden für interaktive Befehle nicht unterstützt.

> **Note**
>
> Um Sitzungsdaten von Windows Server-verwalteten Knoten zu streamen, müssen Sie PowerShell 5.1 oder höher installiert haben. Standardmäßig ist bei Windows Server 2016 und höher die erforderliche PowerShell-Version installiert. Bei Windows Server 2012 und 2012 R2 ist die erforderliche PowerShell-Version jedoch nicht standardmäßig installiert. Wenn Sie PowerShell noch nicht auf Ihren von Windows Server 2012 oder 2012 R2 verwalteten Knoten aktualisiert haben, können Sie dies mit Run Command tun. Informationen zur Aktualisierung von PowerShell mithilfe von Run Command finden Sie unter PowerShell Aktualisierung mit Run Command.

> **Important**
>
> Wenn Sie die Einstellung der PowerShell Transkriptionsrichtlinie auf Ihren Windows Server verwalteten Knoten konfiguriert haben, können Sie keine Sitzungsdaten streamen.

Um Sitzungsdaten mit Amazon CloudWatch Logs zu streamen (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Präferenzen und anschließend Bearbeiten aus.
4. Aktivieren Sie unter CloudWatch Protokollierung das Kontrollkästchen neben Aktivieren.
5. Wählen Sie die Option Sitzungsungsprotokolle streamen aus.
6. (Empfohlen) Aktivieren Sie das Kontrollkästchen neben Nur verschlüsselte CloudWatch Protokollgruppen zulassen. Wenn diese Funktion aktiviert ist, werden die Protokolldaten mithilfe des serverseitigen Verschlüsselungsschlüssels, der für die Protokollgruppe angegeben wurde, verschlüsselt. Wenn Sie die Protokolldaten, die an CloudWatch Logs gesendet werden, nicht verschlüsseln möchten, deaktivieren Sie das Kontrollkästchen. Außerdem müssen Sie das Kontrollkästchen deaktivieren, wenn die Verschlüsselung für die Protokollgruppe nicht aktiviert ist.
7. Wählen Sie für CloudWatch Protokolle eine der folgenden Optionen aus, AWS-Konto um die bestehende CloudWatch Protokollgruppe Logs in die Sie die Sitzungsprotokolle hochladen möchten, anzugeben:

   - Geben Sie den Namen einer bereits in Ihrem Konto erstellten Protokollgruppe in das Textfeld ein, um die Sitzungsprotokolldaten zu speichern.
   - Protokollgruppen durchsuchen: Wählen Sie eine bereits in Ihrem Konto erstellte Protokollgruppe aus, um die Sitzungsprotokolldaten zu speichern.
8. Wählen Sie Speichern.

Protokollieren von Sitzungsdaten mithilfe von Amazon S3 (Konsole)

Sie können Sitzungsprotokolldaten zu Debugging- und Fehlerbehebungszwecken in einem angegebenen Amazon Simple Storage Service (Amazon S3)-Bucket speichern. Standardmäßig werden Protokolle an einen verschlüsselten Amazon S3-Bucket gesendet. Die Verschlüsselung erfolgt mit dem Schlüssel, der für den Bucket angegebenen wurde. Dabei handelt es sich entweder um einen AWS KMS key-Schlüssel oder einen Amazon S3–Schlüssel mit serverseitiger Verschlüsselung (SSE) (AES-256).

> **Important**
>
> Bei Verwendung von Buckets im Stil eines virtuellen Hostings mit SSL (Secure Sockets Layer) stimmt das SSL-Wildcard-Zertifikat nur mit Buckets überein, die keine Punkte enthalten. Um dies zu umgehen, verwenden Sie HTTP oder schreiben Sie Ihre eigene Logik zur Verifizierung von Zertifikaten. Wir empfehlen, bei der Verwendung von Buckets im Stil des virtuellen Hostings keine Punkte („.“) in Bucket-Namen zu verwenden.

Verschlüsselung von Amazon S3-Bucket

Um Protokolle mit Verschlüsselung an Ihren Amazon S3-Bucket senden zu können, muss die Verschlüsselungsfunktion für den Bucket aktiviert sein. Weitere Informationen zur Amazon S3 Bucket-Verschlüsselung finden Sie unter Amazon S3-Standardverschlüsselung für S3-Buckets.

Kundenverwalteter Schlüssel

Wenn Sie zum Verschlüsseln Ihres Buckets einen KMS-Schlüssel verwenden, den Sie selbst verwalten, muss das Ihren Instances angefügte IAM-Instance-Profil explizite Berechtigungen zum Lesen des Schlüssels besitzen. Wenn Sie einen Von AWS verwalteter Schlüssel verwenden, benötigt die Instance diese explizite Berechtigung nicht. Weitere Informationen zum Bereitstellen des Instance-Profils mit Zugriff auf die Verwendung des Schlüssels finden Sie unter Gestattet Schlüsselbenutzern die Verwendung des Schlüssels im AWS Key Management Service-Entwicklerhandbuch.

Gehen Sie wie folgt vor, um Session Manager zum Speichern von Sitzungsprotokollen im Amazon S3-Bucket zu konfigurieren.

> **Note**
>
> Sie können den Amazon S3-Bucket, an den die Sitzungsdaten gesendet werden, auch über die AWS CLI angeben oder ändern. Weitere Informationen finden Sie unter Aktualisieren von Session Manager-Einstellungen (Befehlszeile).

Protokollieren von Sitzungsdaten mithilfe von Amazon S3 (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Präferenzen und anschließend Bearbeiten aus.
4. Wählen Sie bei S3-Protokollierung neben Aktivieren das Kontrollkästchen aus.
5. (Empfohlen) Aktivieren Sie das Kontrollkästchen neben Nur verschlüsselte S3-Buckets zulassen. Wenn diese Funktion aktiviert ist, werden die Protokolldaten mithilfe des serverseitigen Verschlüsselungsschlüssels, der für den Bucket angegeben wurde, verschlüsselt. Wenn Sie die an Amazon S3 gesendeten Protokolldaten nicht verschlüsseln möchten, aktivieren Sie das Kontrollkästchen. Außerdem müssen Sie das Kontrollkästchen deaktivieren, wenn die Verschlüsselung für den S3-Bucket nicht aktiviert ist.
6. Wählen Sie in S3 bucket name (Name des S3-Buckets) eine der folgenden Optionen aus:

> **Note**
>
> Wir empfehlen, bei der Verwendung von Buckets im Stil des virtuellen Hostings keine Punkte („.“) in Bucket-Namen zu verwenden. Weitere Informationen zu Namenskonventionen für Amazon-S3-Buckets finden Sie unter Bucket-Einschränkungen und -Limits im Benutzerhandbuch zu Amazon Simple Storage Service.

   - Choose a bucket name from the list (Bucket-Namen aus der Liste auswählen): Wählen Sie einen bereits in Ihrem Konto erstellten Amazon S3-Bucket aus, um Sitzungsprotokolldaten zu speichern.
   - Enter a bucket name in the text box (Geben Sie einen Namen für den Bucket in das Textfeld ein): Geben Sie den Namen eines bereits in Ihrem Konto erstellten Amazon S3-Buckets ein, um Sitzungsprotokolldaten zu speichern.
7. (Optional) Geben Sie in S3 key prefix (S3-Schlüsselpräfix) den Namen eines vorhandenen oder neuen Ordners ein, in dem die Protokolle im ausgewählten Bucket gespeichert werden sollen.
8. Wählen Sie Speichern.

Weitere Informationen zum Arbeiten mit Amazon S3 und Amazon-S3-Buckets finden Sie im Benutzerhandbuch zu Amazon Simple Storage Service und im Benutzerhandbuch zu Amazon Simple Storage Service.

Protokollierung von Sitzungsdaten mit Amazon CloudWatch Logs (Konsole)

Mit Amazon CloudWatch Logs können Sie Protokolldateien aus verschiedenen Quellen überwachen, speichern und darauf zugreifen AWS-Services. Sie können Sitzungsprotokolldaten zu Debugging- und Fehlerbehebungszwecken an eine CloudWatch Logs-Protokollgruppe senden. Standardmäßig werden Protokolldaten nach Verschlüsselung mit Ihrem KMS-Schlüssel gesendet. Sie können die Daten jedoch mit oder ohne Verschlüsselung an ihre Protokollgruppe senden.

Gehen Sie wie folgt vor, um AWS Systems Manager Session Manager zu konfigurieren, dass Sitzungsprotokolldaten am Ende Ihrer Sitzungen an eine CloudWatch Logs-Protokollgruppe gesendet werden.

> **Note**
>
> Sie können den auch verwenden AWS CLI , um die CloudWatch Logs-Protokollgruppe anzugeben oder zu ändern, an die Sitzungsdaten gesendet werden. Weitere Informationen finden Sie unter Aktualisieren von Session Manager-Einstellungen (Befehlszeile).

So protokollieren Sie Sitzungsdaten mit Amazon CloudWatch Logs (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Präferenzen und anschließend Bearbeiten aus.
4. Aktivieren Sie unter CloudWatch Protokollierung das Kontrollkästchen neben Aktivieren.
5. Wählen Sie die Option Sitzungsungsprotokolle hochladen aus.
6. (Empfohlen) Aktivieren Sie das Kontrollkästchen neben Nur verschlüsselte CloudWatch Protokollgruppen zulassen. Wenn diese Funktion aktiviert ist, werden die Protokolldaten mithilfe des serverseitigen Verschlüsselungsschlüssels, der für die Protokollgruppe angegeben wurde, verschlüsselt. Wenn Sie die Protokolldaten, die an CloudWatch Logs gesendet werden, nicht verschlüsseln möchten, deaktivieren Sie das Kontrollkästchen. Außerdem müssen Sie das Kontrollkästchen deaktivieren, wenn die Verschlüsselung für die Protokollgruppe nicht aktiviert ist.
7. Wählen Sie für CloudWatch Protokolle eine der folgenden Optionen aus, AWS-Konto um die bestehende CloudWatch Protokollgruppe Logs in die Sie die Sitzungsprotokolle hochladen möchten, anzugeben:

- Choose a log group from the list (Eine Protokollgruppe aus der Liste auswählen): Wählen Sie eine bereits in Ihrem Konto erstellte Protokollgruppe aus, in die die Sitzungsprotokolldaten gespeichert werden sollen.
- Enter a log group name in the text box (Geben Sie den Namen einer Protokollgruppe in das Textfeld ein): Geben Sie den Namen einer bereits in Ihrem Konto erstellten Protokollgruppe ein, in die die Sitzungsprotokolldaten gespeichert werden sollen.

8. Wählen Sie Speichern.

Weitere Informationen zur Arbeit mit CloudWatch Logs finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.

Konfigurieren der Sitzungsprotokollierung auf Festplatte

Nachdem Sie die Session Manager Protokollierung bei Amazon S3 CloudWatch oder Amazon S3 aktiviert haben, werden alle während einer Sitzung ausgeführten Befehle (und die daraus resultierenden Ausgaben) in einer temporären Datei auf der Festplatte der Ziel-Instance protokolliert. Die temporäre Datei hat den Namen `ipcTempFile.log`.

`ipcTempFile.log` wird durch den `SessionLogsDestination`-Parameter in der SSM Agent-Konfigurationsdatei gesteuert. Dieser Parameter akzeptiert die folgenden Werte:

- Festplatte: Wenn Sie diesen Parameter angeben und die Sitzungsprotokollierung auf CloudWatch oder Amazon S3 aktiviert ist, SSM Agent erstellt die `ipcTempFile.log` temporäre Protokolldatei und protokolliert Sitzungsbefehle sowie die Ausgabe auf der Festplatte. Session Managerlädt dieses Protokoll je nach Protokollierungskonfiguration während oder nach der Sitzung entweder CloudWatch auf S3 hoch. Das Protokoll wird dann entsprechend der für den SSM Agent `SessionLogsRetentionDurationHours` Konfigurationsparameter angegebenen Dauer gelöscht.

  Wenn Sie diesen Parameter angeben und die Sitzungsprotokollierung bei Amazon S3 deaktiviert ist, werden SSM Agent trotzdem der Befehlsverlauf und die Ausgabe in der `ipcTempFile.log` Datei protokolliert. CloudWatch Die Datei wird entsprechend der für den SSM Agent `SessionLogsRetentionDurationHours` Konfigurationsparameter angegebenen Dauer gelöscht.
- none: Wenn Sie diesen Parameter angeben und die Sitzungsprotokollierung bei CloudWatch oder Amazon S3 aktiviert ist, funktioniert die Protokollierung auf der Festplatte genauso, als ob

Sie den `disk` Parameter angegeben hätten. SSM Agentbenötigt die temporäre Datei, wenn die Sitzungsprotokollierung bei CloudWatch oder Amazon S3 aktiviert ist.

Wenn Sie diesen Parameter angeben und die Sitzungsprotokollierung bei CloudWatch oder Amazon S3 deaktiviert SSM Agent ist, wird die `ipcTempFile.log` Datei nicht erstellt.

Gehen Sie wie folgt vor, um die Erstellung der `ipcTempFile.log`-temporären Protokolldatei auf der Festplatte zu aktivieren oder zu deaktivieren, wenn eine Sitzung gestartet wird.

Um die Erstellung der Session Manager-temporären Protokolldatei auf der Festplatte zu aktivieren oder zu deaktivieren

1. Installieren Sie SSM Agent entweder auf Ihrer Instance oder führen Sie ein Upgrade auf Version 3.2.2086 oder höher durch. Weitere Informationen zum Überprüfen der Agent-Versionsnummer finden Sie unter Überprüfen der SSM Agent-Versionsnummer. Informationen zur manuellen Installation des Agenten finden Sie in den folgenden Abschnitten nach dem Verfahren für Ihr Betriebssystem:

   - Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux
   - Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für macOS
   - Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Windows Server
2. Erstellen Sie eine Verbindung zu Ihrer Instance und suchen Sie die `amazon-ssm-agent.json`-Datei am folgenden Speicherort.

   - Linux:/etc/amazon/ssm/
   - macOS: /opt/aws/ssm/
   - Windows Server: C:\Program Files\Amazon\SSM

   Wenn die Datei `amazon-ssm-agent.json` nicht existiert, kopieren Sie den Inhalt von `amazon-ssm-agent.json.template` in eine neue Datei im selben Verzeichnis. Benennen Sie die Datei `amazon-ssm-agent.json`.
3. Geben Sie entweder `none` oder `disk` für den `SessionLogsDestination`-Parameter an. Speichern Sie Ihre Änderungen.
4. Starten Sie SSM Agent neu.

Wenn Sie `disk` für den `SessionLogsDestination`-Parameter angegeben haben, können Sie überprüfen, ob die temporäre Protokolldatei SSM Agent erstellt wird, indem Sie eine neue Sitzung starten und die `ipcTempFile.log`-Datei dann am folgenden Speicherort suchen:

- Linux://var/lib/amazon/ssm*target ID*/Sitzung/Orchestration/ *session ID* / Standard_Stream/ .log ipcTempFile
- macOS: opt/aws/ssm/data*target ID*/*session ID*/sitzung/orchestration/ ipcTempFile / Standard_Stream/ .log
- Windows Server: C:\\ AmazonProgramData\ SSM\\ SitzungInstanceData\ *target ID* Orchestrierung\\ Standard_Stream*session ID*\ .log ipcTempFile

> **Note**
>
> Standardmäßig wird die temporäre Protokolldatei 14 Tage lang auf der Instance gespeichert.

Wenn Sie den `SessionLogsDestination`-Parameter für mehrere Instances aktualisieren möchten, empfehlen wir Ihnen, ein SSM-Dokument zu erstellen, das die neue Konfiguration spezifiziert. Anschließend können Sie Systems Manager Run Command verwenden, um die Änderung auf Ihren Instances zu implementieren. Weitere Informationen finden Sie unter Eigene AWS Systems Manager Dokumente schreiben (Blog) und. Ausführen von Befehlen auf verwalteten Knoten

Anpassen, wie lange die Session Manager temporäre Protokolldatei auf der Festplatte gespeichert wird

Nachdem Sie die Session Manager Protokollierung bei Amazon S3 CloudWatch oder Amazon S3 aktiviert haben, werden alle während einer Sitzung ausgeführten Befehle (und die daraus resultierenden Ausgaben) in einer temporären Datei auf der Festplatte der Ziel-Instance protokolliert. Die temporäre Datei hat den Namen `ipcTempFile.log`. Session ManagerLädt dieses temporäre Protokoll während einer Sitzung oder nach deren Abschluss entweder CloudWatch auf S3 hoch. Das temporäre Protokoll wird dann entsprechend der für den SSM Agent `SessionLogsRetentionDurationHours` Konfigurationsparameter angegebenen Dauer gelöscht. Standardmäßig wird die temporäre Protokolldatei 14 Tage lang am folgenden Speicherort auf der Instance gespeichert:

- Linux:/var/lib/amazon/ssm*target ID*/session/orchestration/ *session ID* /Standard_Stream/ .log ipcTempFile
- macOS: opt/aws/ssm/data*target ID*/*session ID*/sitzung/orchestration/ ipcTempFile / Standard_Stream/ .log
- Windows Server: C:\\ AmazonProgramData\ SSM\\ SitzungInstanceData\ *target ID* Orchestrierung\\ Standard_Stream*session ID*\ .log ipcTempFile

Gehen Sie wie folgt vor, um einzustellen, wie lange die Session Manager temporäre Protokolldatei auf der Festplatte gespeichert wird.

Um einzustellen, wie lange die **`ipcTempFile.log`**-Datei auf der Festplatte gespeichert wird

1. Erstellen Sie eine Verbindung zu Ihrer Instance und suchen Sie die `amazon-ssm-agent.json`-Datei am folgenden Speicherort.

   - etc/amazon/ssmLinux://
   - macOS: /opt/aws/ssm/
   - Windows Server: C:\Program Files\Amazon\SSM

   Wenn die Datei `amazon-ssm-agent.json` nicht existiert, kopieren Sie den Inhalt von `amazon-ssm-agent.json.template` in eine neue Datei im selben Verzeichnis. Benennen Sie die Datei `amazon-ssm-agent.json`.
2. Ändern Sie den Wert von `SessionLogsRetentionDurationHours` auf die gewünschte Anzahl von Stunden. Wenn `SessionLogsRetentionDurationHours` auf 0 gesetzt ist, wird die temporäre Protokolldatei während der Sitzung erstellt und nach Abschluss der Sitzung gelöscht. Diese Einstellung sollte sicherstellen, dass die Protokolldatei nach dem Ende der Sitzung nicht erhalten bleibt.
3. Speichern Sie Ihre Änderungen.
4. Starten Sie SSM Agent neu.

Deaktivierung der Session Manager Protokollierung in CloudWatch Logs und Amazon S3

Sie können die Systems Manager Manager-Konsole verwenden oder AWS CLI die Sitzungsprotokollierung in Ihrem Konto deaktivieren.

So deaktivieren Sie die Sitzungsprotokollierung (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Session Manager aus.
3. Wählen Sie die Registerkarte Präferenzen und anschließend Bearbeiten aus.
4. Um die CloudWatch Protokollierung zu deaktivieren, deaktivieren CloudWatch Sie im Bereich Protokollierung das Kontrollkästchen Aktivieren.
5. Um die S3-Protokollierung zu deaktivieren, deaktivieren Sie im Abschnitt S3-Protokollierung das Kontrollkästchen Aktivieren.
6. Wählen Sie Speichern.

So können Sie Sitzungsprotokollierung deaktivieren (AWS CLI)

Um die Sitzungsprotokollierung mit dem zu deaktivieren AWS CLI, folgen Sie den Anweisungen unterAktualisieren von Session Manager-Einstellungen (Befehlszeile).

Stellen Sie in Ihrer JSON-Datei sicher, dass die Eingaben `s3BucketName` und `cloudWatchLogGroupName` keine Werte enthalten. Zum Beispiel:

```
"inputs": {
        "s3BucketName": "",
        ...
        "cloudWatchLogGroupName": "",
        ...
    }
```

Alternativ können Sie, um die Protokollierung zu deaktivieren, alle `S3*` und `cloudWatch*` Eingaben aus Ihrer JSON-Datei entfernen, um die Protokollierung zu deaktivieren.

> **Note**
>
> Abhängig von Ihrer Konfiguration kann es sein, dass nach der Deaktivierung CloudWatch von S3 immer noch eine temporäre Protokolldatei auf der Festplatte von generiert wirdSSM Agent. Weitere Informationen, wie Sie die Protokollierung auf der Festplatte deaktivieren können, finden Sie unter Konfigurieren der Sitzungsprotokollierung auf Festplatte.

Schema des Sitzungsdokuments

In den folgenden Informationen werden die Schemaelemente eines Sitzungsdokuments beschrieben. AWS Systems Manager Session Manager verwendet Sitzungsdokumente, um zu bestimmen, welche Art von Sitzung gestartet werden soll, z. B. eine Standardsitzung, eine Port-Weiterleitungssitzung oder eine Sitzung zum Ausführen eines interaktiven Befehls.

schemaVersion

Die Schemaversion des Sitzungsdokuments. Sitzungsdokumente unterstützen nur die Version 1.0.

Typ: Zeichenfolge

Erforderlich: Ja

description

Eine Beschreibung, die Sie für das Sitzungsdokument angeben. Beispiel: „Dokument zum Starten der Port-Weiterleitungssitzung mit Session Manager“.

Typ: Zeichenfolge

Erforderlich: Nein

sessionType

Der Sitzungstyp, mit dem das Sitzungsdokument erstellt wird.

Typ: Zeichenfolge

Erforderlich: Ja

Zulässige Werte: `InteractiveCommands` | `NonInteractiveCommands` | `Port` | `Standard_Stream`

inputs

Die Sitzungseinstellungen, die für Sitzungen verwendet werden, die mit diesem Sitzungsdokument erstellt wurden. Dieses Element ist für Sitzungsdokumente erforderlich, die zum Erstellen von `Standard_Stream`-Sitzungen verwendet werden.

Typ: StringMap

Erforderlich: Nein

s3BucketName

Der Amazon Simple Storage Service (Amazon S3)-Bucket, an den Sie am Ende Ihrer Sitzungen Sitzungsprotokolle senden möchten.

Typ: Zeichenfolge

Erforderlich: Nein

s3KeyPrefix

Das Präfix, das beim Senden von Protokollen an den Amazon S3-Bucket verwendet werden soll, den Sie in der `s3BucketName`-Eingabe angegeben haben. Weitere Hinweise zur Verwendung eines freigegebenen Präfixes für in Amazon S3 gespeicherte Objekte finden Sie unter Wie kann ich Ordner in einem S3-Bucket verwenden? im Benutzerhandbuch zu Amazon Simple Storage Service.

Typ: Zeichenfolge

Erforderlich: Nein

s3EncryptionEnabled

Wenn auf `true` eingestellt ist, muss der Amazon S3-Bucket, den Sie in der `s3BucketName`-Eingabe angegeben haben, verschlüsselt werden.

Typ: Boolesch

Erforderlich: Ja

cloudWatchLogGroupName

Der Name der Amazon CloudWatch Logs (CloudWatch Logs)-Gruppe, an die Sie am Ende Ihrer Sitzungen Sitzungsprotokolle senden möchten.

Typ: Zeichenfolge

Erforderlich: Nein

cloudWatchEncryptionEnabled

Wenn auf `true` eingestellt ist, muss die Protokollgruppe, die Sie in der `cloudWatchLogGroupName`-Eingabe angegeben haben, verschlüsselt werden.

Typ: Boolesch

Erforderlich: Ja

cloudWatchStreamingEnabled

Wenn auf `true` eingestellt ist, wird ein kontinuierlicher Stream von Sitzungsdaten an die Protokollgruppe gesendet, die Sie in der `cloudWatchLogGroupName`-Eingabe angegeben haben. Wenn auf `false` eingestellt ist, werden Sitzungsprotokolle am Ende Ihrer Sitzungen an die Protokollgruppe gesendet, die Sie in der `cloudWatchLogGroupName`-Eingabe angegeben haben.

Typ: Boolesch

Erforderlich: Ja

kmsKeyId

Die ID des AWS KMS key, den Sie verwenden möchten, um Daten zwischen dem lokalen Client-Maschinen und den von Amazon Elastic Compute Cloud (Amazon EC2) verwalteten Knoten, zu denen Sie eine Verbindung herstellen, weiter zu verschlüsseln.

Typ: Zeichenfolge

Erforderlich: Nein

runAsEnabled

Wenn auf `true` eingestellt ist, müssen Sie in der `runAsDefaultUser`-Eingabe ein Benutzerkonto angeben, das auf den verwalteten Knoten vorhanden ist, mit denen Sie eine Verbindung herstellen möchten. Andernfalls können Sitzungen nicht gestartet werden. Standardmäßig werden Sitzungen mit dem von AWS Systems Manager SSM Agent erstellten `ssm-user`-Konto gestartet. Das Run-As-Feature wird nur für die Verbindung mit Linux und macOS-verwalteten Knoten unterstützt.

Typ: Boolesch

Erforderlich: Ja

runAsDefaultUser

Der Name des Benutzerkontos, mit dem Sitzungen auf Linux- und macOS-verwalteten Knoten gestartet werden sollen, wenn die `runAsEnabled`-Eingabe auf `true` eingestellt ist. Das Benutzerkonto, das Sie für diese Eingabe angeben, muss auf den verwalteten Knoten

vorhanden sein, mit denen Sie eine Verbindung herstellen möchten. Andernfalls können Sitzungen nicht gestartet werden.

Typ: Zeichenfolge

Erforderlich: Nein

idleSessionTimeout

Die Zeit der Inaktivität, die Sie zulassen möchten, bevor eine Sitzung beendet wird. Diese Eingabe wird in Minuten gemessen.

Typ: Zeichenfolge

Zulässige Werte: 1 bis 60

Erforderlich: Nein

maxSessionDuration

Die maximale Zeit, die Sie erlauben möchten, bevor eine Sitzung beendet wird. Diese Eingabe wird in Minuten gemessen.

Typ: Zeichenfolge

Zulässige Werte: 1–1 440

Erforderlich: Nein

shellProfile

Die Einstellungen, die Sie pro Betriebssystem angeben und die in Sitzungen angewendet werden, wie Shell-Einstellungen, Umgebungsvariablen, Arbeitsverzeichnissen und das Ausführen mehrerer Befehle.

Typ: StringMap

Erforderlich: Nein

windows

Die Shell-Einstellungen, Umgebungsvariablen, Arbeitsverzeichnisse und Befehle, die Sie für Sitzungen auf Windows Server-verwalteten Knoten angeben.

Typ: Zeichenfolge

Erforderlich: Nein

linux

Die Shell-Einstellungen, Umgebungsvariablen, Arbeitsverzeichnisse und Befehle, die Sie für Sitzungen auf Linux- und macOS-verwalteten Knoten angeben.

Typ: Zeichenfolge

Erforderlich: Nein

parameters

Ein Objekt, das die Parameter definiert, die das Dokument akzeptiert. Weitere Informationen zum Definieren von Dokumentparametern finden Sie unter Parameter im Top-Level-Datenelemente. Wir empfehlen, Parameter, auf die Sie häufig verweisen, im Systems Manager Parameter Store abzuspeichern und dann auf sie zu verweisen. In diesem Abschnitt des Dokuments können Sie die Parameter Store-Parameter `String` und `StringList` referenzieren. In diesem Abschnitt des Dokuments können Sie die Parameter Store-Parameter `SecureString` nicht referenzieren. Sie können über das folgende Format einen Parameter Store-Parameter referenzieren.

```
{{ssm:parameter-name}}
```

Mehr über Parameter Store erfahren Sie unter AWS Systems Manager Parameter Store.

Typ: StringMap

Erforderlich: Nein

properties

Ein Objekt, dessen Werte Sie angeben, die in der `StartSession` API-Operation verwendet werden.

Für Sitzungsdokumente, die für `InteractiveCommands`-Sitzungen verwendet werden,enthält das Eigenschaftenobjekt die Befehle, die auf den von Ihnen angegebenen Betriebssystemen ausgeführt werden sollen. Mit der `runAsElevated` booleschen Eigenschaft können Sie auch festlegen, ob Befehle als `root` ausgeführt werden. Weitere Informationen finden Sie unter Zugriff auf Befehle in einer Sitzung beschränken.

Für Sitzungsdokumente, die für `Port`-Sitzungen verwendet werden, enthält das Eigenschaftenobjekt die Portnummer, an die der Datenverkehr umgeleitet werden soll. Ein

Beispiel ist das Beispiel zum Sitzungsdokument des Typs `Port` an späterer Stelle in diesem Thema.

Typ: StringMap

Erforderlich: Nein

Beispiel für Sitzungsdokument vom Typ `Standard_Stream`

YAML

```
---
schemaVersion: '1.0'
description: Document to hold regional settings for Session Manager
sessionType: Standard_Stream
inputs:
  s3BucketName: ''
  s3KeyPrefix: ''
  s3EncryptionEnabled: true
  cloudWatchLogGroupName: ''
  cloudWatchEncryptionEnabled: true
  cloudWatchStreamingEnabled: true
  kmsKeyId: ''
  runAsEnabled: true
  runAsDefaultUser: ''
  idleSessionTimeout: '20'
  maxSessionDuration: '60'
  shellProfile:
    windows: ''
    linux: ''
```

JSON

```
{
    "schemaVersion": "1.0",
    "description": "Document to hold regional settings for Session Manager",
    "sessionType": "Standard_Stream",
    "inputs": {
        "s3BucketName": "",
        "s3KeyPrefix": "",
        "s3EncryptionEnabled": true,
        "cloudWatchLogGroupName": "",
```

```
        "cloudWatchEncryptionEnabled": true,
        "cloudWatchStreamingEnabled": true,
        "kmsKeyId": "",
        "runAsEnabled": true,
        "runAsDefaultUser": "",
        "idleSessionTimeout": "20",
        "maxSessionDuration": "60",
        "shellProfile": {
            "windows": "date",
            "linux": "pwd;ls"
        }
    }
}
```

Beispiel für Sitzungsdokument vom Typ `InteractiveCommands`

YAML

```
---
schemaVersion: '1.0'
description: Document to view a log file on a Linux instance
sessionType: InteractiveCommands
parameters:
  logpath:
    type: String
    description: The log file path to read.
    default: "/var/log/amazon/ssm/amazon-ssm-agent.log"
    allowedPattern: "^[a-zA-Z0-9-_/]+(.log)$"
properties:
  linux:
    commands: "tail -f {{ logpath }}"
    runAsElevated: true
```

JSON

```
{
    "schemaVersion": "1.0",
    "description": "Document to view a log file on a Linux instance",
    "sessionType": "InteractiveCommands",
    "parameters": {
        "logpath": {
            "type": "String",
```

```
            "description": "The log file path to read.",
            "default": "/var/log/amazon/ssm/amazon-ssm-agent.log",
            "allowedPattern": "^[a-zA-Z0-9-_/]+(.log)$"
        }
    },
    "properties": {
        "linux": {
            "commands": "tail -f {{ logpath }}",
            "runAsElevated": true
        }
    }
}
```

Beispiel für Sitzungsdokument vom Typ `Port`

YAML

```
---
schemaVersion: '1.0'
description: Document to open given port connection over Session Manager
sessionType: Port
parameters:
  paramExample:
    type: string
    description: document parameter
properties:
  portNumber: anyPortNumber
```

JSON

```
{
    "schemaVersion": "1.0",
    "description": "Document to open given port connection over Session Manager",
    "sessionType": "Port",
    "parameters": {
        "paramExample": {
            "type": "string",
            "description": "document parameter"
        }
    },
    "properties": {
        "portNumber": "anyPortNumber"
```

```
    }
}
```

Beispiel für Sitzungsdokument mit Sonderzeichen

YAML

```
---
schemaVersion: '1.0'
description: Example document with quotation marks
sessionType: InteractiveCommands
parameters:
  Test:
    type: String
    description: Test Input
    maxChars: 32
properties:
  windows:
    commands: |
        $Test = '{{ Test }}'
        $myVariable = \"Computer name is $env:COMPUTERNAME\"
        Write-Host "Test variable: $myVariable`.`nInput parameter: $Test"
    runAsElevated: false
```

JSON

```
{
   "schemaVersion":"1.0",
   "description":"Test document with quotation marks",
   "sessionType":"InteractiveCommands",
   "parameters":{
      "Test":{
         "type":"String",
         "description":"Test Input",
         "maxChars":32
      }
   },
   "properties":{
      "windows":{
         "commands":[
            "$Test = '{{ Test }}'",
            "$myVariable = \\\"Computer name is $env:COMPUTERNAME\\\"",
```

```
            "Write-Host \"Test variable: $myVariable`.`nInput parameter: $Test\""
         ],
         "runAsElevated":false
      }
   }
}
```

Fehlerbehebung für Session Manager

Im Folgenden finden Sie Informationen zur Behandlung von Problemen mit AWS Systems Manager Session Manager.

Themen

- AccessDeniedException beim Aufrufen der TerminateSession Operation
- Der Dokumentvorgang ist unerwartet fehlgeschlagen: Zeitlimit für den Dokumentenarbeiter überschritten
- Session Managerkann von der EC2 Amazon-Konsole aus keine Verbindung herstellen
- Keine Berechtigung zum Starten einer Sitzung
- SSM Agent nicht online
- Keine Berechtigung zum Ändern von Sitzungspräferenzen
- Verwalteter Knoten ist nicht verfügbar oder nicht für Session Manager konfiguriert
- Session Manager-Plugin nicht gefunden
- Session Manager-Plugin wurde nicht automatisch zumBefehlszeilenpfad hinzugefügt (Windows)
- Session Manager-Plugin reagiert nicht mehr
- TargetNotConnected
- Nach dem Starten einer Sitzung wird ein leerer Bildschirm angezeigt
- Verwalteter Knoten reagiert während langer Sitzungen nicht mehr
- Beim Aufrufen des Vorgangs ist ein Fehler aufgetreten (InvalidDocument) StartSession

AccessDeniedException beim Aufrufen der TerminateSession Operation

Problem: Beim versuchten Beenden einer Sitzung gibt Systems Manager den folgenden Fehler zurück:

```
An error occurred (AccessDeniedException) when calling the TerminateSession operation:
```

```
User: <user_arn> is not authorized to perform: ssm:TerminateSession on resource:
<ssm_session_arn> because no identity-based policy allows the ssm:TerminateSession
 action.
```

Lösung A: Stellen Sie sicher, dass die neueste Version des Session Manager-Plug-ins auf dem Knoten installiert ist

Geben Sie im Terminal den folgenden Befehl ein und drücken Sie anschließend die Eingabetaste.

```
session-manager-plugin --version
```

Lösung B: Installieren oder reinstallieren Sie die neueste Version des Plugins

Weitere Informationen finden Sie unter Installieren des Session Manager-Plugin für die AWS CLI.

Lösung C: Versuchen Sie, eine Verbindung zum Knoten wiederherzustellen

Stellen Sie sicher, dass der Knoten auf Anfragen reagiert. Versuchen Sie, die Sitzung wiederherzustellen. Oder öffnen Sie bei Bedarf die EC2 Amazon-Konsole und überprüfen Sie, ob die Instance ausgeführt wird.

Der Dokumentvorgang ist unerwartet fehlgeschlagen: Zeitlimit für den Dokumentenarbeiter überschritten

Problem: Beim Starten einer Sitzung auf einem Linux-Host gibt Systems Manager den folgenden Fehler zurück:

```
document process failed unexpectedly: document worker timed out,
check [ssm-document-worker]/[ssm-session-worker] log for crash reason
```

Wenn Sie die SSM Agent-Protokollierung wie unter Anzeigen von SSM Agent-Protokollen beschrieben konfiguriert haben, können Sie weitere Details im Debugging-Protokoll einsehen. Für dieses Problem zeigt Session Manager den folgende Protokolleintrag an:

```
failed to create channel: too many open files
```

Dieser Fehler weist in der Regel darauf hin, dass zu viele Session Manager-Arbeitsprozesse ausgeführt werden und das zugrunde liegende Betriebssystem ein Limit erreicht hat. Sie haben zwei Möglichkeiten, dieses Problem zu lösen.

Lösung A: Das Limit für Dateibenachrichtigungen im Betriebssystem erhöhen

Sie können das Limit erhöhen, indem Sie den folgenden Befehl von einem separaten Linux-Host aus ausführen. Dieser Befehl verwendet Systems Manager Run Command. Der angegebene Wert erhöht sich `max_user_instances` auf 8 192. Dieser Wert ist erheblich höher als der Standardwert 128, belastet aber die Hostressourcen nicht:

```
aws ssm send-command --document-name AWS-RunShellScript \
--instance-id i-02573cafcfEXAMPLE  --parameters \
"commands=sudo sysctl fs.inotify.max_user_instances=8192"
```

Lösung B: Die von Session Manager im Zielhost verwendeten Dateibenachrichtigungen reduzieren

Führen Sie den folgenden Befehl von einem separaten Linux-Host aus, um die auf dem Zielhost laufenden Sitzungen aufzulisten:

```
aws ssm describe-sessions --state Active --filters key=Target,value=i-02573cafcfEXAMPLE
```

Überprüfen Sie die Befehlsausgabe, um nicht mehr benötigte Sitzungen zu identifizieren. Sie können diese Sitzung beenden, indem Sie den folgenden Befehl von einem separaten Linux-Host ausführen:

```
aws ssm terminate-session –session-id session ID
```

Wenn auf dem Remoteserver keine weiteren Sitzungen mehr laufen, können Sie optional zusätzliche Ressourcen freigeben, indem Sie den folgenden Befehl von einem separaten Linux-Host aus ausführen. Dieser Befehl beendet alle Session Manager-Prozesse, die auf dem Remote-Host ausgeführt werden, und folglich alle Sitzungen auf dem Remote-Host. Bevor Sie diesen Befehl ausführen, stellen Sie sicher, dass keine laufenden Sitzungen vorhanden sind, die Sie behalten möchten:

```
aws ssm send-command --document-name AWS-RunShellScript \
            --instance-id i-02573cafcfEXAMPLE --parameters \
'{"commands":["sudo kill $(ps aux | grep ssm-session-worker | grep -v grep | awk
 '"'"'{print $2}'"'"')"]}'
```

Session Managerkann von der EC2 Amazon-Konsole aus keine Verbindung herstellen

Problem: Nachdem Sie eine neue Instance erstellt haben, haben Sie in der Amazon Elastic Compute Cloud (Amazon EC2) -Konsole über die Schaltfläche Connect > Registerkarte Session Manager keine Möglichkeit, eine Verbindung herzustellen.

Lösung A: Erstellen Sie ein Instance-Profil: Falls Sie dies noch nicht getan haben (wie in den Informationen auf der Registerkarte Session Manager in der EC2 Konsole beschrieben), erstellen Sie ein AWS Identity and Access Management (IAM-) Instance-Profil mithilfe von. Quick Setup Quick Setupist ein Tool in. AWS Systems Manager

Session Manager erfordert ein IAM-Instance-Profil, um eine Verbindung zu Ihrer Instance herzustellen. Sie können ein Instance-Profil erstellen und es Ihrer Instance zuweisen, indem Sie eine Host-Verwaltungs-Konfiguration mit Quick Setup erstellen. Eine Host-Verwaltungs-Konfiguration erstellt ein Instance-Profil mit den erforderlichen Berechtigungen und weist es Ihrer Instance zu. Eine Host-Verwaltungs-Konfiguration ermöglicht auch andere Systems-Manager-Tools und erstellt IAM-Rollen für die Ausführung dieser Tools. Die Nutzung von Quick Setup oder der Tools, die durch die Host-Verwaltungs-Konfiguration aktiviert werden, ist kostenlos. Öffnen Sie Quick Setup und erstellen Sie eine Host-Verwaltungs-Konfiguration.

> **Important**
>
> Nachdem Sie die Host-Management-Konfiguration erstellt haben, EC2 kann es einige Minuten dauern, bis Amazon die Änderung registriert und die Registerkarte „Session Manager“ aktualisiert hat. Wenn auf der Registerkarte nach zwei Minuten keine Schaltfläche Verbinden angezeigt wird, starten Sie Ihre Instance neu. Wenn nach dem Neustart immer noch keine Verbindungsoption angezeigt wird, öffnen Sie Quick Setup und überprüfen Sie, dass Sie nur eine Hostverwaltungskonfiguration haben. Wenn es zwei gibt, löschen Sie die ältere Konfiguration und warten Sie einige Minuten.

Wenn Sie nach dem Erstellen einer Host-Verwaltungs-Konfiguration immer noch keine Verbindung herstellen können oder wenn Sie eine Fehlermeldung erhalten, einschließlich einer Fehlermeldung zu SSM Agent, nutzen Sie eine der folgenden Lösungen:

- Lösung B: Kein Fehler, aber es kann immer noch keine Verbindung hergestellt werden
- Lösung C: Fehler wegen fehlendem SSM Agent

Lösung B: Kein Fehler, aber es kann immer noch keine Verbindung hergestellt werden

Wenn Sie die Host-Verwaltungs-Konfiguration erstellt haben, mehrere Minuten gewartet haben, bevor Sie versucht haben, eine Verbindung herzustellen, und immer noch keine Verbindung herstellen können, müssen Sie die Host-Management-Konfiguratio möglicherweise manuell auf Ihre Instance

anwenden. Gehen Sie wie folgt vor, um eine Quick Setup Host-Verwaltungs-Konfiguration zu aktualisieren und Änderungen auf eine Instance anzuwenden.

So aktualisieren Sie eine Host-Verwaltungs-Konfiguration mit Quick Setup

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie in der Konfigurationsliste die Host-Verwaltungs-Konfiguration aus, die Sie erstellt haben.
4. Wählen Sie Aktionen und wählen Sie dann Konfiguration bearbeiten.
5. Wählen Sie unten im Bereich Ziele unter Wählen Sie aus, wie Sie auf Instances abzielen möchten die Option Manuell aus.
6. Wählen Sie im Abschnitt Instances die Instance aus, die Sie erstellt haben.
7. Wählen Sie Aktualisieren aus.

Warten Sie einige Minuten EC2 , bis die Registerkarte Session Manager aktualisiert ist. Wenn Sie immer noch keine Verbindung herstellen können oder eine Fehlermeldung angezeigt wird, überprüfen Sie die verbleibenden Lösungen für dieses Problem.

Lösung C: Fehler wegen fehlendem SSM Agent

Wenn Sie mithilfe von keine Host-Management-Konfiguration erstellen konnten oder wenn Sie die Fehlermeldung erhalten habenQuick Setup, dass Sie SSM Agent nicht installiert wurden, müssen Sie die Installation möglicherweise manuell SSM Agent auf Ihrer Instance durchführen. SSM Agentist eine Amazon-Software, mit der Systems Manager mithilfe von eine Verbindung zu Ihrer Instance herstellen kannSession Manager. SSM Agentist standardmäßig auf den meisten Amazon Machine Images (AMIs) installiert. Wenn Ihre Instance aus einem nicht standardmäßigen AMI oder einem älteren AMI erstellt wurde, müssen Sie den Agenten möglicherweise manuell installieren. Informationen zum Installationsverfahren von SSM Agent finden Sie im folgenden Thema, das Ihrem Instance-Betriebssystem entspricht.

- Windows Server
- macOS
- AlmaLinux
- Amazon Linux 2 und AL2 023

- Debian Server
- Oracle Linux
- Red Hat Enterprise Linux
- Rocky Linux
- Ubuntu Server

Bei Problemen mit SSM Agent siehe Fehlerbehebung für SSM Agent.

Keine Berechtigung zum Starten einer Sitzung

Problem: Sie versuchen, eine Sitzung zu starten. Das System teilt Ihnen jedoch mit, dass Sie nicht über die erforderlichen Berechtigungen verfügen.

- Lösung: Ein Systemadministrator hat Ihnen keine AWS Identity and Access Management (IAM-) Richtlinienberechtigungen zum Starten Session Manager von Sitzungen erteilt. Weitere Informationen finden Sie unter Kontrollieren des Sitzungszugriffs von Benutzern auf Instances.

SSM Agent nicht online

Problem: Auf der Session ManagerRegisterkarte EC2 Amazon-Instance wird eine Meldung angezeigt, die besagt: "SSM Agentist nicht online. SSM Agent konnte keine Verbindung zu einem Systems-Manager-Endpunkt herstellen, um sich beim Service zu registrieren.“

Lösung: SSM Agent ist Amazon-Software, die auf EC2 Amazon-Instances ausgeführt wird, sodass eine Verbindung zu ihnen hergestellt werden Session Manager kann. Wenn Sie diesen Fehler sehen, kann SSM Agent keine Verbindung mit dem Systems-Manager-Endpunkt herstellen. Die Ursache des Problems könnten Firewall-Einschränkungen, Routing-Probleme oder mangelnde Internetverbindung sein. Sie lösen dieses Problem, indem Sie die Probleme mit der Netzwerkverbindung untersuchen. Weitere Informationen erhalten Sie unter Fehlerbehebung für SSM Agent und Problembehandlung bei der Verfügbarkeit verwalteter Knoten. Informationen zu Endpunkten von Systems Manager finden Sie unter AWS Systems Manager -Endpunkte und Kontingente in der Allgemeinen AWS -Referenz.

Keine Berechtigung zum Ändern von Sitzungspräferenzen

Problem: Sie versuchen, globale Sitzungspräferenzen für Ihre Organisation zu aktualisieren. Das System teilt Ihnen jedoch mit, dass Sie nicht über die erforderlichen Berechtigungen verfügen.

- Lösung: Ihnen wurden vom Systemadministrator keine IAM-Richtlinienberechtigungen zum Festlegen von Session Manager-Präferenzen erteilt. Weitere Informationen finden Sie unter Gewähren oder Verweigern von Benutzerberechtigungen zum Aktualisieren von Session Manager-Einstellungen.

Verwalteter Knoten ist nicht verfügbar oder nicht für Session Manager konfiguriert

Problem 1: Sie möchten auf der Seite Start a session (Sitzung starten) der Konsole eine Sitzung starten. Es befindet sich jedoch kein verwalteter Knoten in der Liste.

- Lösung A: Der verwaltete Knoten, zu dem Sie eine Verbindung herstellen möchten, wurde möglicherweise nicht konfiguriert AWS Systems Manager. Weitere Informationen finden Sie unter Einrichten der einheitlichen Systems-Manager-Konsole für eine Organisation.

  > **Note**
  >
  > Wenn er bereits auf einem verwalteten Knoten ausgeführt AWS Systems Manager SSM Agent wird, wenn Sie das IAM-Instanzprofil anhängen, müssen Sie den Agenten möglicherweise neu starten, bevor die Instanz auf der Seite Sitzungskonsole starten aufgeführt wird.

- Lösung B: Die Proxy-Konfiguration, die Sie auf den SSM Agent auf Ihrem verwalteten Knoten angewendet haben, ist möglicherweise falsch. Wenn die Proxy-Konfiguration falsch ist, kann der verwaltete Knoten die erforderlichen Service-Endpunkte nicht erreichen, oder der Knoten meldet sich möglicherweise als anderes Betriebssystem beim Systems Manager. Weitere Informationen erhalten Sie unter Konfigurieren Sie SSM Agent, um einen Proxy in Linux-Knoten zu verwenden und Konfigurieren des SSM Agent zur Nutzung eines Proxys für Windows Server-Instances.

Problem 2: Ein verwalteter Knoten, mit dem Sie eine Verbindung herstellen möchten, befindet sich in der Liste auf der Seite Start a session (Sitzung starten) der Konsole. Die Seite meldet jedoch, dass „die von Ihnen ausgewählte Instance nicht für die Verwendung mit Session Manager konfiguriert wurde.“

- Lösung A: Der verwaltete Knoten wurde für die Verwendung mit dem Systems-Manager-Service konfiguriert. Das dem Knoten angefügte IAM-Instance-Profil enthält jedoch möglicherweise keine Berechtigungen für das Session Manager-Tool. Informationen hierzu finden Sie unter Überprüfen oder Erstellen eines IAM-Instance-Profils mit Session Manager-Berechtigungen.

- Lösung B: Der verwaltete Knoten wird nicht auf einer Version von SSM Agent ausgeführt, die den Session Manager unterstützt. Aktualisieren Sie SSM Agent auf dem Knoten auf die Version 2.3.68.0 oder höher.

  Sie können SSM Agent manuell auf einem verwalteten Knoten aktualisieren, indem Sie die in Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Windows Server, Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux oder Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für macOS beschriebenen Schritte ausführen, abhängig vom Betriebssystem.

  Alternativ können Sie das Run Command-Dokument `AWS-UpdateSSMAgent` verwenden, um die Agent-Version auf einem oder mehreren verwalteten Knoten gleichzeitig zu aktualisieren. Weitere Informationen finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command.

  > **Tip**
  >
  > Damit Ihr Agent stets auf dem aktuellen Stand ist, sollten Sie die Aktualisierung des SSM Agent auf die jeweils aktuelle Version anhand eines automatisierten Zeitplans ausführen, den Sie mittels einer der beiden folgenden Methoden definieren:
  >
  > - Führen Sie `AWS-UpdateSSMAgent` als Teil einer State Manager-Zuordnung aus. Weitere Informationen finden Sie unter Exemplarische Vorgehensweise: Automatisches Update SSM Agent mit AWS CLI.
  > - Führen Sie `AWS-UpdateSSMAgent` als Teil eines Wartungsfensters aus. Weitere Informationen zum Arbeiten mit Wartungsfenstern finden Sie unter Wartungsfenster mit der Konsole erstellen und verwalten und Tutorial: Erstellen und konfigurieren Sie ein Wartungsfenster mit dem AWS CLI.

- Lösung C: Der verwaltete Knoten kann die erforderlichen Service-Endpunkte nicht erreichen. Sie können die Sicherheitslage Ihrer verwalteten Knoten verbessern, indem Sie Schnittstellenendpunkte verwenden, die mit Strom versorgt werden, AWS PrivateLink um eine Verbindung zu Systems Manager Manager-Endpunkten herzustellen. Die Alternative zur Verwendung von Schnittstellenendpunkten ist das Erlauben von ausgehendem Internetzugriff auf Ihre verwalteten Knoten. Weitere Informationen finden Sie unter Verwenden PrivateLink zum Einrichten eines VPC-Endpunkts für Session Manager.
- Lösung D: Der verwaltete Knoten verfügt über begrenzte verfügbare CPU- oder Speicherressourcen. Auch wenn Ihr verwalteter Knoten ansonsten funktionsfähig ist, können Sie

keine Sitzung einrichten, wenn der Knoten nicht über genügend verfügbare Ressourcen verfügt. Weitere Informationen finden Sie unter Problembehandlung bei unerreichbaren Instances.

Session Manager-Plugin nicht gefunden

Um die Befehle AWS CLI to run session verwenden zu können, muss das Session Manager Plugin auch auf Ihrem lokalen Computer installiert sein. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugin für die AWS CLI.

Session Manager-Plugin wurde nicht automatisch zumBefehlszeilenpfad hinzugefügt (Windows)

Wenn Sie das Session Manager-Plugin auf Windows installieren, sollte die ausführbare Datei `session-manager-plugin` der Umgebungsvariablen `PATH` Ihres Betriebssystems automatisch hinzugefügt werden. Wenn die Prüfung auf korrekte Installation des Session Manager-Plugins (`aws ssm start-session --target` *`instance-id`*) ergibt, dass der Befehl fehlgeschlagen ist, müssen Sie das Plugin möglicherweise mittels des folgenden Verfahrens manuell einrichten.

So ändern Sie die Variable PATH (Windows)

1. Betätigen Sie die Windows-Taste und geben Sie **`environment variables`** ein.
2. Wählen Sie Edit environment variables for your account (Umgebungsvariablen für Ihr Konto bearbeiten).
3. Wählen Sie PATH und anschließend Edit.
4. Fügen Sie dem Feld Variable value (Variablenwert) Pfade hinzu, die durch Semikolons getrennt sind, wie in diesem Beispiel dargestellt: *`C:\existing\path`*;**`C:\new\path`**

   `C:\existing\path` stellt den Wert dar, der sich bereits im Feld befindet. **`C:\new\path`** stellt den Pfad dar, den Sie hinzufügen möchten, wie im folgenden Beispielen dargestellt.

   - 64-Bit-Computer: `C:\Program Files\Amazon\SessionManagerPlugin\bin\`
5. Klicken Sie zweimal auf OK, um die neuen Einstellungen anzuwenden.
6. Schließen Sie alle etwa ausgeführten Eingabeaufforderungen und öffnen Sie erneut.

Session Manager-Plugin reagiert nicht mehr

Während einer Port-Weiterleitungssitzung kann der Datenverkehr nicht mehr weitergeleitet werden, wenn auf Ihrem lokalen Computer Antivirus-Software installiert ist. In einigen Fällen stört Antivirus-

Software das Session Manager-Plugin, was zu Prozess-Deadlocks führt. Um dieses Problem zu beheben, erlauben Sie das Session Manager-Plugin in der Antivirus-Software oder schließen Sie es aus. Weitere Informationen zum Standardinstallationspfad für das Session Manager-Plugin finden Sie unter Installieren des Session Manager-Plugin für die AWS CLI.

TargetNotConnected

Problem: Sie versuchen, eine Sitzung zu starten, aber das System gibt die Fehlermeldung „Beim Aufrufen der StartSession Operation ist ein Fehler aufgetreten (TargetNotConnected): *InstanceID* ist nicht verbunden.“ zurück.

- Lösung A: Dieser Fehler wird zurückgegeben, wenn der angegebene verwaltete Knoten für die Sitzung nicht vollständig für die Verwendung mit dem Session Manager konfiguriert wurde. Weitere Informationen finden Sie unter Einrichten von Session Manager.
- Lösung B: Dieser Fehler wird auch zurückgegeben, wenn Sie versuchen, eine Sitzung auf einem verwalteten Knoten zu starten, der sich in einem anderen AWS-Konto oder befindet AWS-Region.

Nach dem Starten einer Sitzung wird ein leerer Bildschirm angezeigt

Problem: Sie starten eine Sitzung und Session Manager zeigt einen leeren Bildschirm an.

- Lösung A: Dieses Problem kann auftreten, wenn das Root-Volume des verwalteten Knotens voll ist. Aufgrund von mangelndem Speicherplatz funktioniert SSM Agent auf dem Knoten nicht mehr. Um dieses Problem zu lösen, verwenden Sie Amazon, CloudWatch um Metriken und Protokolle von den Betriebssystemen zu sammeln. Weitere Informationen finden Sie unter Erfassung von Metriken, Protokollen und Traces mit dem CloudWatch Agenten im CloudWatch Amazon-Benutzerhandbuch.
- Lösung B: Möglicherweise wird ein leerer Bildschirm angezeigt, wenn Sie über einen Link auf die Konsole zugegriffen haben, der ein nicht übereinstimmendes Endpunkt- und Regionspaar enthält. Beispielsweise ist in der folgenden Konsolen-URL `us-west-2` der angegebene Endpunkt, aber `us-west-1` die angegebene AWS-Region.

  ```
  https://us-west-2.console.aws.amazon.com/systems-manager/session-manager/sessions?
  region=us-west-1
  ```

- Lösung C: Der verwaltete Knoten stellt über VPC-Endpunkte eine Verbindung zu Systems Manager her, und Ihre Session Manager Einstellungen schreiben die Sitzungsausgabe in einen Amazon S3 S3-Bucket oder eine Amazon CloudWatch Logs-Protokollgruppe, aber ein `s3` Gateway-

Endpunkt oder `logs` Schnittstellenendpunkt ist in der VPC nicht vorhanden. Ein `s3`-Endpunkt im Format **`com.amazonaws.`*`region`*`.s3`** ist erforderlich, wenn Ihre verwalteten Knoten eine Verbindung zu Systems Manager mithilfe von VPC-Endpunkten herstellen und Ihre Session Manager-Einstellungen die Sitzungsausgabe in einen Amazon-S3-Bucket schreiben. Alternativ **`com.amazonaws.`*`region`*`.logs`**ist ein `logs` Endpunkt in diesem Format erforderlich, wenn Ihre verwalteten Knoten über VPC-Endpunkte eine Verbindung zu Systems Manager herstellen und Ihre Session Manager Einstellungen die Sitzungsausgabe in eine CloudWatch Logs-Protokollgruppe schreiben. Weitere Informationen finden Sie unter Erstellen von VPC-Endpunkten für Systems Manager.

- Lösung D: Die Protokollgruppe oder der Amazon S3-Bucket, die/den Sie in Ihren Sitzungseinstellungen angegeben haben, wurde gelöscht. Aktualisieren Sie Ihre Sitzungseinstellungen mit einer gültigen Protokollgruppe oder einem gültigen S3-Bucket, um dieses Problem zu beheben.
- Lösung E: Die Protokollgruppe oder der Amazon S3-Bucket, die/den Sie in Ihren Sitzungseinstellungen angegeben haben, ist nicht verschlüsselt, aber Sie haben die `cloudWatchEncryptionEnabled`- oder `s3EncryptionEnabled`-Eingabe auf `true` eingestellt. Um dieses Problem zu beheben, aktualisieren Sie Ihre Sitzungseinstellungen mit einer Protokollgruppe oder einem Amazon S3-Bucket, die/der verschlüsselt ist, oder stellen Sie die `cloudWatchEncryptionEnabled`- oder `s3EncryptionEnabled`-Eingabe auf `false` ein. Dieses Szenario gilt nur für Kunden, die Sitzungseinstellungen mithilfe von Befehlszeilentools erstellen.

Verwalteter Knoten reagiert während langer Sitzungen nicht mehr

Problem: Ihr verwalteter Knoten reagiert nicht oder stürzt während einer langen Sitzung ab.

Lösung: Verringern Sie die SSM Agent-Protokollaufbewahrungsdauer für Session Manager.

So verringern Sie die SSM Agent-Protokollaufbewahrungsdauer für Sitzungen

1. Suchen Sie `amazon-ssm-agent.json.template` im `/etc/amazon/ssm/`-Verzeichnis für Linux oder `C:\Program Files\Amazon\SSM` für Windows.
2. Kopieren Sie den Inhalt von `amazon-ssm-agent.json.template` in eine neue Datei im selben Verzeichnis namens `amazon-ssm-agent.json`.
3. Verringern Sie den Standardwert des `SessionLogsRetentionDurationHours`-Werts in der `SSM`-Eigenschaft und speichern Sie die Datei.
4. Starten Sie die SSM Agent neu.

Beim Aufrufen des Vorgangs ist ein Fehler aufgetreten (InvalidDocument) StartSession

Problem: Sie erhalten den folgenden Fehler, wenn Sie eine Sitzung mit dem AWS CLI starten.

```
An error occurred (InvalidDocument) when calling the StartSession operation: Document
 type: 'Command' is not supported. Only type: 'Session' is supported for Session
 Manager.
```

Lösung: Das SSM-Dokument, das Sie für den `--document-name`-Parameter angegeben haben, ist kein Sitzungsdokument. Gehen Sie wie folgt vor, um eine Liste von Sitzungsdokumenten in der AWS-Managementkonsole anzuzeigen.

So rufen Sie eine Liste von Sitzungsdokumenten auf

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie in der Liste Kategorien die Option Sitzungsdokumente aus.

AWS Systems Manager State Manager

State Manager, ein Tool in AWS Systems Manager, ist ein sicherer und skalierbarer Konfigurationsmanagement-Service, der den Prozess automatisiert, Ihre verwalteten Knoten und andere AWS Ressourcen in einem von Ihnen definierten Zustand zu halten. Um mit State Manager zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich State Manager aus.

> **Note**
>
> State Manager und Maintenance Windows können ähnliche Arten von Updates für Ihre verwalteten Knoten ausführen. Welche Option Sie wählen, hängt davon ab, ob Sie die System-Compliance automatisieren oder zeitkritische Aufgaben mit hoher Priorität während der von Ihnen angegebenen Zeiträume ausführen müssen.
> Weitere Informationen finden Sie unter Auswahl zwischen State Manager und Maintenance Windows.

Welche Vorteile bietet State Manager meiner Organisation?

Durch die Verwendung vorkonfigurierter Systems-Manager-Dokumente (SSM-Dokumente), bietet State Manager die folgenden Vorteile für die Verwaltung Ihrer Knoten:

- Bootstrap von Knoten mit bestimmter Software beim Startup.
- Herunterladen und Aktualisieren von Agenten nach einem definierten Zeitplan, einschließlich des SSM Agent.
- Konfigurieren von Netzwerkeinstellungen.
- Verbinden Sie Knoten mit einer Microsoft-Active-Directory-Domain.
- Ausführen von Skripts auf Linux-, macOS- und Windows Server-verwalteten Knoten während des gesamten Lebenszyklus.

Um Konfigurationsabweichungen zwischen anderen AWS Ressourcen zu verwalten, können Sie Automation, ein Tool in Systems Manager, verwenden, mit State Manager dem Sie die folgenden Arten von Aufgaben ausführen können:

- Ordnen Sie Amazon Elastic Compute Cloud (Amazon EC2) -Instances eine Systems Manager Manager-Rolle zu, um sie zu verwalteten Knoten zu machen.
- Erzwingen Sie die gewünschten Eingangs- und Ausgangsregeln für eine Sicherheitsgruppe.
- Erstellen oder löschen Sie Amazon DynamoDB-Backups.
- Erstellen oder löschen Sie Amazon Elastic Block Store (Amazon EBS)-Snapshots.
- Deaktivieren Sie Lese- und Schreibberechtigungen für Amazon Simple Storage Service (Amazon S3)-Buckets.
- Starten, Stoppen oder starten Sie verwaltete Knoten und Amazon Relational Database Service (Amazon RDS)-Instances neu.
- Anwenden von Patches auf Linux, macOS und Windows AMIs.

Weitere Informationen zur Verwendung von State Manager mit Automation-Runbooks finden Sie unter Planen von Automatisierungen mit State Manager-Zuordnungen.

An wen richtet sich State Manager?

State Managerist für jeden AWS Kunden geeignet, der die Verwaltung und Steuerung seiner AWS Ressourcen verbessern und Konfigurationsabweichungen reduzieren möchte.

Über welche Features verfügt State Manager?

Nachstehend sind einige der wichtigsten Features von State Manager aufgelistet:

- State Manager-Zuordnungen

  Eine State Manager Zuordnung ist eine Konfiguration, die Sie Ihren AWS Ressourcen zuweisen. Die Konfiguration definiert den Status, den Sie auf Ihren Ressourcen beibehalten möchten. Beispiel: Eine Zuordnung kann angeben, dass auf einem verwalteten Knoten Antiviren-Software installiert sein und ausgeführt werden muss oder dass bestimmte Ports geschlossen sein müssen.

  Eine Assoziation gibt einen Zeitplan an, wann die Konfiguration und die Ziele für die Assoziation angewendet werden sollen. Beispielsweise könnte eine Zuordnung für Antivirensoftware einmal täglich auf allen verwalteten Knoten in einem AWS-Konto ausgeführt werden. Wenn die Software nicht auf einem Knoten installiert ist, könnte die Assoziation State Manager anweisen, sie zu installieren. Wenn die Software installiert ist, aber der Service nicht ausgeführt wird, könnte die Assoziation State Manager anweisen, den Service zu starten.
- Flexible Planungs-Optionen

  State Manager bietet die folgenden Optionen zum Planen, wenn eine Assoziation ausgeführt wird:

  - Sofortige oder verzögerte Verarbeitung

    Wenn Sie eine Assoziation erstellen, führt das System diese standardmäßig sofort auf den angegebenen Ressourcen aus. Nach der ersten Ausführung wird die Assoziation gemäß dem von Ihnen festgelegten Zeitplan in Intervallen ausgeführt.

    Sie können State Manager anweisen, eine Assoziation nicht sofort auszuführen, indem Sie die Apply association only at next specified Cron-interval (Übernehmen der Assoziation erst für das nächste angegebene Cron-Intervall)-Option in der Konsole oder im`ApplyOnlyAtCronInterval`-Parameter von der Befehlszeile aus verwenden.
  - Cron- und Rate-Ausdrücke

    Wenn Sie eine Zuordnung erstellen, geben Sie den Zeitpunkt an, zu dem State Manager die Konfiguration anwendet. State Manager unterstützt Standardausdrücke für Cron- und Rate-Ausdrücke für die Planung, wenn eine Zuordnung ausgeführt wird. State Manager unterstützt auch Cron-Ausdrücke, die einen Wochentag und das Zahlenzeichen (#) enthalten, um den x-ten Tag eines Monats festzulegen, um eine Zuordnung auszuführen, und das (L)-Zeichen, um den letzten X Tag des Monats anzugeben.

> **Note**
>
> State Manager unterstützt derzeit nicht die Angabe von Monaten in Cron-Ausdrücken für Zuordnungen.

Um weiter zu steuern, wann eine Assoziation ausgeführt wird, z. B. wenn Sie zwei Tage nach dem Patch-Dienstag eine Assoziation ausführen möchten, können Sie einen Offset angeben. Ein Offset definiert, wie viele Tage nach dem geplanten Tag gewartet werden müssen, um eine Assoziation auszuführen.

Weitere Informationen zum Erstellen von Cron- und Rate-Ausdrücken finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

- Mehrere Targeting-Optionen

  Eine Assoziation gibt auch die Ziele für die Assoziation an. State Managerunterstützt die gezielte Ausrichtung von AWS Ressourcen mithilfe von Tags AWS -Ressourcengruppen IDs, einzelnen Knoten oder allen verwalteten Knoten im aktuellen AWS-Region und AWS-Konto.

- Amazon-S3-Support

  Speichern Sie die Befehlsausgabe von Zuordnungsausführungen in einem Amazon-S3-Bucket Ihrer Wahl. Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager.

- EventBridge Unterstützung

  Dieses Systems Manager Manager-Tool wird in EventBridge Amazon-Regeln sowohl als Ereignistyp als auch als Zieltyp unterstützt. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge und Referenz: EventBridge Amazon-Ereignistypen und -muster für Systems Manager.

Entstehen Kosten für die Verwendung von State Manager?

State Manager ist ohne Aufpreis erhältlich.

Themen

- Grundlegendes zur Funktionsweise von State Manager
- Arbeiten mit Zuordnungen in Systems Manager

- Erstellen von Zuordnungen, die MOF-Dateien ausführen
- Erstellen von Zuordnungen, die Ansible-Playbooks ausführen
- Erstellen von Zuordnungen, die Chef-Rezepte ausführen
- Exemplarische Vorgehensweise: Automatisches Update SSM Agent mit AWS CLI
- Exemplarische Vorgehensweise: Automatisches Aktualisieren von PV-Treibern auf EC2 Instanzen für Windows Server

Weitere Informationen

- Bekämpfung von Konfigurationsabweichungen mithilfe von Amazon EC2 Systems Manager und Windows PowerShell DSC
- Konfigurieren Sie EC2 Amazon-Instances in einer Auto Scaling Scaling-Gruppe mit State Manager

Grundlegendes zur Funktionsweise von State Manager

State Manager, ein Tool in AWS Systems Manager, ist ein sicherer und skalierbarer Service, der den Prozess automatisiert, verwaltete Knoten in einer Hybrid- und Multi-Cloud-Infrastruktur in einem von Ihnen definierten Zustand zu halten.

So funktioniert State Manager:

1. Bestimmen Sie den Status, den Sie auf Ihre AWS-Ressourcen anwenden wollen.

   Möchten Sie sicherstellen, dass Ihr verwaltete Knoten mit spezifischen Anwendungen, wie z. B. Antiviren- oder Malware-Anwendungen, konfiguriert ist? Möchten Sie die den Prozess zur Aktualisierung des SSM Agent oder andere AWS-Pakete wie z. B. `AWSPVDriver` automatisieren? Müssen Sie sicherstellen, dass bestimmte Ports geöffnet oder geschlossen sind? Für erste Schritte mit State Manager bestimmen Sie zunächst den Status, den Sie auf Ihre AWS-Ressource anwenden möchten. Der Status, den Sie anwenden möchten, legt fest, welches SSM-Dokument Sie verwenden können, um eine State Manager-Zuordnung zu erstellen.

   Eine State Manager-Assoziation ist eine Konfiguration, die Sie Ihren AWS-Ressourcen zuweisen. Die Konfiguration definiert den Status, den Sie auf Ihren Ressourcen beibehalten möchten. Beispiel: Eine Zuordnung kann angeben, dass auf einem verwalteten Knoten Antiviren-Software installiert sein und ausgeführt werden muss oder dass bestimmte Ports geschlossen sein müssen.

Eine Assoziation gibt einen Zeitplan an, wann die Konfiguration und die Ziele für die Assoziation angewendet werden sollen. Beispielsweise könnte eine Zuordnung für Antivirensoftware einmal täglich auf allen verwalteten Knoten in einem AWS-Konto ausgeführt werden. Wenn die Software nicht auf einem Knoten installiert ist, könnte die Assoziation State Manager anweisen, sie zu installieren. Wenn die Software installiert ist, aber der Service nicht ausgeführt wird, könnte die Assoziation State Manager anweisen, den Service zu starten.

2. Stellen Sie fest, ob ein vorkonfiguriertes SSM-Dokument Ihnen bei der Erstellung des gewünschten Status in Ihren AWS-Ressourcen helfen kann.

   Systems Manager umfasst Dutzende von vorkonfigurierten SSM-Dokumenten, die Sie verwenden können, um eine Zuordnung zu erstellen. Vorkonfigurierte Dokumente sind dafür vorbereitet, allgemeine Aufgaben wie das Installieren von Anwendungen, das Konfigurieren von Amazon CloudWatch, das Ausführen von AWS Systems Manager-Automatisierungen, das Ausführen der PowerShell und von Shell-Skripts und bei Active Directory den Beitritt von verwalteten Knoten zu einer Directory Service-Domain auszuführen.

   Sie können alle SSM-Dokumente in der Systems Manager-Konsole anzeigen. Wählen Sie den Namen eines Dokuments an, um mehr darüber zu erfahren. Nachfolgend finden Sie zwei Beispiele: `AWS-ConfigureAWSPackage` und `AWS-InstallApplication`.

3. Erstellen einer Assoziation.

   Sie können die Assoziation über die Systems Manager-Konsole, die AWS Command Line Interface (AWS CLI), AWS Tools for Windows PowerShell (Tools for Windows PowerShell) oder die Systems Manager API erstellen. Beim Erstellen einer Assoziation geben Sie die folgenden Informationen an:

   - Ein Name für die Assoziation.
   - Die Parameter für das SSM-Dokument (z. B. den Pfad zu der zu installierenden Anwendung oder zu dem Skript, das auf den Knoten ausgeführt werden soll).
   - Ziele für die Zuordnung. Sie können verwaltete Knoten als Ziel auswählen, indem Sie Tags angeben, einzelne Instance-IDs oder eine Gruppe in AWS -Ressourcengruppen wählen. Sie können auch alle verwalteten Knoten in der aktuellen AWS-Region und AWS-Konto als Ziel auswählen. Wenn Ihre Ziele mehr als 1 000 Knoten umfassen, verwendet das System einen stündlichen Drosselungsmechanismus. Das bedeutet, dass Sie möglicherweise Ungenauigkeiten bei der Anzahl Ihrer Statusaggregationen feststellen, da der Aggregationsprozess stündlich und nur dann ausgeführt wird, wenn sich der Ausführungsstatus für einen Knoten ändert.

- Einen Zeitplan für die Häufigkeit der Statusanwendung. Sie können einen Cron- oder Rate-Ausdruck festlegen. Weitere Informationen zum Erstellen von Zeitplänen mit cron- und Rate-Ausdrücken für Zuordnungen finden Sie unter Cron- und Rate-Ausdrücke für Zuordnungen.

> **Note**
>
> State Manager unterstützt derzeit nicht die Angabe von Monaten in Cron-Ausdrücken für Assoziationen.

Wenn Sie den Befehl ausführen, um die Assoziation zu erstellen, bindet Systems Manager die von Ihnen angegebenen Informationen (Zeitplan, Ziele, SSM-Dokument und Parameter) an die anvisierten Ressourcen. Der Status der Zuordnung wird zunächst als „Pending (ausstehend)“ angezeigt, während das System versucht, alle Ziele zu erreichen und sofort den in der Zuordnung angegebenen Status anzuwenden.

> **Note**
>
> Wenn Sie eine neue Zuordnung erstellen, die ausgeführt werden soll, solange eine frühere Zuordnung noch ausgeführt wird, führt dies zu einer Unterbrechung der früheren Zuordnung und die neue Zuordnung wird ausgeführt.

Systems Manager meldet den Status der Anforderung zum Erstellen von Assoziationen auf den Ressourcen. Sie können Statusdetails in der Konsole oder (für verwaltete Knoten) mithilfe der API-Operation DescribeInstanceAssociationsStatus anzeigen. Wenn Sie die Ausgabe des Befehls beim Erstellen einer Zuordnung in Amazon Simple Storage Service (Amazon S3) auswählen, können Sie die Ausgabe auch im angegebenen Amazon S3-Bucket anzeigen.

Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager.

> **Note**
>
> API-Vorgänge, die während der Ausführung einer Zuordnung durch das SSM-Dokument initiiert werden, werden in AWS CloudTrail nicht protokolliert.

4. Überwachen und aktualisieren Sie.

   Nach dem Erstellen der Zuordnung wendet State Manager den Status entsprechend des in der Zuordnung definierten Zeitplans erneut an. Sie können den Status Ihrer Zuordnungen auf der State Manager-Seite in der Konsole oder mit dem direkten Aufruf der Zuordnungs-ID anzeigen, die von Systems Manager beim Erstellen der Zuordnung generiert wurde. Weitere Informationen finden Sie unter Anzeigen von Zuordnungsverläufen. Sie können Ihre Zuordnungsdokumente aktualisieren und Sie bei Bedarf erneut anwenden. Sie können auch mehrere Versionen einer Zuordnung erstellen. Weitere Informationen finden Sie unter Bearbeiten und Erstellen einer neuen Version einer Zuordnung.

Verstehen, wann Zuordnungen auf Ressourcen angewendet werden

Wenn Sie eine Zuordnung erstellen, geben Sie ein SSM-Dokument an, das die Konfiguration, eine Liste der Zielressourcen und einen Zeitplan für die Anwendung der Konfiguration definiert. Standardmäßig führt State Manager die Zuordnung aus, wenn Sie sie erstellen und dann nach Ihrem Zeitplan. State Manager versucht auch, die Zuordnung in den folgenden Situationen auszuführen:

- Zuordnung bearbeiten – State Manager führt die Zuordnung aus, nachdem ein Benutzer seine Änderungen bearbeitet und in einem der folgenden Zuordnungsfelder gespeichert hat: `DOCUMENT_VERSION`, `PARAMETERS`, `SCHEDULE_EXPRESSION`, `OUTPUT_S3_LOCATION`.
- Dokumentbearbeitung – State Manager führt die Zuordnung aus, nachdem ein Benutzer Änderungen am SSM-Dokument, das den Konfigurationsstatus der Zuordnung definiert, bearbeitet und gespeichert hat. Insbesondere wird die Zuordnung nach den folgenden Änderungen am Dokument ausgeführt:
  - Ein Benutzer gibt eine neue `$DEFAULT`-Dokumentversion an, und die Zuordnung wurde mit der `$DEFAULT`-Version erstellt.
  - Ein Benutzer aktualisiert ein Dokument und die Zuordnung wurde mit der `$LATEST`-Version erstellt.
  - Ein Benutzer löscht das Dokument, das beim Erstellen der Zuordnung angegeben wurde.
- Manueller Start – State Manager führt die Zuordnung aus, wenn sie vom Benutzer entweder über die Systems-Manager-Konsole oder programmgesteuert initiiert wird.
- Zieländerungen – State Manager führt die Zuordnung aus, nachdem eine der folgenden Aktivitäten auf einem Zielknoten auftritt:
  - Ein verwalteter Knoten wird zum ersten Mal online geschaltet.
  - Ein verwalteter Knoten geht online, nachdem ein geplanter Zuordnungslauf verpasst wurde.

- Ein verwalteter Knoten wird online gestellt, nachdem er länger als 30 Tage angehalten war.

> **Note**
>
> State Manager überwacht keine Dokumente oder Pakete, die in Zuordnungen in AWS-Konten verwendet werden. Wenn Sie ein Dokument oder Paket in einem Konto aktualisieren, führt die Aktualisierung nicht dazu, dass die Zuordnung im zweiten Konto ausgeführt wird. Sie müssen die Zuordnung im zweiten Konto manuell ausführen.

Verhindern, dass Zuordnungen ausgeführt werden, wenn sich ein Ziel ändert

In einigen Fällen möchten Sie möglicherweise nicht, dass eine Zuordnung ausgeführt wird, wenn sich ein Ziel, das aus verwalteten Knoten besteht, ändert, sondern nur gemäß dem angegebenen Zeitplan.

> **Note**
>
> Das Ausführen eines Automatisierungs-Runbooks ist mit Kosten verbunden. Wenn eine Zuordnung mit einem Automation-Runbook auf alle Instances in Ihrem Konto abzielt und Sie regelmäßig eine große Anzahl von Instances starten, wird das Runbook beim Start auf jeder der Instances ausgeführt. Dies kann zu erhöhten Automatisierungskosten führen.

Um zu verhindern, dass eine Zuordnung ausgeführt wird, wenn sich die Ziele für diese Zuordnung ändern, aktivieren Sie das Kontrollkästchen Zuordnung nur im nächsten angegebenen Cron-Intervall anwenden. Dieses Kontrollkästchen befindet sich im Bereich Zeitplan angeben auf den Seiten Zuordnung erstellen und Zuordnung bearbeiten.

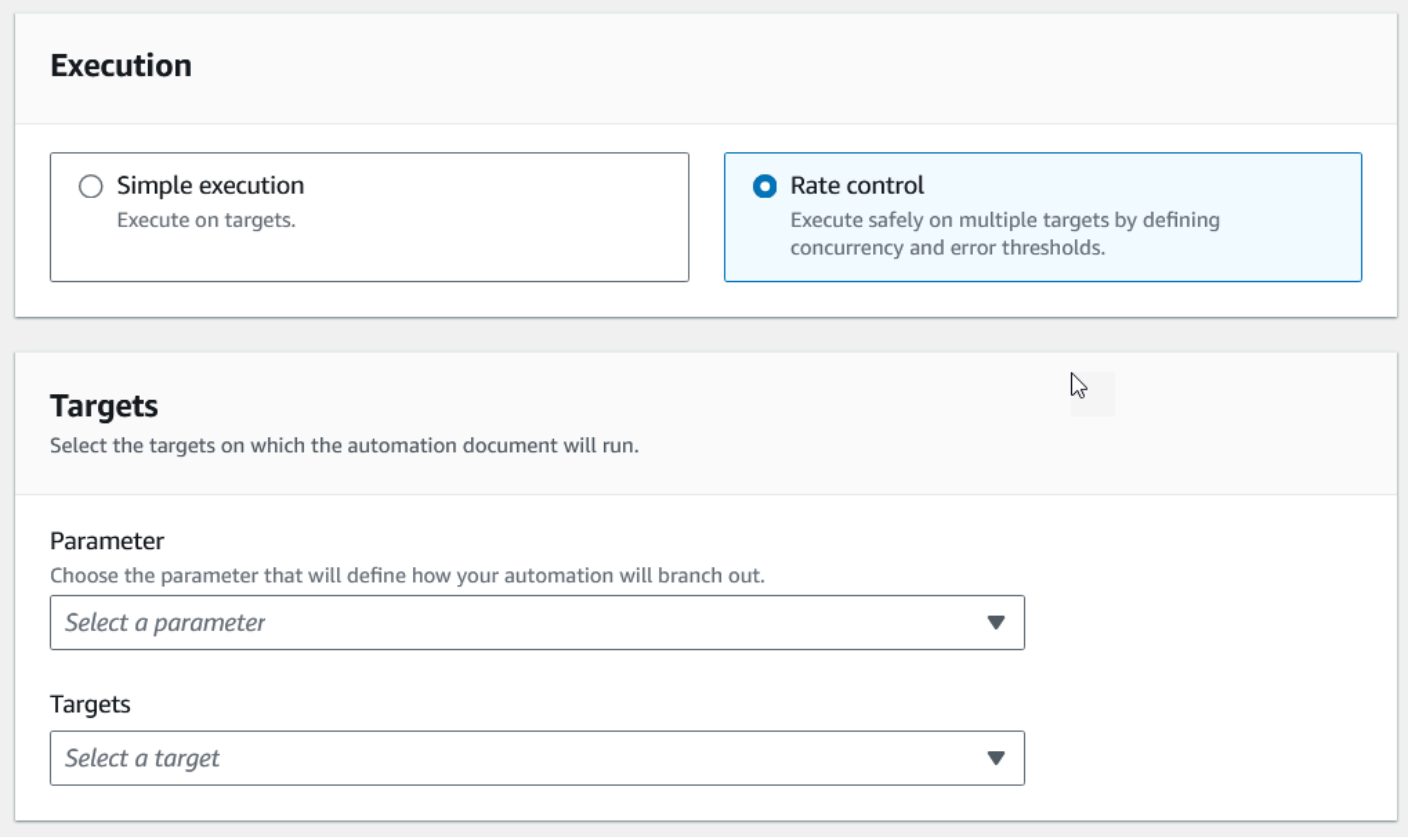
Diese Option gilt für Zuordnungen, die entweder ein Automatisierungs-Runbook oder ein SSM-Dokument enthalten.

Informationen zu Zielupdates mit Automation-Runbooks

Damit Verknüpfungen, die mit Automation-Runbooks erstellt wurden, angewendet werden können, wenn neue Zielknoten erkannt werden, müssen die folgenden Bedingungen erfüllt sein:

- Die Zuordnung muss durch eine Quick Setup-Konfiguration erstellt worden sein. Quick Setup ist ein Tool in AWS Systems Manager. Verknüpfungen, die von anderen Prozessen erstellt werden derzeit nicht unterstützt.
- Das Automation-Runbook muss explizit auf den Ressourcentyp `AWS::EC2::Instance` oder `AWS::SSM::ManagedInstance` abzielen.
- Die Zuordnung muss sowohl Parameter als auch Ziele angeben.

  In der Konsole werden die Felder Parameter und Ziele angezeigt, wenn Sie eine Ausführung mit Ratensteuerung wählen.



  Wenn Sie die API-Aktionen CreateAssociation, CreateAssociationBatch oder UpdateAssociation verwenden, können Sie diese Werte mithilfe der Eingaben `AutomationTargetParameterName` und `Targets` angeben. In jeder dieser API-Aktionen können Sie auch verhindern, dass die Zuordnung bei jeder Änderung eines Ziels ausgeführt wird, indem Sie den `ApplyOnlyAtCronInterval`-Parameter auf `true` setzen.

  Informationen zur Verwendung der Konsole zur Steuerung der Ausführung von Zuordnungen, einschließlich Informationen zur Vermeidung unerwartet hoher Kosten für

Automatisierungsausführungen, finden Sie unter Verstehen, wann Zuordnungen auf Ressourcen angewendet werden.

Arbeiten mit Zuordnungen in Systems Manager

In diesem Abschnitt wird beschrieben, wie Sie State Manager-Assoziation mit der AWS Systems Manager-Konsole, der AWS Command Line Interface (AWS CLI) und AWS -Tools für PowerShell erstellen und verwalten.

Themen

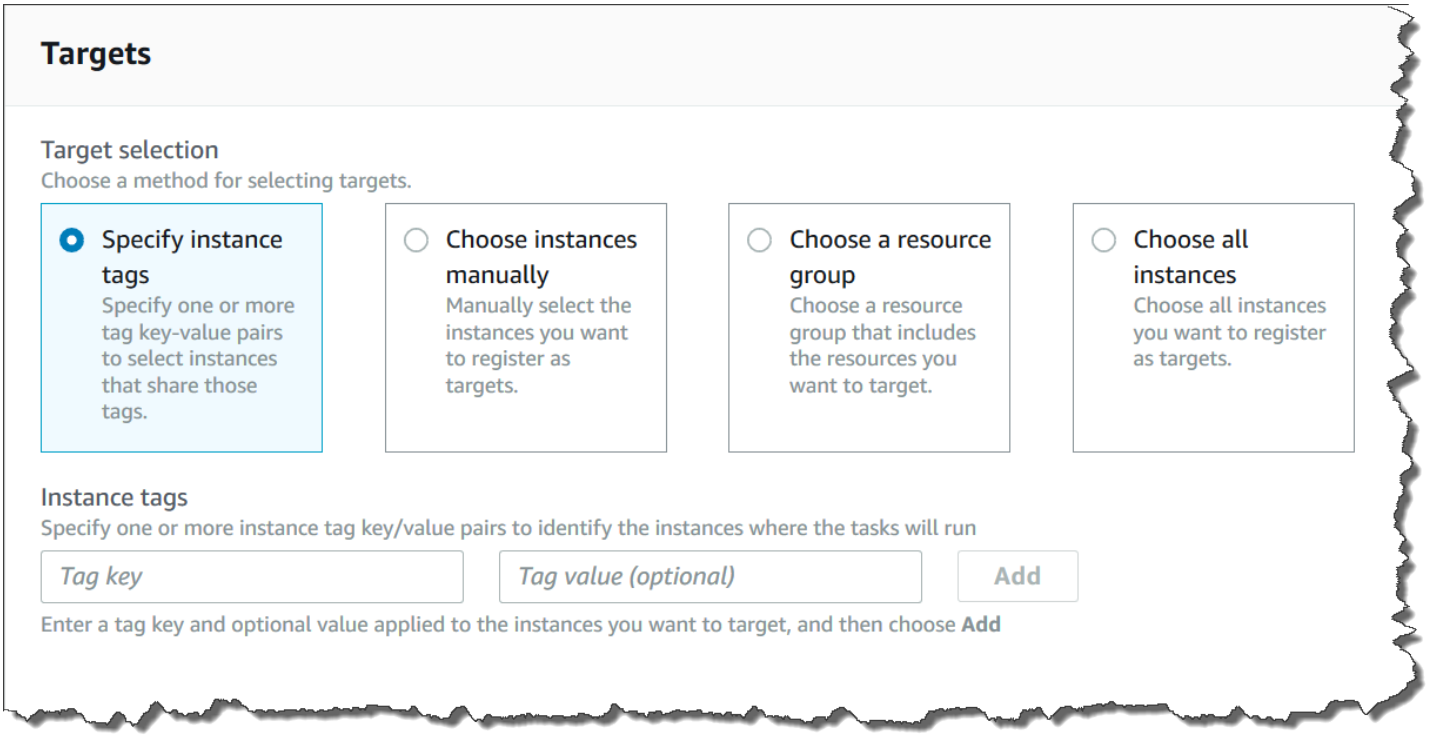
- Verstehen von Zielen und Ratensteuerungen in State Manager Zuordnungen
- Erstellen von Zuordnungen
- Bearbeiten und Erstellen einer neuen Version einer Zuordnung
- Löschen von Zuordnungen
- Ausführen von Auto-Scaling-Gruppen mit Zuordnungen
- Anzeigen von Zuordnungsverläufen
- Arbeiten mit Zuordnungen mithilfe von IAM

Verstehen von Zielen und Ratensteuerungen in State Manager Zuordnungen

In diesem Thema werden Features von State Manager beschrieben, mit denen Sie eine Zuordnung für Dutzende oder Hunderte von Knoten bereitstellen und gleichzeitig steuern können, wie viele Knoten die Zuordnung zum geplanten Zeitpunkt ausführen. State Manager ist ein Tool von AWS Systems Manager.

Nutzung von Zielen

Wenn Sie eine State Manager-Assoziation erstellen, wählen Sie im Abschnitt Targets (Ziele) der Systems Manager-Konsole, welche Knoten mit der Assoziation konfiguriert werden sollen, wie hier gezeigt.



Wenn Sie mithilfe eines Befehlszeilen-Tools wie AWS Command Line Interface (AWS CLI) eine Zuordnung erstellen, geben Sie den Parameter `targets` an. Durch die Ausrichtung auf Knoten können Sie Dutzende, Hunderte oder Tausende von Knoten mit einer Zuordnung konfigurieren, ohne einzelne Knoten angeben oder auswählen zu müssen IDs.

Jeder verwaltete Knoten kann von maximal 20 Zuordnungen betroffen sein.

State Manager enthält die folgenden Zieloptionen bei der Erstellung einer Zuordnung.

Tags angeben

Verwenden Sie diese Option, um einen Tag-Schlüssel und (optional) einen Tag-Wert anzugeben, die Ihren Knoten zugewiesen sind. Wenn Sie die Anforderung ausführen, versucht das System, die Assoziation auf allen Knoten zu erstellen, die dem angegebenen Tag-Schlüssel und -Wert entsprechen. Wenn Sie mehrere Tag-Werte angegeben haben, zielt die Assoziation auf jeden Knoten mit mindestens einem dieser Tag-Werte ab. Wenn das System die Zuordnung erstmals erstellt, führt es die Zuordnung aus. Nach dieser erstmaligen Ausführung führt das System die Zuordnung dem angegebenen Zeitplan entsprechend aus.

Wenn Sie neue Knoten erstellen und diesen Konten den angegebenen Tag-Schlüssel und -Wert zuweisen, wendet das System die Assoziation automatisch an und führt sie sofort und anschließend dem angegebenen Zeitplan entsprechend aus. Dies gilt, wenn die Zuordnung ein Befehls- oder

Richtliniendokument verwendet und nicht angewendet wird, wenn die Zuordnung ein Automation-Runbook verwendet. Wenn Sie die angegebenen Tags aus einem Knoten löschen, führt das System die Assoziation für diese Knoten nicht mehr aus.

> **Note**
>
> Wenn Sie Automation-Runbooks mit verwenden State Manager und die Tagging-Beschränkung Sie daran hindert, ein bestimmtes Ziel zu erreichen, sollten Sie die Verwendung von Automations-Runbooks mit Amazon in Betracht ziehen. EventBridge Weitere Informationen finden Sie unter Führen Sie Automatisierungen auf EventBridge der Grundlage von Ereignissen aus. Weitere Informationen zur Verwendung von State Manager mithilfe von Runbooks finden Sie unter Planen von Automatisierungen mit State Manager-Zuordnungen.

Als bewährte Methode empfehlen wir die Verwendung von Tags, wenn Sie Zuordnungen erstellen, die ein Befehls- oder Richtlinien-Dokument verwenden. Wir empfehlen auch die Verwendung von Tags, wenn Sie Zuordnungen zu Auto-Scaling-Gruppen erstellen. Weitere Informationen finden Sie unter Ausführen von Auto-Scaling-Gruppen mit Zuordnungen.

> **Note**
>
> Notieren Sie die folgenden Informationen:
>
> - Wenn Sie eine Assoziation in der erstellen AWS-Managementkonsole , die mithilfe von Tags auf Knoten abzielt, können Sie nur einen Tag-Schlüssel für eine Automations-Assoziation und fünf Tag-Schlüssel für eine Befehlszuordnung angeben. Alle in der Zuordnung angegebenen Tag-Schlüssel müssen dem Knoten aktuell zugewiesen sein. Sind sie das nicht, kann State Manager den Knoten nicht für eine Zuordnung anvisieren.
> - Wenn Sie die Konsole verwenden und Ihre Knoten gezielt ansprechen möchten, indem Sie mehr als einen Tag-Schlüssel für eine Automatisierungs-Zuordnung und fünf Tag-Tasten für eine Befehlszuweisung verwenden, weisen Sie die Tag-Schlüssel einer AWS -Ressourcengruppen Gruppe zu und fügen Sie der Gruppe die Knoten hinzu. Sie können dann die Option Ressourcengruppe in der Zielliste auswählen, wenn Sie die State Manager-Zuordnung erstellen.
> - Sie können mit der AWS CLI maximal fünf Tag-Schlüssel angeben. Wenn Sie den verwenden AWS CLI, müssen alle im `create-association` Befehl angegebenen Tag-

Tasten dem Knoten aktuell zugewiesen sein. Sind sie das nicht, kann State Manager den Knoten nicht für eine Zuordnung anvisieren.

Manuelles Auswählen von Knoten

Verwenden Sie diese Option, um die Knoten, auf denen Sie die Assoziation erstellen möchten, manuell auszuwählen. Im Bereich Instanzen werden alle von Systems Manager verwalteten Knoten im aktuellen AWS-Konto und angezeigt AWS-Region. Sie können beliebig viele Knoten manuell auswählen. Wenn das System die Zuordnung erstmals erstellt, führt es die Zuordnung aus. Nach dieser erstmaligen Ausführung führt das System die Zuordnung dem angegebenen Zeitplan entsprechend aus.

> **Note**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

Eine Ressourcengruppe auswählen

Verwenden Sie diese Option, um eine Zuordnung für alle Knoten zu erstellen, die von einer AWS -Ressourcengruppen tag- oder AWS CloudFormation stapelbasierten Abfrage zurückgegeben wurden.

Unten folgen Details zum Auswählen von Ressourcengruppen als Ziel für eine Zuordnung.

- Wenn Sie einer Gruppe neue Knoten hinzufügen, ordnet das System die Knoten automatisch der Assoziation zu, die die Ressourcengruppe zum Ziel hat. Wenn das System die Änderung erkennt, wendet es die Assoziation auf die Knoten an. Nach dieser erstmaligen Ausführung führt das System die Zuordnung dem angegebenen Zeitplan entsprechend aus.
- Wenn Sie eine Zuordnung erstellen, die auf eine Ressourcengruppe abzielt, und der `AWS::SSM::ManagedInstance` Ressourcentyp für diese Gruppe angegeben wurde, wird die Zuordnung standardmäßig sowohl auf Amazon Elastic Compute Cloud (Amazon EC2) -Instances als auch auf EC2 Nicht-Knoten in einer Hybrid- und Multi-Cloud-Umgebung ausgeführt.

  Umgekehrt gilt dies auch. Wenn Sie eine Zuordnung erstellen, die auf eine Ressourcengruppe abzielt, und der `AWS::EC2::Instance` Ressourcentyp für diese Gruppe angegeben wurde, wird

die Zuordnung standardmäßig sowohl auf EC2 Nicht-Knoten in einer Hybrid- und Multi-Cloud-Umgebung als auch auf (Amazon- EC2) Instances ausgeführt.

- Wenn Sie eine Zuordnung erstellen, die auf eine Ressourcengruppe abzielt, dürfen der Ressourcengruppe nicht mehr als fünf Tag-Schlüssel zugewiesen oder mehr als fünf Werte für einen Tag-Schlüssel angegeben werden. Wenn eine dieser Bedingungen auf die Tags und Schlüssel zutrifft, die Ihrer Ressourcengruppe zugewiesen sind, kann die Zuordnung nicht ausgeführt werden und gibt einen `InvalidTarget`-Fehler zurück.
- Wenn Sie mithilfe von Tags eine Zuordnung erstellen, die auf eine Ressourcengruppe abzielt, können Sie für den Tagwert nicht die Option (leerer Wert) wählen.
- Wenn Sie eine Ressourcengruppe löschen, führen alle Instances in dieser Gruppe die Zuordnung nicht mehr aus. Es ist ratsam, Zuordnungen zu löschen, die die Gruppe zum Ziel haben.
- Sie können für eine Zuordnung maximal eine einzelne Ressourcengruppe als Ziel auswählen. Mehrere oder verschachtelte Gruppen werden nicht unterstützt.
- Nachdem Sie eine Zuordnung erstellt haben, aktualisiert State Manager die Zuordnung in regelmäßigen Abständen mit Informationen zu Ressourcen in der Ressourcengruppe. Wenn Sie einer Ressourcengruppe neue Ressourcen hinzufügen, hängt es von verschiedenen Faktoren ab, wann das System die Zuordnung auf die neuen Ressourcen anwendet. Sie können den Status der Zuordnung auf der Seite State Manager der Systems Manager-Konsole bestimmen.

> **Warning**
>
> Ein AWS Identity and Access Management (IAM-) Benutzer, eine Gruppe oder eine Rolle mit der Berechtigung, eine Zuordnung zu erstellen, die auf eine Ressourcengruppe von EC2 Amazon-Instances abzielt, hat automatisch die Kontrolle über alle Instances in der Gruppe auf Stammebene. Sie sollten nur vertrauenswürdigen Administratoren die Berechtigung erteilen, Zuordnungen zu erstellen.

Weitere Informationen zu Resource Groups finden Sie unter Was ist AWS -Ressourcengruppen? im AWS -Ressourcengruppen -Benutzerhandbuch.

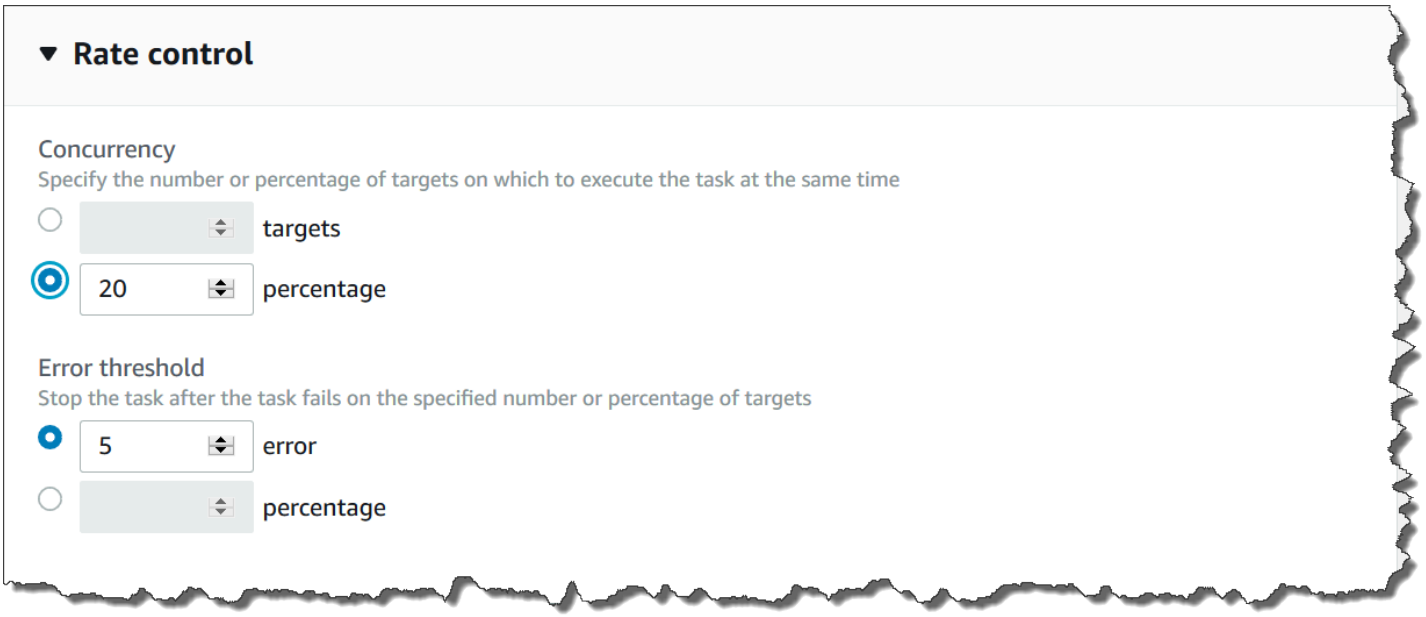
Wählen Sie alle Knoten

Verwenden Sie diese Option, um alle Knoten im aktuellen und als Ziel festzulegen. AWS-Konto AWS-Region Wenn Sie die Anforderung ausführen, sucht das System nach der Zuordnung und versucht, sie auf allen Knoten im aktuellen AWS-Konto und AWS-Region zu erstellen. Wenn das System die

Zuordnung erstmals erstellt, führt es die Zuordnung aus. Nach dieser erstmaligen Ausführung führt das System die Zuordnung dem angegebenen Zeitplan entsprechend aus. Wenn Sie neue Instances erstellen, wendet das System die Assoziation automatisch an und führt sie sofort und anschließend dem angegebenen Zeitplan entsprechend aus.

Verwenden von Ratensteuerungen

Sie können die Ausführung einer Assoziation für Ihre Knoten steuern, indem Sie einen Gleichzeitigkeitswert und einen Fehlerschwellenwert angeben. Der Gleichzeitigkeitswert gibt an, wie viele Knoten die Assoziation gleichzeitig ausführen können. Der Fehlerschwellenwert gibt an, wie viele Assoziationsausführungen fehlschlagen dürfen, bevor Systems Manager einen Befehl an alle mit dieser Assoziation konfigurierten Knoten sendet, um die Ausführung der Assoziation zu beenden. Der Befehl verhindert, dass die Zuordnung vor der nächsten geplanten Zuordnung ausgeführt wird. Die Gleichzeitigkeits- und die Fehlergrenzwertfeature werden gemeinsam als Ratensteuerungen bezeichnet.



Nebenläufigkeit

Durch Angabe eines Gleichzeitigkeitswerts können die Auswirkungen der Ausführung auf Ihre Knoten begrenzt werden, indem Sie angeben, dass jeweils nur eine bestimmte Anzahl von Knoten eine Assoziation gleichzeitig verarbeiten kann. Sie können entweder eine absolute Anzahl an verwalteten Knoten, z. B. 20, oder einen Prozentsatz des Ziel-Knotensatzes, beispielsweise 10 %, angeben.

Bitte beachten Sie die folgenden Einschränkungen und Begrenzungen für die Gleichzeitigkeitsfunktion bei State Manager:

- Wenn Sie eine Assoziation erstellen, indem Sie Ziele angeben, aber keinen Gleichzeitigkeitswert festlegen, dann gibt State Manager automatisch eine maximale Gleichzeitigkeit von 50 Knoten vor.
- Wenn eine Assoziation ausgeführt wird, die die Gleichzeitigkeitsfunktion verwendet, und ein neuer Knoten online geht, der den Zielkriterien entspricht, führen diese neuen Knoten die Assoziation aus, wenn damit der Gleichzeitigkeitswert nicht überschritten wird. Wenn der Gleichzeitigkeitswert überschritten wird, wird der Knoten für das aktuelle Assoziationsausführungsintervall ignoriert. Die Knoten werden dann zum nächsten geplanten Intervall bei normaler Beachtung der Gleichzeitigkeitsbeschränkung ausgeführt.
- Wenn Sie eine Assoziation aktualisieren, die die Gleichzeitigkeitsfunktion verwendet, und diese Assoziation wird gerade auf einer oder mehreren Knoten ausgeführt, dann erhalten diese Knoten die Erlaubnis, die Ausführung der Assoziation abzuschließen. Zuordnungen, deren Ausführung noch nicht begonnen hat, werden nicht mehr ausgeführt. Nachdem die Ausführung laufender Assoziationen abgeschlossen wurde, führen alle Ziel-Knoten die Assoziation sofort erneut aus, da sie aktualisiert wurde. Auch bei dieser erneuten Ausführung gilt der festgelegte Gleichzeitigkeitswert.

Fehlerschwellenwerte

Ein Fehlergrenzwert gibt an, wie viele Assoziationsausführungen auftreten dürfen, bevor Systems Manager einen Befehl zu jedem mit dieser Assoziation konfigurierten Knoten sendet. Der Befehl verhindert, dass die Zuordnung vor der nächsten geplanten Zuordnung ausgeführt wird. Sie können entweder eine absolute Anzahl an Fehlern, z. B. 10, oder einen Prozentsatz des festgelegten Ziels, beispielsweise 10 % festlegen.

Wenn Sie z. B. die absolute Zahl von drei Fehlern angeben, sendet State Manager einen Befehl zum Anhalten, wenn der vierte Fehler zurückgegeben wird. Wenn Sie 0 angeben, sendet State Manager den Befehl zum Anhalten, wenn der erste Fehler gemeldet wird.

Wenn Sie einen Fehlerschwellenwert von 10 % für 50 Zuordnungen angeben, sendet State Manager den Befehl zum Anhalten, wenn der sechste Fehler zurückgegeben wird. Zuordnungen, die bereits ausgeführt werden, wenn ein Fehlerschwellenwert erreicht wird, werden noch abgeschlossen, es besteht jedoch die Möglichkeit, dass einige dieser Zuordnungen fehlschlagen. Um sicherzustellen, dass nicht mehr Fehler als im Fehlerschwellenwert angegeben auftreten, setzen Sie den Wert für die Concurrency (Gleichzeitigkeit) auf 1, sodass die Zuordnungen jeweils einzeln ausgeführt werden.

Es gelten die folgenden Einschränkungen und Begrenzungen für Fehlerschwellenwert in State Manager:

- Fehlerschwellenwerte werden für das aktuelle Intervall übernommen.
- Informationen zu den einzelnen Fehlern werden mit detaillierten Informationen zu den Arbeitsschritten im Zuordnungsverlauf aufgezeichnet.
- Wenn Sie eine Zuordnung unter Verwendung von Zielen erstellen, aber keinen Fehlerschwellenwert angeben, dann gibt State Manager automatisch einen Schwellenwert von 100 % Fehlern vor.

Erstellen von Zuordnungen

State Manager, ein Tool in AWS Systems Manager, hilft Ihnen dabei, Ihre AWS Ressourcen in einem von Ihnen definierten Zustand zu halten und Konfigurationsabweichungen zu reduzieren. Um dies zu tun, verwendet State Manager Assoziationen. Eine Assoziation ist eine Konfiguration, die Sie Ihren AWS -Ressourcen zuweisen. Die Konfiguration definiert den Status, den Sie auf Ihren Ressourcen beibehalten möchten. Beispiel: Eine Zuordnung kann angeben, dass auf einem verwalteten Knoten Antiviren-Software installiert sein und ausgeführt werden muss oder dass bestimmte Ports geschlossen sein müssen.

Eine Assoziation gibt einen Zeitplan an, wann die Konfiguration und die Ziele für die Assoziation angewendet werden sollen. Beispielsweise könnte eine Zuordnung für Antivirensoftware einmal täglich auf allen verwalteten Knoten in einem AWS-Konto ausgeführt werden. Wenn die Software nicht auf einem Knoten installiert ist, könnte die Assoziation State Manager anweisen, sie zu installieren. Wenn die Software installiert ist, aber der Service nicht ausgeführt wird, könnte die Assoziation State Manager anweisen, den Service zu starten.

> **Warning**
>
> Wenn Sie eine Zuordnung erstellen, können Sie eine AWS Ressourcengruppe verwalteter Knoten als Ziel für die Zuordnung auswählen. Wenn ein AWS Identity and Access Management (IAM-) Benutzer, eine Gruppe oder eine Rolle berechtigt ist, eine Zuordnung zu erstellen, die auf eine Ressourcengruppe verwalteter Knoten abzielt, hat dieser Benutzer, diese Gruppe oder Rolle automatisch die Kontrolle über alle Knoten in der Gruppe auf Stammebene. Sie sollten nur vertrauenswürdigen Administratoren die Berechtigung erteilen, Assoziationen zu erstellen.

Zuordnungsziele und Ratensteuerungen

Eine Zuordnung gibt an, welche verwalteten Knoten oder Ziele die Zuordnung erhalten sollen. State Manager enthält mehrere Funktionen, mit denen Sie Ihre verwalteten Knoten gezielt ausrichten und steuern können, wie die Zuordnung für diese Ziele bereitgestellt wird. Weitere Informationen zu Zielen und Ratensteuerungen finden Sie unter Verstehen von Zielen und Ratensteuerungen in State Manager Zuordnungen.

Markierungs-Zuordnungen

Sie können einer Assoziation bei der Erstellung Tags zuweisen, indem Sie ein Befehlszeilentool wie oder verwenden. AWS CLI AWS -Tools für PowerShell Das Hinzufügen von Tags zu einer Zuordnung über die Systems-Manager-Konsole wird nicht unterstützt.

Ausgeführte Zuordnungen

Standardmäßig wird von State Manager eine Zuordnung sofort nach deren Erstellung und dann gemäß dem von Ihnen definierten Zeitplan ausgeführt.

Das System führt auch Zuordnungen nach den folgenden Regeln aus:

- State Manager versucht, die Assoziation während eines Intervalls auf allen angegebenen oder als Ziel ausgewählten Knoten auszuführen.
- Wenn eine Assoziation in einem Intervall nicht ausgeführt wird (weil beispielsweise die Anzahl der Knoten, die die Zuordnung gleichzeitig ausführen können, durch einen Gleichzeitigkeitswert begrenzt wird), versucht State Manager, die Assoziation im nächsten Intervall auszuführen.
- State Manager führt die Zuordnung nach Änderungen an der Konfiguration, den Zielknoten, Dokumenten oder Parametern der Zuordnung aus. Weitere Informationen finden Sie unter Verstehen, wann Zuordnungen auf Ressourcen angewendet werden.
- State Manager zeichnet einen Verlauf für alle übersprungenen Datensätze an. Sie können den Verlauf auf der Registerkarte Execution History (Ausführungsverlauf) anzeigen.

Planen von Zuordnungen

Sie können Zuordnungen so planen, dass sie in einfachen Intervallen ausgeführt werden, z. B. alle 10 Stunden, oder Sie können erweiterte Zeitpläne erstellen, indem Sie benutzerdefinierte Cron- und Rate-Ausdrücke verwenden. Sie können auch verhindern, dass Zuordnungen ausgeführt werden, wenn Sie diese zum ersten Mal erstellen.

Verwenden von Cron- und Rate-Ausdrücken zur Planung von Ausführungen von Zuordnungen

Zusätzlich zu den standardmäßigen Cron- und Rate-Ausdrücken werden von State Manager auch Cron-Ausdrücke unterstützt, die einen Wochentag und das Nummernzeichen (#) enthalten, um den x-ten Tag eines Monats für die Ausführung einer Assoziation zu bestimmen. Hier ist ein Beispiel, das am dritten Dienstag jeden Monats um 23:30 Uhr UTC einen Cron-Zeitplan ausführt:

```
cron(30 23 ? * TUE#3 *)
```

Hier ist ein Beispiel, das am zweiten Donnerstag jeden Monats um Mitternacht UTC läuft:

```
cron(0 0 ? * THU#2 *)
```

State Manager unterstützt auch das (L)-Zeichen, um den letzten X Tag des Monats anzugeben. Hier ist ein Beispiel, das am letzten Dienstag jeden Monats um Mitternacht UTC einen Cron-Zeitplan ausführt:

```
cron(0 0 ? * 3L *)
```

Um weiter zu steuern, wann eine Assoziation ausgeführt wird, z. B. wenn Sie zwei Tage nach dem Patch-Dienstag eine Assoziation ausführen möchten, können Sie einen Offset angeben. Ein Offset definiert, wie viele Tage nach dem geplanten Tag gewartet werden müssen, um eine Assoziation auszuführen. Wenn Sie beispielsweise einen Cron-Zeitplan mit `cron(0 0 ? * THU#2 *)` angegeben haben, können Sie die Nummer 3 im Feld Planversatz angeben, um die Assoziation jeden Sonntag nach dem zweiten Donnerstag im Monat auszuführen.

> **Note**
>
> Um Offsets zu verwenden, müssen Sie entweder Zuordnung nur beim nächsten angegebenen Cron-Intervall in der Konsole anwenden auswählen oder den `ApplyOnlyAtCronInterval`-Parameter über die Befehlszeile angeben. Wenn eine dieser Optionen aktiviert ist, führt State Manager die Zuordnung nicht sofort nach dem Erstellen aus.

Weitere Informationen zu cron- und Rate-Ausdrücken finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

Erstellen einer Zuordnung (Konsole)

Im folgenden Verfahren wird beschrieben, wie mithilfe der Systems Manager-Konsole eine State Manager-Zuordnung erstellt wird.

> **Note**
>
> Notieren Sie die folgenden Informationen:
>
> - Dieses Verfahren beschreibt, wie eine Zuordnung erstellt wird, die entweder ein `Command`- oder ein `Policy`-Dokument zum Ansprechen verwalteter Knoten verwendet. Weitere Informationen zum Erstellen einer Assoziation, die ein Automation-Runbook verwendet, um Knoten oder andere Arten von AWS -Ressourcen anzuvisieren, finden Sie unter Planen von Automatisierungen mit State Manager-Zuordnungen.
> - Sie können bei der Erstellung einer Zuordnung mit der AWS-Managementkonsole maximal fünf Tag-Schlüssel angeben. Alle für die Zuordnung angegebenen Tag-Schlüssel müssen dem Knoten aktuell zugewiesen sein. Sind sie das nicht, kann State Manager den Knoten nicht für die Zuordnung anvisieren.

So erstellen Sie eine State Manager-Zuordnung

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager aus.
3. Wählen Sie Create association (Zuordnung erstellen) aus.
4. Geben Sie im Feld Name einen Namen an.
5. Wählen Sie in der Liste Document (Dokument) die Option neben dem Namen des Dokuments aus. Beachten Sie den Dokumenttyp. Dieses Verfahren gilt für `Command`- und `Policy`-Dokumente. Weitere Informationen zum Erstellen einer Zuordnung, die ein Automation-Runbook verwendet, finden Sie unter Planen von Automatisierungen mit State Manager-Zuordnungen.

   > **Important**
   >
   > State Manager unterstützt nicht das Ausführen von Zuordnungen, die eine neue Version eines Dokuments verwenden, wenn dieses Dokument von einem anderen Konto freigegeben wird. State Manager führt immer die `default`-Version eines Dokuments aus, wenn es von einem anderen Konto freigegeben wird, obwohl die Systems-Manager-Konsole anzeigt, dass eine neue Version verarbeitet wurde. Wenn Sie eine Zuordnung mit einer neuen Version eines Dokuments ausführen möchten, das von einem anderen Konto freigegeben wurde, müssen Sie die Dokumentversion auf `default` einstellen.

6. Geben Sie für Parameters (Parameter) die erforderlichen Eingabeparameter an.
7. (Optional) Wählen Sie einen CloudWatch Alarm aus, den Sie bei Ihrem Verband zur Überwachung beantragen möchten.

   > **Note**
   >
   > Bitte beachten Sie die folgenden Informationen über diesen Schritt.
   >
   > - Die Liste der Alarme zeigt maximal 100 Alarme. Wenn Sie Ihren Alarm nicht in der Liste sehen, verwenden Sie den, AWS Command Line Interface um die Zuordnung zu erstellen. Weitere Informationen finden Sie unter Erstellen einer Zuordnung (Befehlszeile).
   > - Um Ihrem Befehl einen CloudWatch Alarm anzuhängen, muss der IAM-Principal, der die Zuordnung erstellt, über die entsprechende Berechtigung für die `iam:createServiceLinkedRole` Aktion verfügen. Weitere Informationen zu CloudWatch Alarmen finden Sie unter CloudWatch Amazon-Alarme verwenden.
   > - Ausstehende Befehlsaufrufe oder Automatisierungen werden nicht ausgeführt, wenn Ihr Alarm aktiviert wird.

8. Wählen Sie für Ziele eine Option aus. Weitere Informationen zur Verwendung von Zielen finden Sie unter Verstehen von Zielen und Ratensteuerungen in State Manager Zuordnungen.

   > **Note**
   >
   > Damit Verknüpfungen, die mit Automations-Runbooks erstellt wurden, angewendet werden können, wenn neue Zielknoten erkannt werden, müssen bestimmte Bedingungen erfüllt sein. Weitere Informationen finden Sie unter Informationen zu Zielupdates mit Automation-Runbooks.

9. Wählen Sie im Abschnitt Zeitplan angeben entweder Nach Zeitplan oder Kein Zeitplan aus. Wenn Sie On schedule (Auf Zeitplan) auswählen, verwenden Sie die verfügbaren Schaltflächen zum Erstellen eines cron- oder rate-Zeitplans für die Zuordnung.

   Wenn Sie nicht möchten, dass eine Zuordnung unmittelbar nach der Erstellung ausgeführt wird, wählen Sie Apply association only at the next specified Cron interval (Zuordnung erst beim nächsten angegebenen Cron-Intervall anwenden).

10. (Optional) Im Schedule offset (Planversatz), geben Sie eine Zahl zwischen 1 und 6 an.

11. Im Abschnitt Advanced options (Erweiterte Optionen) wählen Sie mit Compliance severity (Compliance-Schweregrad) einen Schweregrad für die Zuordnung und mit Change Calendars (Änderungskalender) einen Änderungskalender für die Zuordnung aus.

    In den Compliance-Berichten finden Sie Informationen dazu, ob die Zuordnung konform ist, zusammen mit dem Schweregrad, den Sie hier angeben. Weitere Informationen finden Sie unter Informationen zu State Manager-Zuordnungs-Compliance.

    Der Änderungskalender bestimmt, wann die Zuordnung ausgeführt wird. Wenn der Kalender geschlossen ist, wird die Zuordnung nicht angewendet. Wenn der Kalender geöffnet ist, wird die Zuordnung entsprechend ausgeführt. Weitere Informationen finden Sie unter AWS Systems Manager Change Calendar.

12. Wählen Sie im Abschnitt Rate control (Ratensteuerung) Optionen für die Ausführung der Assoziation auf mehreren Knoten aus. Weitere Informationen zu Ratensteuerungen finden Sie unter Verstehen von Zielen und Ratensteuerungen in State Manager Zuordnungen.

    Wählen Sie im Abschnitt Gleichzeitigkeit eine Option aus:

    - Wählen Sie Ziele aus, um eine absolute Anzahl von Zielen einzugeben, die die Zuordnung gleichzeitig ausführen können.
    - Wählen Sie Prozentsatz aus, um einen Prozentsatz der Ziele anzugeben, die die Zuordnung gleichzeitig ausführen können.

    Wählen Sie im Abschnitt Fehlerschwellenwert eine Option aus:

    - Wählen Sie Fehler aus und geben Sie die absolute Anzahl erlaubter Fehler an, bis State Manager die Ausführung von Zuordnungen für weitere Ziele beendet.
    - Wählen Sie Prozentsatz aus und geben Sie den Prozentsatz erlaubter Fehler an, bis State Manager die Ausführung von Zuordnungen für weitere Ziele beendet.

13. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

    > **Note**
    >
    > Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht

diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

Im Folgenden finden Sie die minimalen Berechtigungen, die erforderlich sind, um Amazon S3-Ausgabe für eine Zuordnung zu aktivieren. Sie können den Zugriff weiter einschränken, indem Sie IAM-Richtlinien an Benutzer oder Rollen innerhalb eines Kontos anfügen. Ein EC2 Amazon-Instance-Profil sollte mindestens eine IAM-Rolle mit der `AmazonSSMManagedInstanceCore` verwalteten Richtlinie und der folgenden Inline-Richtlinie haben.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Für minimale Berechtigungen muss der Amazon S3-Bucket, in den Sie exportieren, über die von der Amazon S3-Konsole definierten Standardeinstellungen verfügen. Weitere Informationen zum Erstellen eines Amazon S3-Buckets finden Sie unter Erstellen eines Buckets im Amazon S3-Benutzerhandbuch.

> **Note**
>
> API-Vorgänge, die während der Ausführung einer Zuordnung durch das SSM-Dokument initiiert werden, werden in AWS CloudTrail nicht protokolliert.

14. Wählen Sie Zuordnung erstellen.

> **Note**
>
> Wenn Sie die von Ihnen erstellte Zuordnung löschen, wird die Zuordnung nicht mehr auf Zielen dieser Zuordnung ausgeführt.

Erstellen einer Zuordnung (Befehlszeile)

Das folgende Verfahren beschreibt, wie Sie die Tools AWS CLI (unter Linux oderWindows Server) oder verwenden PowerShell , um eine State Manager Zuordnung zu erstellen. Dieser Abschnitt enthält einige Beispiele, die zeigen, wie Ziele und Ratensteuerungen verwendet werden. Mit Zielen und Ratensteuerungen können Sie Dutzenden oder Hunderten von Knoten eine Assoziation zuweisen, während Sie die Ausführung dieser Assoziationen steuern. Weitere Informationen zu Zielen und Ratensteuerungen finden Sie unter Verstehen von Zielen und Ratensteuerungen in State Manager Zuordnungen.

> **Important**
>
> Dieses Verfahren beschreibt, wie eine Zuordnung erstellt wird, die entweder ein `Command`- oder ein `Policy`-Dokument zum Ansprechen verwalteter Knoten verwendet. Informationen zum Erstellen einer Assoziation, die mithilfe eines Automatisierungs-Runbooks auf Knoten oder andere AWS Ressourcentypen abzielt, finden Sie unterPlanen von Automatisierungen mit State Manager-Zuordnungen.

Bevor Sie beginnen

Der Parameter `targets` ist ein Array von Suchkriterien, die Knoten mit einer Kombination aus `Key` und `Value`, die Sie angeben, als Ziel auswählen. Wenn Sie planen, mithilfe des Parameters

`targets` eine Assoziation für Dutzende oder Hunderte von Knoten zu erstellen, überprüfen Sie die folgenden Optionen für die Zielauswahl, bevor Sie mit dem Verfahren beginnen.

Spezifische Knoten ansprechen, indem Sie Folgendes angeben IDs

```
--targets Key=InstanceIds,Values=instance-id-1,instance-id-2,instance-id-3
```

```
--targets
 Key=InstanceIds,Values=i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE,i-07782c72faEXAMPLE
```

Instances mithilfe von -Tags als Ziel auswählen

```
--targets Key=tag:tag-key,Values=tag-value-1,tag-value-2,tag-value-3
```

```
--targets Key=tag:Environment,Values=Development,Test,Pre-production
```

Zielknoten mithilfe von AWS -Ressourcengruppen

```
--targets Key=resource-groups:Name,Values=resource-group-name
```

```
--targets Key=resource-groups:Name,Values=WindowsInstancesGroup
```

Zielt auf alle Instanzen in der aktuellen Version ab AWS-Konto und AWS-Region

```
--targets Key=InstanceIds,Values=*
```

> **Note**
>
> Notieren Sie die folgenden Informationen:
>
> - State Manager unterstützt nicht das Ausführen von Zuordnungen, die eine neue Version eines Dokuments verwenden, wenn dieses Dokument von einem anderen Konto freigegeben wird. State Manager führt immer die `default`-Version eines Dokuments aus, wenn es von einem anderen Konto freigegeben wird, obwohl die Systems-Manager-Konsole anzeigt, dass eine neue Version verarbeitet wurde. Wenn Sie eine Zuordnung mit einer neuen Version eines Dokuments ausführen möchten, das von einem anderen Konto freigegeben wurde, müssen Sie die Dokumentversion auf `default` einstellen.

- State Managerunterstützt den `IncludeChildOrganizationUnits` Parameter für nicht TargetLocation.
- Sie können mit der AWS CLI maximal fünf Tag-Schlüssel angeben. Wenn Sie den verwenden AWS CLI, müssen alle im `create-association` Befehl angegebenen Tag-Schlüssel dem Knoten aktuell zugewiesen sein. Sind sie das nicht, kann State Manager den Knoten nicht für eine Zuordnung anvisieren.
- Beim Erstellen der Zuordnung geben Sie die auch den Zeitplan für die Ausführung an. Geben Sie den Zeitplan mit einem cron- oder Rate-Ausdruck an. Weitere Informationen zu cron- und Rate-Ausdrücken finden Sie unter Cron- und Rate-Ausdrücke für Zuordnungen.
- Damit Verknüpfungen, die mit Automations-Runbooks erstellt wurden, angewendet werden können, wenn neue Zielknoten erkannt werden, müssen bestimmte Bedingungen erfüllt sein. Weitere Informationen finden Sie unter Informationen zu Zielupdates mit Automation-Runbooks.

So erstellen Sie eine Zuordnung

1. Installieren und konfigurieren Sie den AWS CLI oder den AWS -Tools für PowerShell, falls Sie das noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.

2. Verwenden Sie das folgende Format, um einen Befehl zu erstellen, der eine State Manager-Zuordnung erstellt. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen.

   Linux & macOS

```
aws ssm create-association \
    --name document_name \
    --document-version version_of_document_applied \
    --instance-id instances_to_apply_association_on \
    --parameters (if any) \
    --targets target_options \
    --schedule-expression "cron_or_rate_expression" \
    --apply-only-at-cron-interval required_parameter_for_schedule_offsets \
    --schedule-offset number_between_1_and_6 \
    --output-location s3_bucket_to_store_output_details \
```

```
    --association-name association_name \
    --max-errors a_number_of_errors_or_a_percentage_of_target_set \
    --max-concurrency a_number_of_instances_or_a_percentage_of_target_set \
    --compliance-severity severity_level \
    --calendar-names change_calendar_names \
    --target-locations aws_region_or_account \
    --tags "Key=tag_key,Value=tag_value"
```

Windows

```
aws ssm create-association ^
    --name document_name ^
    --document-version version_of_document_applied ^
    --instance-id instances_to_apply_association_on ^
    --parameters (if any) ^
    --targets target_options ^
    --schedule-expression "cron_or_rate_expression" ^
    --apply-only-at-cron-interval required_parameter_for_schedule_offsets ^
    --schedule-offset number_between_1_and_6 ^
    --output-location s3_bucket_to_store_output_details ^
    --association-name association_name ^
    --max-errors a_number_of_errors_or_a_percentage_of_target_set ^
    --max-concurrency a_number_of_instances_or_a_percentage_of_target_set ^
    --compliance-severity severity_level ^
    --calendar-names change_calendar_names ^
    --target-locations aws_region_or_account ^
    --tags "Key=tag_key,Value=tag_value"
```

PowerShell

```
New-SSMAssociation `
    -Name document_name `
    -DocumentVersion version_of_document_applied `
    -InstanceId instances_to_apply_association_on `
    -Parameters (if any) `
    -Target target_options `
    -ScheduleExpression "cron_or_rate_expression" `
    -ApplyOnlyAtCronInterval required_parameter_for_schedule_offsets `
    -ScheduleOffSet number_between_1_and_6 `
    -OutputLocation s3_bucket_to_store_output_details `
    -AssociationName association_name `
    -MaxError  a_number_of_errors_or_a_percentage_of_target_set
    -MaxConcurrency a_number_of_instances_or_a_percentage_of_target_set `
```

```
    -ComplianceSeverity severity_level `
    -CalendarNames change_calendar_names `
    -TargetLocations aws_region_or_account `
    -Tags "Key=tag_key,Value=tag_value"
```

In dem folgenden Beispiel wird eine Assoziaition für Knoten erstellt, die mit `"Environment,Linux"` getaggt sind. Die Assoziation verwendet das Dokument `AWS-UpdateSSMAgent`, um den SSM Agent auf den Ziel-Knoten jeden Sonntagmorgen um 2:00 Uhr UTC zu aktualisieren. Diese Assoziation wird jeweils auf maximal 10 Knoten gleichzeitig ausgeführt. Die Ausführung dieser Assoziation wird außerdem für ein bestimmtes Ausführungsintervall auf weiteren Knoten gestoppt, wenn die Fehlerzählung 5 überschreitet. Der Zuordnung wird für Compliance-Berichte der Schweregrad Mittel zugewiesen.

Linux & macOS

```
aws ssm create-association \
  --association-name Update_SSM_Agent_Linux \
  --targets Key=tag:Environment,Values=Linux \
  --name AWS-UpdateSSMAgent  \
  --compliance-severity "MEDIUM" \
  --schedule-expression "cron(0 2 ? * SUN *)" \
  --max-errors "5" \
  --max-concurrency "10"
```

Windows

```
aws ssm create-association ^
  --association-name Update_SSM_Agent_Linux ^
  --targets Key=tag:Environment,Values=Linux ^
  --name AWS-UpdateSSMAgent  ^
  --compliance-severity "MEDIUM" ^
  --schedule-expression "cron(0 2 ? * SUN *)" ^
  --max-errors "5" ^
  --max-concurrency "10"
```

PowerShell

```
New-SSMAssociation `
  -AssociationName Update_SSM_Agent_Linux `
  -Name AWS-UpdateSSMAgent `
```

```
  -Target @{
      "Key"="tag:Environment"
      "Values"="Linux"
    } `
  -ComplianceSeverity MEDIUM `
  -ScheduleExpression "cron(0 2 ? * SUN *)" `
  -MaxConcurrency 10 `
  -MaxError 5
```

Das folgende Beispiel zielt auf den Knoten ab, IDs indem ein Platzhalterwert (*) angegeben wird. Dadurch kann Systems Manager eine Zuordnung auf allen Knoten im aktuellen AWS-Konto und erstellen AWS-Region. Diese Assoziation wird jeweils auf maximal 10 Knoten gleichzeitig ausgeführt. Die Ausführung dieser Assoziation wird außerdem für ein bestimmtes Ausführungsintervall auf weiteren Knoten gestoppt, wenn die Fehlerzählung 5 überschreitet. Der Zuordnung wird für Compliance-Berichte der Schweregrad Mittel zugewiesen. Diese Assoziation verwendet einen Zeitplan-Offset, was bedeutet, dass sie zwei Tage nach dem angegebenen Cron-Zeitplan ausgeführt wird. Sie enthält auch den `ApplyOnlyAtCronInterval`-Parameter, der erforderlich ist, um den Zeitplan-Offset zu verwenden, und was bedeutet, dass die Assoziation nicht sofort nach ihrer Erstellung ausgeführt wird.

Linux & macOS

```
aws ssm create-association \
  --association-name Update_SSM_Agent_Linux \
  --name "AWS-UpdateSSMAgent" \
  --targets "Key=instanceids,Values=*" \
  --compliance-severity "MEDIUM" \
  --schedule-expression "cron(0 2 ? * SUN#2 *)" \
  --apply-only-at-cron-interval \
  --schedule-offset 2 \
  --max-errors "5" \
  --max-concurrency "10" \
```

Windows

```
aws ssm create-association ^
  --association-name Update_SSM_Agent_Linux ^
  --name "AWS-UpdateSSMAgent" ^
  --targets "Key=instanceids,Values=*" ^
```

```
  --compliance-severity "MEDIUM" ^
  --schedule-expression "cron(0 2 ? * SUN#2 *)" ^
  --apply-only-at-cron-interval ^
  --schedule-offset 2 ^
  --max-errors "5" ^
  --max-concurrency "10" ^
  --apply-only-at-cron-interval
```

PowerShell

```
New-SSMAssociation `
  -AssociationName Update_SSM_Agent_All `
  -Name AWS-UpdateSSMAgent `
  -Target @{
      "Key"="InstanceIds"
      "Values"="*"
    } `
  -ScheduleExpression "cron(0 2 ? * SUN#2 *)" `
  -ApplyOnlyAtCronInterval `
  -ScheduleOffset 2 `
  -MaxConcurrency 10 `
  -MaxError 5 `
  -ComplianceSeverity MEDIUM `
  -ApplyOnlyAtCronInterval
```

Im folgenden Beispiel wird eine Assoziation für Knoten in Ressourcengruppen erstellt. Die Gruppe trägt den Namen „HR-Department“. Die Assoziation verwendet das Dokument `AWS-UpdateSSMAgent`, um den SSM Agent auf den Ziel-Knoten jeden Sonntagmorgen um 2:00 Uhr UTC zu aktualisieren. Diese Assoziation wird jeweils auf maximal 10 Knoten gleichzeitig ausgeführt. Die Ausführung dieser Assoziation wird außerdem für ein bestimmtes Ausführungsintervall auf weiteren Knoten gestoppt, wenn die Fehlerzählung 5 überschreitet. Der Zuordnung wird für Compliance-Berichte der Schweregrad Mittel zugewiesen. Diese Assoziation wird entsprechend dem angegebenen Cron-Zeitplan ausgeführt. Sie wird nicht unmittelbar nach dem Erstellen der Zuordnung ausgeführt.

Linux & macOS

```
aws ssm create-association \
  --association-name Update_SSM_Agent_Linux \
  --targets Key=resource-groups:Name,Values=HR-Department \
```

```
  --name AWS-UpdateSSMAgent  \
  --compliance-severity "MEDIUM" \
  --schedule-expression "cron(0 2 ? * SUN *)" \
  --max-errors "5" \
  --max-concurrency "10" \
  --apply-only-at-cron-interval
```

Windows

```
aws ssm create-association ^
  --association-name Update_SSM_Agent_Linux ^
  --targets Key=resource-groups:Name,Values=HR-Department ^
  --name AWS-UpdateSSMAgent  ^
  --compliance-severity "MEDIUM" ^
  --schedule-expression "cron(0 2 ? * SUN *)" ^
  --max-errors "5" ^
  --max-concurrency "10" ^
  --apply-only-at-cron-interval
```

PowerShell

```
New-SSMAssociation `
  -AssociationName Update_SSM_Agent_Linux `
  -Name AWS-UpdateSSMAgent `
  -Target @{
      "Key"="resource-groups:Name"
      "Values"="HR-Department"
    } `
  -ScheduleExpression "cron(0 2 ? * SUN *)" `
  -MaxConcurrency 10 `
  -MaxError 5 `
  -ComplianceSeverity MEDIUM `
  -ApplyOnlyAtCronInterval
```

Im folgenden Beispiel wird eine Zuordnung erstellt, die auf Knoten ausgeführt wird, die mit einer bestimmten Knoten-ID gekennzeichnet sind. Die Assoziation verwendet das SSM Agent-Dokument zur Aktualisierung von SSM Agent für die Zielknoten, wenn der Änderungskalender geöffnet ist. Die Zuordnung überprüft den Kalenderstatus, wenn er ausgeführt wird. Wenn der Kalender beim Start geschlossen ist und die Zuordnung nur einmal ausgeführt wird, wird diese

nicht erneut ausgeführt, da das Ausführungsfenster der Zuordnung abgelaufen ist. Wenn der Kalender geöffnet ist, wird die Zuordnung entsprechend ausgeführt.

> **Note**
>
> Wenn Sie neue Knoten zu den Tags oder Ressourcengruppen hinzufügen, auf die eine Assoziation wirkt, wenn der Änderungskalender geschlossen ist, wird die Assoziation auf diese Knoten angewendet, sobald der Änderungskalender geöffnet wird.

Linux & macOS

```
aws ssm create-association \
  --association-name CalendarAssociation \
  --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f" \
  --name AWS-UpdateSSMAgent  \
  --calendar-names "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1" \
  --schedule-expression "rate(1day)"
```

Windows

```
aws ssm create-association ^
  --association-name CalendarAssociation ^
  --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f" ^
  --name AWS-UpdateSSMAgent  ^
  --calendar-names "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1" ^
  --schedule-expression "rate(1day)"
```

PowerShell

```
New-SSMAssociation `
  -AssociationName CalendarAssociation `
  -Target @{
      "Key"="tag:instanceids"
      "Values"="i-0cb2b964d3e14fd9f"
    } `
  -Name AWS-UpdateSSMAgent `
  -CalendarNames "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1" `
  -ScheduleExpression "rate(1day)"
```

Im folgenden Beispiel wird eine Zuordnung erstellt, die auf Knoten ausgeführt wird, die mit einer bestimmten Knoten-ID gekennzeichnet sind. Die ZuoAssoziation verwendet das Dokument SSM Agent, um jeden Sonntag um 2:00 Uhr SSM Agent auf den Zielknoten zu aktualisieren. Diese Assoziation wird nur zum angegebenen Cron-Zeitplan ausgeführt, wenn der Änderungskalender geöffnet ist. Wenn die Zuordnung erstellt wird, überprüft sie den Kalenderstatus. Wenn der Kalender geschlossen ist, wird die Zuordnung nicht angewendet. Wenn das Intervall zum Anwenden der Zuordnung am Sonntag um 2:00 Uhr beginnt, prüft die Zuordnung, ob der Kalender geöffnet ist. Wenn der Kalender geöffnet ist, wird die Zuordnung entsprechend ausgeführt.

> **Note**
>
> Wenn Sie neue Knoten zu den Tags oder Ressourcengruppen hinzufügen, auf die eine Assoziation wirkt, wenn der Änderungskalender geschlossen ist, wird die Assoziation auf diese Knoten angewendet, sobald der Änderungskalender geöffnet wird.

Linux & macOS

```
aws ssm create-association \
  --association-name MultiCalendarAssociation \
  --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f" \
  --name AWS-UpdateSSMAgent  \
  --calendar-names "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1"
 "arn:aws:ssm:us-east-2:123456789012:document/testCalendar2" \
  --schedule-expression "cron(0 2 ? * SUN *)"
```

Windows

```
aws ssm create-association ^
  --association-name MultiCalendarAssociation ^
  --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f" ^
  --name AWS-UpdateSSMAgent  ^
  --calendar-names "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1"
 "arn:aws:ssm:us-east-2:123456789012:document/testCalendar2" ^
  --schedule-expression "cron(0 2 ? * SUN *)"
```

PowerShell

```
New-SSMAssociation `
  -AssociationName MultiCalendarAssociation `
  -Name AWS-UpdateSSMAgent `
  -Target @{
      "Key"="tag:instanceids"
      "Values"="i-0cb2b964d3e14fd9f"
    } `
  -CalendarNames "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1"
 "arn:aws:ssm:us-east-2:123456789012:document/testCalendar2" `
  -ScheduleExpression "cron(0 2 ? * SUN *)"
```

> **Note**
>
> Wenn Sie die von Ihnen erstellte Zuordnung löschen, wird die Zuordnung nicht mehr auf Zielen dieser Zuordnung ausgeführt. Wenn Sie den Parameter `apply-only-at-cron-interval` angegeben haben, können Sie diese Option auch zurücksetzen. Geben Sie dazu den Parameter `no-apply-only-at-cron-interval` an, wenn Sie die Zuordnung über die Befehlszeile aktualisieren. Dieser Parameter erzwingt die sofortige Ausführung der Zuordnung nach dem Aktualisieren der Zuordnung und gemäß dem angegebenen Intervall.

Bearbeiten und Erstellen einer neuen Version einer Zuordnung

Sie können eine State Manager-Zuordnung bearbeiten, um den Namen, den Zeitplan, den Schweregrad, Ziele, oder andere Werte zu ändern. Bei Zuordnungen, die auf Dokumenten vom Typ „SSM-Befehl“ basieren, können Sie die Ausgabe des Befehls auch in einen Amazon Simple Storage Service (Amazon S3)-Bucket schreiben. Nachdem Sie eine Zuordnung bearbeitet haben, erstellt State Manager eine neue Version. Sie können unterschiedliche Versionen, wie in den folgenden Schritten beschrieben, nach der Bearbeitung anzeigen.

> **Note**
>
> Damit Verknüpfungen, die mit Automations-Runbooks erstellt wurden, angewendet werden können, wenn neue Zielknoten erkannt werden, müssen bestimmte Bedingungen erfüllt

sein. Weitere Informationen finden Sie unter Informationen zu Zielupdates mit Automation-Runbooks.

Die folgenden Verfahren beschreiben, wie Sie mithilfe der Systems Manager Manager-Konsole () und AWS Command Line Interface AWS -Tools für PowerShell (Tools für AWS CLI PowerShell) eine neue Version einer Zuordnung bearbeiten und erstellen.

Important

State Manager unterstützt nicht das Ausführen von Zuordnungen, die eine neue Version eines Dokuments verwenden, wenn dieses Dokument von einem anderen Konto freigegeben wird. State Manager läuft immer die `default`-Version eines Dokuments, wenn es von einem anderen Konto freigegeben wird, obwohl die Systems-Manager-Konsole anzeigt, dass eine neue Version verarbeitet wurde. Wenn Sie eine Zuordnung mit einer neuen Version eines Dokuments ausführen möchten, das von einem anderen Konto freigegeben wurde, müssen Sie die Dokumentversion auf `default` einstellen.

Bearbeiten einer Zuordnung (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie mithilfe der Systems Manager-Konsole eine neue Version einer Zuordnung bearbeiten und erstellen.

Note

Für Zuordnungen, die SSM-Befehlsdokumente und keine Automation-Runbooks verwenden, erfordert dieses Verfahren, dass Sie Schreibzugriff auf einen vorhandenen Amazon-S3-Bucket haben. Wenn Sie Amazon S3 bisher nicht verwendet haben, bedenken Sie, dass Gebühren für die Nutzung von Amazon S3 anfallen. Weitere Informationen zum Erstellen eines Buckets finden Sie unter Erstellen eines Buckets.

So bearbeiten Sie eine State Manager-Zuordnung

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager aus.

3. Wählen Sie eine vorhandene Zuordnung und dann Bearbeiten aus.
4. Konfigurieren Sie die Zuordnung so erneut, dass sie Ihre aktuellen Anforderungen erfüllt.

   Informationen zu den Zuordnungsoptionen mit `Command`- und `Policy`-Dokumenten finden Sie unter Erstellen von Zuordnungen. Informationen zu Zuordnungsoptionen mit Automation-Runbooks finden Sie unter Planen von Automatisierungen mit State Manager-Zuordnungen.
5. Wählen Sie Save Changes.
6. (Optional) Um Zuordnungsinformationen anzuzeigen, wählen Sie auf der Seite Zuordnungen den Namen der von Ihnen bearbeiteten Zuordnung und dann die Registerkarte Versionen aus. Das System listet alle Version der Zuordnung auf, die Sie erstellt und bearbeitet haben.
7. (Optional) Gehen Sie wie folgt vor, um die Ausgabe für Verknüpfungen anzuzeigen, die auf `Command`-SSM-Dokumenten basieren:

   a. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
   b. Wählen Sie den Namen des Amazon S3-Buckets, den Sie zum Speichern von Befehlsausgaben angegeben haben, und wählen Sie dann den Ordner, dessen Name der ID dem Knoten entspricht, der die Assoziation ausgeführt hat. (Wenn Sie festgelegt haben, Ausgaben in einem Ordner im Bucket zu speichern, öffnen Sie diesen zuerst.)
   c. Zeigen Sie die `stdout`-Datei in einer tieferen Ebenen im Ordner `awsrunPowerShell` an.
   d. Wählen Sie Open oder Download, um den Hostnamen anzuzeigen.

Bearbeiten einer Zuordnung (Befehlszeile)

Das folgende Verfahren beschreibt, wie Sie die AWS CLI (unter Linux oderWindows Server) verwenden oder AWS -Tools für PowerShell eine neue Version einer Assoziation bearbeiten und erstellen.

So bearbeiten Sie eine State Manager-Zuordnung

1. Installieren und konfigurieren Sie die AWS CLI oder die AWS -Tools für PowerShell, falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.
2. Verwenden Sie das folgende Format, um einen Befehl zum Bearbeiten und Erstellen einer neuen Version einer vorhandenen State Manager-Zuordnung zu erstellen. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen.

> **Important**
>
> Wenn Sie `update-association` aufrufen, löscht das System alle optionalen Parameter aus der Anforderung und überschreibt die Zuordnung mit Nullwerten für diese Parameter. Dies ist beabsichtigt. Sie müssen alle optionalen Parameter im Aufruf angeben, auch wenn Sie die Parameter nicht ändern. Dies umfasst den `--name`-Parameter. Bevor Sie diese Aktion aufrufen, empfehlen wir, dass Sie den `describe-association`-API-Vorgang aufrufen und sich alle optionalen Parameter notieren, die für Ihren `update-association`-Aufruf erforderlich sind.

Linux & macOS

```
aws ssm update-association \
    --name document_name \
    --document-version version_of_document_applied \
    --instance-id instances_to_apply_association_on \
    --parameters (if any) \
    --targets target_options \
    --schedule-expression "cron_or_rate_expression" \
    --schedule-offset "number_between_1_and_6" \
    --output-location s3_bucket_to_store_output_details \
    --association-name association_name \
    --max-errors a_number_of_errors_or_a_percentage_of_target_set \
    --max-concurrency a_number_of_instances_or_a_percentage_of_target_set \
    --compliance-severity severity_level \
    --calendar-names change_calendar_names \
    --target-locations aws_region_or_account
```

Windows

```
aws ssm update-association ^
    --name document_name ^
    --document-version version_of_document_applied ^
    --instance-id instances_to_apply_association_on ^
    --parameters (if any) ^
    --targets target_options ^
    --schedule-expression "cron_or_rate_expression" ^
    --schedule-offset "number_between_1_and_6" ^
    --output-location s3_bucket_to_store_output_details ^
```

```
    --association-name association_name ^
    --max-errors a_number_of_errors_or_a_percentage_of_target_set ^
    --max-concurrency a_number_of_instances_or_a_percentage_of_target_set ^
    --compliance-severity severity_level ^
    --calendar-names change_calendar_names ^
    --target-locations aws_region_or_account
```

PowerShell

```
Update-SSMAssociation `
    -Name document_name `
    -DocumentVersion version_of_document_applied `
    -InstanceId instances_to_apply_association_on `
    -Parameters (if any) `
    -Target target_options `
    -ScheduleExpression "cron_or_rate_expression" `
    -ScheduleOffset "number_between_1_and_6" `
    -OutputLocation s3_bucket_to_store_output_details `
    -AssociationName association_name `
    -MaxError  a_number_of_errors_or_a_percentage_of_target_set
    -MaxConcurrency a_number_of_instances_or_a_percentage_of_target_set `
    -ComplianceSeverity severity_level `
    -CalendarNames change_calendar_names `
    -TargetLocations aws_region_or_account
```

Im folgenden Beispiel wird eine vorhandene Zuordnung aktualisiert, um den Namen in `TestHostnameAssociation2` zu ändern. Die neue Zuordnungsversion wird stündlich ausgeführt und schreibt die Ausgabe der Befehle in den angegebenen Amazon S3-Bucket.

Linux & macOS

```
aws ssm update-association \
  --association-id 8dfe3659-4309-493a-8755-01234EXAMPLE \
  --association-name TestHostnameAssociation2 \
  --parameters commands="echo Association" \
  --output-location S3Location='{OutputS3Region=us-
east-1,OutputS3BucketName=amzn-s3-demo-bucket,OutputS3KeyPrefix=logs}' \
  --schedule-expression "cron(0 */1 * * ? *)"
```

Windows

```
aws ssm update-association ^
  --association-id 8dfe3659-4309-493a-8755-01234EXAMPLE ^
  --association-name TestHostnameAssociation2 ^
  --parameters commands="echo Association" ^
  --output-location S3Location='{OutputS3Region=us-
east-1,OutputS3BucketName=amzn-s3-demo-bucket,OutputS3KeyPrefix=logs}' ^
  --schedule-expression "cron(0 */1 * * ? *)"
```

PowerShell

```
Update-SSMAssociation `
  -AssociationId b85ccafe-9f02-4812-9b81-01234EXAMPLE `
  -AssociationName TestHostnameAssociation2 `
  -Parameter @{"commands"="echo Association"} `
  -S3Location_OutputS3BucketName amzn-s3-demo-bucket `
  -S3Location_OutputS3KeyPrefix logs `
  -S3Location_OutputS3Region us-east-1 `
  -ScheduleExpression "cron(0 */1 * * ? *)"
```

Im folgenden Beispiel wird eine vorhandene Zuordnung aktualisiert, um den Namen in `CalendarAssociation` zu ändern. Die neue Zuordnung wird ausgeführt, wenn der Kalender geöffnet ist, und schreibt die Befehlsausgabe in den angegebenen Amazon S3-Bucket.

Linux & macOS

```
aws ssm update-association \
  --association-id 8dfe3659-4309-493a-8755-01234EXAMPLE \
  --association-name CalendarAssociation \
  --parameters commands="echo Association" \
  --output-location S3Location='{OutputS3Region=us-
east-1,OutputS3BucketName=amzn-s3-demo-bucket,OutputS3KeyPrefix=logs}' \
  --calendar-names "arn:aws:ssm:us-east-1:123456789012:document/testCalendar2"
```

Windows

```
aws ssm update-association ^
  --association-id 8dfe3659-4309-493a-8755-01234EXAMPLE ^
  --association-name CalendarAssociation ^
```

```
  --parameters commands="echo Association" ^
  --output-location S3Location='{OutputS3Region=us-
east-1,OutputS3BucketName=amzn-s3-demo-bucket,OutputS3KeyPrefix=logs}' ^
  --calendar-names "arn:aws:ssm:us-east-1:123456789012:document/testCalendar2"
```

PowerShell

```
Update-SSMAssociation `
  -AssociationId b85ccafe-9f02-4812-9b81-01234EXAMPLE `
  -AssociationName CalendarAssociation `
  -AssociationName OneTimeAssociation `
  -Parameter @{"commands"="echo Association"} `
  -S3Location_OutputS3BucketName amzn-s3-demo-bucket `
  -CalendarNames "arn:aws:ssm:us-east-1:123456789012:document/testCalendar2"
```

Im folgenden Beispiel wird eine vorhandene Zuordnung aktualisiert, um den Namen in `MultiCalendarAssociation` zu ändern. Die neue Zuordnung wird ausgeführt, wenn die Kalender geöffnet sind, und schreibt die Befehlsausgabe in den angegebenen Amazon S3-Bucket.

Linux & macOS

```
aws ssm update-association \
  --association-id 8dfe3659-4309-493a-8755-01234EXAMPLE \
  --association-name MultiCalendarAssociation \
  --parameters commands="echo Association" \
  --output-location S3Location='{OutputS3Region=us-
east-1,OutputS3BucketName=amzn-s3-demo-bucket,OutputS3KeyPrefix=logs}' \
  --calendar-names "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1"
 "arn:aws:ssm:us-east-2:123456789012:document/testCalendar2"
```

Windows

```
aws ssm update-association ^
  --association-id 8dfe3659-4309-493a-8755-01234EXAMPLE ^
  --association-name MultiCalendarAssociation ^
  --parameters commands="echo Association" ^
  --output-location S3Location='{OutputS3Region=us-
east-1,OutputS3BucketName=amzn-s3-demo-bucket,OutputS3KeyPrefix=logs}' ^
```

```
  --calendar-names "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1"
 "arn:aws:ssm:us-east-2:123456789012:document/testCalendar2"
```

PowerShell

```
Update-SSMAssociation `
  -AssociationId b85ccafe-9f02-4812-9b81-01234EXAMPLE `
  -AssociationName MultiCalendarAssociation `
  -Parameter @{"commands"="echo Association"} `
  -S3Location_OutputS3BucketName amzn-s3-demo-bucket `
  -CalendarNames "arn:aws:ssm:us-east-1:123456789012:document/testCalendar1"
 "arn:aws:ssm:us-east-2:123456789012:document/testCalendar2"
```

3. Um die neue Version der Zuordnung anzuzeigen, führen Sie den folgenden Befehl aus.

   Linux & macOS

   ```
   aws ssm describe-association \
     --association-id b85ccafe-9f02-4812-9b81-01234EXAMPLE
   ```

   Windows

   ```
   aws ssm describe-association ^
     --association-id b85ccafe-9f02-4812-9b81-01234EXAMPLE
   ```

   PowerShell

   ```
   Get-SSMAssociation `
     -AssociationId b85ccafe-9f02-4812-9b81-01234EXAMPLE | Select-Object *
   ```

   Das System gibt unter anderem folgende Informationen zurück

   Linux & macOS

   ```
   {
       "AssociationDescription": {
           "ScheduleExpression": "cron(0 */1 * * ? *)",
           "OutputLocation": {
               "S3Location": {
                   "OutputS3KeyPrefix": "logs",
   ```

```
                "OutputS3BucketName": "amzn-s3-demo-bucket",
                "OutputS3Region": "us-east-1"
            }
        },
        "Name": "AWS-RunPowerShellScript",
        "Parameters": {
            "commands": [
                "echo Association"
            ]
        },
        "LastExecutionDate": 1559316400.338,
        "Overview": {
            "Status": "Success",
            "DetailedStatus": "Success",
            "AssociationStatusAggregatedCount": {}
        },
        "AssociationId": "b85ccafe-9f02-4812-9b81-01234EXAMPLE",
        "DocumentVersion": "$DEFAULT",
        "LastSuccessfulExecutionDate": 1559316400.338,
        "LastUpdateAssociationDate": 1559316389.753,
        "Date": 1559314038.532,
        "AssociationVersion": "2",
        "AssociationName": "TestHostnameAssociation2",
        "Targets": [
            {
                "Values": [
                    "Windows"
                ],
                "Key": "tag:Environment"
            }
        ]
    }
}
```

Windows

```
{
    "AssociationDescription": {
        "ScheduleExpression": "cron(0 */1 * * ? *)",
        "OutputLocation": {
            "S3Location": {
                "OutputS3KeyPrefix": "logs",
                "OutputS3BucketName": "amzn-s3-demo-bucket",
```

```
                "OutputS3Region": "us-east-1"
            }
        },
        "Name": "AWS-RunPowerShellScript",
        "Parameters": {
            "commands": [
                "echo Association"
            ]
        },
        "LastExecutionDate": 1559316400.338,
        "Overview": {
            "Status": "Success",
            "DetailedStatus": "Success",
            "AssociationStatusAggregatedCount": {}
        },
        "AssociationId": "b85ccafe-9f02-4812-9b81-01234EXAMPLE",
        "DocumentVersion": "$DEFAULT",
        "LastSuccessfulExecutionDate": 1559316400.338,
        "LastUpdateAssociationDate": 1559316389.753,
        "Date": 1559314038.532,
        "AssociationVersion": "2",
        "AssociationName": "TestHostnameAssociation2",
        "Targets": [
            {
                "Values": [
                    "Windows"
                ],
                "Key": "tag:Environment"
            }
        ]
    }
}
```

PowerShell

```
AssociationId                : b85ccafe-9f02-4812-9b81-01234EXAMPLE
AssociationName              : TestHostnameAssociation2
AssociationVersion           : 2
AutomationTargetParameterName :
ComplianceSeverity           :
Date                         : 5/31/2019 2:47:18 PM
DocumentVersion              : $DEFAULT
InstanceId                   :
```

```
LastExecutionDate             : 5/31/2019 3:26:40 PM
LastSuccessfulExecutionDate   : 5/31/2019 3:26:40 PM
LastUpdateAssociationDate     : 5/31/2019 3:26:29 PM
MaxConcurrency                :
MaxErrors                     :
Name                          : AWS-RunPowerShellScript
OutputLocation                :
 Amazon.SimpleSystemsManagement.Model.InstanceAssociationOutputLocation
Overview                      :
 Amazon.SimpleSystemsManagement.Model.AssociationOverview
Parameters                    : {[commands,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
ScheduleExpression            : cron(0 */1 * * ? *)
Status                        :
Targets                       : {tag:Environment}
```

Löschen von Zuordnungen

Gehen Sie wie folgt vor, um eine Zuordnung mithilfe der AWS Systems Manager-Konsole zu löschen.

Löschen einer Zuordnung

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager aus.
3. Wählen Sie eine Zuordnung aus und wählen Sie Löschen aus.

Sie können mehrere Verknüpfungen in einem einzigen Vorgang löschen, indem Sie eine Automatisierung von der AWS Systems Manager-Konsole aus ausführen. Wenn Sie mehrere Verknüpfungen zum Löschen auswählen, startet State Manager die Startseite des Automatisierungs-Runbooks mit den als Eingabeparameterwerten eingegebenen Zuordnungs-IDs.

So können mehrere Verknüpfungen in einem einzigen Vorgang löschen

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager aus.
3. Wählen Sie jede Zuordnung aus, die Sie löschen möchten, und wählen Sie dann Löschen.

4. (Optional) Wählen Sie im Bereich Zusätzliche Eingabeparameter den Amazon-Ressourcennamen (ARN) für die angenommene Rolle aus, die die Automatisierung während der Ausführung verwenden soll. Um eine neue Rolle zu erstellen, wählen Sie Erstellen.
5. Wählen Sie Absenden aus.

Ausführen von Auto-Scaling-Gruppen mit Zuordnungen

Die bewährte Methode beim Verwenden von Zuordnungen zum Ausführen von Auto-Scaling-Gruppen besteht darin, Tag-Ziele zu verwenden. Wenn Sie Tags nicht verwenden, könnten Sie das Zuordnungslimit erreichen.

Wenn alle Knoten mit demselben Schlüssel und demselben Wert versehen sind, benötigen Sie nur eine Assoziation, um Ihre Auto-Scaling-Gruppe auszuführen. Im folgenden Verfahren wird beschrieben, wie Sie so eine Zuordnung erstellen.

Erstellen einer Zuordnung, auf der Auto-Scaling-Gruppen ausgeführt wird

1. Stellen Sie sicher, dass alle Knoten in der Auto-Scaling-Gruppe mit demselben Schlüssel und demselben Wert versehen sind. Weitere Informationen zum Markieren von Knoten finden Sie unter Markieren von Auto-Scaling-Gruppen und Knoten im AWS Auto Scaling - Benutzerhandbuch.
2. Erstellen Sie eine Zurodnung unter Verwendung des Verfahrens in Arbeiten mit Zuordnungen in Systems Manager.

   Wenn Sie in der Konsole arbeiten, wählen Sie Specify instance tags (Instance-Tags angeben) im Feld Targets (Ziele). Geben Sie für Instance-Tags den Tag-Schlüssel und -Welt für Ihre Auto-Scaling-Gruppe ein.

   Wenn Sie AWS Command Line Interface (AWS CLI) verwenden, geben Sie an, `--targets Key=tag:`*`tag-key`*`,Values=`*`tag-value`* wo der Schlüssel und der Wert mit dem übereinstimmen, womit Sie Ihre Knoten markiert haben.

Anzeigen von Zuordnungsverläufen

Mithilfe der DescribeAssociationExecutionsAPI-Operation können Sie alle Ausführungen für eine bestimmte Zuordnungs-ID anzeigen. Verwenden Sie diesen Vorgang, um Status, detaillierten Status, Ergebnisse, Uhrzeit der letzten Ausführung und weitere Informationen zu einer State Manager-Zuordnung einzusehen. State Manager ist ein Tool in AWS Systems Manager. Diese API-

Operation enthält auch Filter, mit denen Sie entsprechend den von Ihnen festgelegten Kriterien nach Zuordnungen suchen können. Sie können beispielsweise genaue Angaben zu Datum und Uhrzeit machen und mithilfe eines GREATER_THAN-Filters Ausführungen anzeigen, die nach dem angegebenen Datum und der angegebenen Uhrzeit verarbeitet wurden.

Wenn beispielsweise die Ausführung einer Zuordnung fehlgeschlagen ist, können Sie mithilfe des DescribeAssociationExecutionTargetsAPI-Vorgangs einen Drilldown zu den Details einer bestimmten Ausführung durchführen. Dieser Vorgang zeigt Ihnen die Ressourcen, z. B. den Knoten IDs, auf dem die Zuordnung ausgeführt wurde, und die verschiedenen Zuordnungsstatus. Anschließend können Sie sehen, bei welchen Ressourcen oder Knoten eine Assoziation nicht ausgeführt werden konnte. Anhand der Ressourcen-ID können Sie dann die Details der Befehlsausführung anzeigen, um zu bestimmen, welcher Schritt in einem Befehl fehlgeschlagen ist.

Die Beispiele in diesem Abschnitt enthalten auch Informationen darüber, wie Sie mithilfe des StartAssociationsOnceAPI-Vorgangs eine Assoziation einmal bei der Erstellung ausführen können. Sie können mithilfe dieser API-Operation fehlgeschlagenen Zuordnungsausführungen nachgehen. Wenn Sie sehen, dass eine Zuordnung fehlgeschlagen ist, können Sie eine Änderung an der Ressource vornehmen und dann die Zuordnung sofort ausführen, um zu sehen, ob die Änderung an der Ressource nun eine erfolgreiche Ausführung der Zuordnung zulässt.

> **Note**
>
> API-Vorgänge, die während der Ausführung einer Zuordnung durch das SSM-Dokument initiiert werden, werden in AWS CloudTrail nicht protokolliert.

Anzeigen von Zuordnungsverläufen (Konsole)

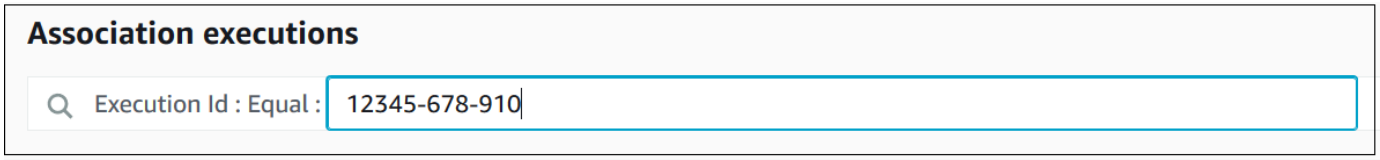
Mit dem folgenden Verfahren können Sie den Ausführungsverlauf für eine bestimmte Zuordnungs-ID und anschließend Ausführungsdetails für eine oder mehrere Ressourcen anzeigen.

So zeigen Sie den Ausführungsverlauf für eine bestimmte Zuordnungs-ID an

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie State Manager.
3. Wählen Sie im Feld Association id (Zuordnungs-ID) eine Zuordnung aus, deren Verlauf Sie anzeigen möchten.
4. Klicken Sie auf die Schaltfläche View details (Details ansehen).

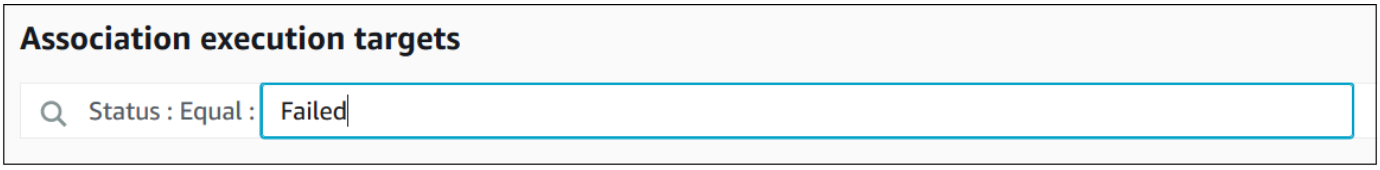
5. Wählen Sie die Registerkarte Execution history (Ausführungsverlauf).
6. Wählen Sie eine Zuordnung aus, für die Sie Ausführungsdetails auf Ressourcenebene anzeigen möchten. Wählen Sie z. B. eine Zuordnung mit dem Status Failed (Fehlgeschlagen) aus. Anschließend können Sie die Ausführungsdetails für die Koten anzeigen, bei denen das Ausführen der Assoziation fehlgeschlagen ist.

   Verwenden Sie die Suchfeldfilter zur Suche nach der Ausführung, für die Sie Details anzeigen möchten.

7. Wählen eine Ausführungs-ID aus. Die Seite Association execution targets (Zuordnungsausführungsziele) wird geöffnet. Diese Seite zeigt alle Ressourcen an, die die Zuordnung ausgeführt haben.
8. Wählen Sie eine Ressourcen-ID aus, um spezifische Informationen zu dieser Ressource anzuzeigen.

   Verwenden Sie die Suchfeldfilter zur Suche nach der Ressource, für die Sie Details anzeigen möchten.

9. Wenn Sie eine Zuordnung untersuchen, die nicht ausgeführt werden konnte, können Sie mit der Schaltfläche Apply association now (Zuordnung nun anwenden) eine Zuordnung nur einmal zum Zeitpunkt der Erstellung ausführen. Nachdem Sie Änderungen an der Ressource vorgenommen haben, bei der die Zuordnung nicht ausgeführt werden konnte, wählen Sie den Link Association ID (Zuordnungs-ID) im Navigations-Breadcrumb aus.
10. Klicken Sie auf die Schaltfläche Apply association now (Zuordnung nun anwenden). Wenn die Ausführung abgeschlossen ist, überprüfen Sie, ob die Zuordnungsausführung erfolgreich war.

Anzeigen von Zuordnungsverläufen (Befehlszeile)

Das folgende Verfahren beschreibt, wie Sie AWS Command Line Interface (AWS CLI) (unter Linux oderWindows Server) verwenden oder AWS -Tools für PowerShell den Ausführungsverlauf für eine

bestimmte Zuordnungs-ID anzeigen. Im Anschluss an dieses Verfahren wird beschrieben, wie Sie Ausführungsdetails für eine oder mehrere Ressourcen anzeigen.

So zeigen Sie den Ausführungsverlauf für eine bestimmte Zuordnungs-ID an

1. Installieren und konfigurieren Sie das AWS CLI oder das AWS -Tools für PowerShell, falls Sie das noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.

2. Führen Sie den folgenden Befehl aus, um eine Liste von Ausführungen für eine bestimmte Zuordnungs-ID anzuzeigen.

   Linux & macOS

   ```
   aws ssm describe-association-executions \
     --association-id ID \
     --filters Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=GREATER_THAN
   ```

   > **Note**
   >
   > Dieser Befehl wendet einen Filter an, um die Ergebnisse auf solche Ausführungen einzuschränken, die nach einem bestimmten Datum und einer bestimmten Uhrzeit aufgetreten sind. Wenn Sie alle Ausführungen für eine bestimmte Zuordnungs-ID anzeigen möchten, entfernen Sie den Parameter `--filters` und den Wert `Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=GREATER_THAN`.

   Windows

   ```
   aws ssm describe-association-executions ^
     --association-id ID ^
     --filters Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=GREATER_THAN
   ```

   > **Note**
   >
   > Dieser Befehl wendet einen Filter an, um die Ergebnisse auf solche Ausführungen einzuschränken, die nach einem bestimmten Datum und einer bestimmten Uhrzeit

aufgetreten sind. Wenn Sie alle Ausführungen für eine bestimmte Zuordnungs-ID anzeigen möchten, entfernen Sie den Parameter `--filters` und den Wert `Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=GREATER_THAN`.

PowerShell

```
Get-SSMAssociationExecution `
  -AssociationId ID `
  -Filter
 @{"Key"="CreatedTime";"Value"="2019-06-01T19:15:38.372Z";"Type"="GREATER_THAN"}
```

> **Note**
>
> Dieser Befehl wendet einen Filter an, um die Ergebnisse auf solche Ausführungen einzuschränken, die nach einem bestimmten Datum und einer bestimmten Uhrzeit aufgetreten sind. Wenn Sie alle Ausführungen für eine bestimmte Zuordnungs-ID anzeigen möchten, entfernen Sie den Parameter `-Filter` und den Wert `@{"Key"="CreatedTime";"Value"="2019-06-01T19:15:38.372Z";"Type"="GREAT`

Das System gibt unter anderem folgende Informationen zurück

Linux & macOS

```
{
   "AssociationExecutions":[
      {
         "Status":"Success",
         "DetailedStatus":"Success",
         "AssociationId":"c336d2ab-09de-44ba-8f6a-6136cEXAMPLE",
         "ExecutionId":"76a5a04f-caf6-490c-b448-92c02EXAMPLE",
         "CreatedTime":1523986028.219,
         "AssociationVersion":"1"
      },
      {
         "Status":"Success",
         "DetailedStatus":"Success",
         "AssociationId":"c336d2ab-09de-44ba-8f6a-6136cEXAMPLE",
         "ExecutionId":"791b72e0-f0da-4021-8b35-f95dfEXAMPLE",
```

```
         "CreatedTime":1523984226.074,
         "AssociationVersion":"1"
      },
      {
         "Status":"Success",
         "DetailedStatus":"Success",
         "AssociationId":"c336d2ab-09de-44ba-8f6a-6136cEXAMPLE",
         "ExecutionId":"ecec60fa-6bb0-4d26-98c7-140308EXAMPLE",
         "CreatedTime":1523982404.013,
         "AssociationVersion":"1"
      }
   ]
}
```

Windows

```
{
   "AssociationExecutions":[
      {
         "Status":"Success",
         "DetailedStatus":"Success",
         "AssociationId":"c336d2ab-09de-44ba-8f6a-6136cEXAMPLE",
         "ExecutionId":"76a5a04f-caf6-490c-b448-92c02EXAMPLE",
         "CreatedTime":1523986028.219,
         "AssociationVersion":"1"
      },
      {
         "Status":"Success",
         "DetailedStatus":"Success",
         "AssociationId":"c336d2ab-09de-44ba-8f6a-6136cEXAMPLE",
         "ExecutionId":"791b72e0-f0da-4021-8b35-f95dfEXAMPLE",
         "CreatedTime":1523984226.074,
         "AssociationVersion":"1"
      },
      {
         "Status":"Success",
         "DetailedStatus":"Success",
         "AssociationId":"c336d2ab-09de-44ba-8f6a-6136cEXAMPLE",
         "ExecutionId":"ecec60fa-6bb0-4d26-98c7-140308EXAMPLE",
         "CreatedTime":1523982404.013,
         "AssociationVersion":"1"
      }
   ]
```

```
}
```

PowerShell

```
AssociationId         : c336d2ab-09de-44ba-8f6a-6136cEXAMPLE
AssociationVersion    : 1
CreatedTime           : 8/18/2019 2:00:50 AM
DetailedStatus        : Success
ExecutionId           : 76a5a04f-caf6-490c-b448-92c02EXAMPLE
LastExecutionDate     : 1/1/0001 12:00:00 AM
ResourceCountByStatus : {Success=1}
Status                : Success

AssociationId         : c336d2ab-09de-44ba-8f6a-6136cEXAMPLE
AssociationVersion    : 1
CreatedTime           : 8/11/2019 2:00:54 AM
DetailedStatus        : Success
ExecutionId           : 791b72e0-f0da-4021-8b35-f95dfEXAMPLE
LastExecutionDate     : 1/1/0001 12:00:00 AM
ResourceCountByStatus : {Success=1}
Status                : Success

AssociationId         : c336d2ab-09de-44ba-8f6a-6136cEXAMPLE
AssociationVersion    : 1
CreatedTime           : 8/4/2019 2:01:00 AM
DetailedStatus        : Success
ExecutionId           : ecec60fa-6bb0-4d26-98c7-140308EXAMPLE
LastExecutionDate     : 1/1/0001 12:00:00 AM
ResourceCountByStatus : {Success=1}
Status                : Success
```

Sie können die Ergebnisse einschränken, indem Sie einen oder mehrere Filter verwenden. Das folgende Beispiel gibt alle Zuordnungen zurück, die vor einem bestimmten Datum und einer bestimmten Uhrzeit ausgeführt wurden.

Linux & macOS

```
aws ssm describe-association-executions \
  --association-id ID \
  --filters Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=LESS_THAN
```

Windows

```
aws ssm describe-association-executions ^
  --association-id ID ^
  --filters Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=LESS_THAN
```

PowerShell

```
Get-SSMAssociationExecution `
  -AssociationId 14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE `
  -Filter
 @{"Key"="CreatedTime";"Value"="2019-06-01T19:15:38.372Z";"Type"="LESS_THAN"}
```

Das folgende Beispiel gibt alle Zuordnungen zurück, die nach einem bestimmten Datum und einer bestimmten Uhrzeit erfolgreich ausgeführt wurden.

Linux & macOS

```
aws ssm describe-association-executions \
  --association-id ID \
  --filters Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=GREATER_THAN
 Key=Status,Value=Success,Type=EQUAL
```

Windows

```
aws ssm describe-association-executions ^
  --association-id ID ^
  --filters Key=CreatedTime,Value="2018-04-10T19:15:38.372Z",Type=GREATER_THAN
 Key=Status,Value=Success,Type=EQUAL
```

PowerShell

```
Get-SSMAssociationExecution `
  -AssociationId 14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE `
  -Filter @{
      "Key"="CreatedTime";
      "Value"="2019-06-01T19:15:38.372Z";
      "Type"="GREATER_THAN"
    },
```

```
    @{
      "Key"="Status";
      "Value"="Success";
      "Type"="EQUAL"
    }
```

3. Führen Sie den folgenden Befehl aus, um alle Ziele anzuzeigen, an denen die betreffende Ausführung ausgeführt wurde.

   Linux & macOS

   ```
   aws ssm describe-association-execution-targets \
     --association-id ID \
     --execution-id ID
   ```

   Windows

   ```
   aws ssm describe-association-execution-targets ^
     --association-id ID ^
     --execution-id ID
   ```

   PowerShell

   ```
   Get-SSMAssociationExecutionTarget `
     -AssociationId 14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE `
     -ExecutionId 76a5a04f-caf6-490c-b448-92c02EXAMPLE
   ```

   Sie können die Ergebnisse einschränken, indem Sie einen oder mehrere Filter verwenden. Das folgende Beispiel gibt Informationen über alle Ziele zurück, an denen die betreffende Zuordnung nicht ausgeführt werden konnte.

   Linux & macOS

   ```
   aws ssm describe-association-execution-targets \
     --association-id ID \
     --execution-id ID \
     --filters Key=Status,Value="Failed"
   ```

Windows

```
aws ssm describe-association-execution-targets ^
  --association-id ID ^
  --execution-id ID ^
  --filters Key=Status,Value="Failed"
```

PowerShell

```
Get-SSMAssociationExecutionTarget `
  -AssociationId 14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE `
  -ExecutionId 76a5a04f-caf6-490c-b448-92c02EXAMPLE `
  -Filter @{
      "Key"="Status";
      "Value"="Failed"
    }
```

Das folgende Beispiel gibt Informationen über einen bestimmten verwalteten Knoten zurück, bei dem eine Assoziation nicht ausgeführt werden konnte.

Linux & macOS

```
aws ssm describe-association-execution-targets \
  --association-id ID \
  --execution-id ID \
  --filters Key=Status,Value=Failed Key=ResourceId,Value="i-02573cafcfEXAMPLE"
 Key=ResourceType,Value=ManagedInstance
```

Windows

```
aws ssm describe-association-execution-targets ^
  --association-id ID ^
  --execution-id ID ^
  --filters Key=Status,Value=Failed Key=ResourceId,Value="i-02573cafcfEXAMPLE"
 Key=ResourceType,Value=ManagedInstance
```

PowerShell

```
Get-SSMAssociationExecutionTarget `
```

```
  -AssociationId 14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE `
  -ExecutionId 76a5a04f-caf6-490c-b448-92c02EXAMPLE `
  -Filter @{
      "Key"="Status";
      "Value"="Success"
    },
    @{
      "Key"="ResourceId";
      "Value"="i-02573cafcfEXAMPLE"
    },
    @{
      "Key"="ResourceType";
      "Value"="ManagedInstance"
    }
```

4. Wenn Sie eine Verknüpfung untersuchen, die nicht ausgeführt werden konnte, können Sie den StartAssociationsOnceAPI-Vorgang verwenden, um eine Verknüpfung sofort und nur einmal auszuführen. Nachdem Sie Änderungen an der Ressource vornehmen, bei der die Zuordnung nicht ausgeführt werden konnte, führen Sie den folgenden Befehl aus, um die Zuordnung sofort und nur einmal auszuführen.

   Linux & macOS

   ```
   aws ssm start-associations-once \
     --association-id ID
   ```

   Windows

   ```
   aws ssm start-associations-once ^
     --association-id ID
   ```

   PowerShell

   ```
   Start-SSMAssociationsOnce `
     -AssociationId ID
   ```

Arbeiten mit Zuordnungen mithilfe von IAM

State Manager, ein Tool in AWS Systems Manager, verwendet Ziele, um auszuwählen, mit welchen Instances Sie Ihre Verknüpfungen konfigurieren. Ursprünglich wurden Zuordnungen erstellt, indem

ein Dokumentname (`Name`) und Instance-ID (`InstanceId`) angegeben wurden. Dadurch wurde eine Zuordnung zwischen einem Dokument und einer Instance oder einem verwalteten Knoten erstellt. Zuordnungen wurden durch diese Parameter identifiziert. Diese Parameter sind jetzt veraltet, werden aber weiterhin unterstützt. Die Ressourcen `instance` und `managed-instance` wurden als Ressourcen zu Aktionen mit `Name` und `InstanceId` hinzugefügt.

AWS Identity and Access Management Das Verhalten bei der Durchsetzung von Richtlinien (IAM) hängt vom angegebenen Ressourcentyp ab. Ressourcen für State Manager-Vorgänge werden nur basierend auf der übergebenen Anforderung erzwungen. State Manager führt keine tiefe Prüfung auf die Eigenschaften von Ressourcen in Ihrem Konto durch. Eine Anforderung wird nur anhand von Richtlinienressourcen validiert, wenn der Anforderungsparameter die angegebenen Richtlinienressourcen enthält. Wenn Sie beispielsweise eine Instance im Ressourcenblock angeben, wird die Richtlinie erzwungen, wenn die Anforderung den `InstanceId`-Parameter verwendet. Der `Targets`-Parameter für jede Ressource im Konto wird nicht für diese `InstanceId` überprüft.

Im Folgenden sind einige Fälle mit verwirrendem Verhalten dargestellt:

- DescribeAssociationDeleteAssociation, und UpdateAssociationverwenden Sie`instance`,, und `document` Ressourcen`managed-instance`, um die veraltete Art des Verweises auf Assoziationen anzugeben. Dies beinhaltet alle Zuordnungen, die mit dem veralteten `InstanceId`-Parameter erstellt wurden.
- CreateAssociationCreateAssociationBatch, und UpdateAssociationverwenden Sie `instance` und `managed-instance` Ressourcen, um die veraltete Art der Bezugnahme auf Assoziationen zu spezifizieren. Dies beinhaltet alle Zuordnungen, die mit dem veralteten `InstanceId`-Parameter erstellt wurden. Der `document`-Ressourcentyp ist Teil der veralteten Methode, auf Zuordnungen zu verweisen und ist eine tatsächliche Eigenschaft einer Zuordnung. Das bedeutet, dass Sie IAM-Richtlinien mit den Berechtigungen `Allow` oder `Deny` für `Create`- und `Update`-Aktionen auf der Grundlage des Dokumentennamens erstellen können.

Weitere Informationen zur Verwendung von IAM-Richtlinien mit Systems Manager finden Sie unter Identity and Access Management für AWS Systems Manager oder Aktionen, Ressourcen und Bedingungsschlüssel für AWS Systems Manager in der Service Authorization-Referenz.

Erstellen von Zuordnungen, die MOF-Dateien ausführen

Sie können MOF-Dateien (Managed Object Format) ausführen, um mithilfe eines Tools in State Manager AWS Systems Manager, mithilfe des `AWS-ApplyDSCMofs` SSM-Dokuments einen Zielstatus auf Windows Server verwalteten Knoten zu erzwingen. Das `AWS-ApplyDSCMofs`-

Dokument weist zwei Ausführungsmodi auf. Mit dem ersten Modus können Sie die Assoziation so konfigurieren, dass sie die verwalteten Knoten scannt und meldet, wenn sie den in den MOF-Dateien definierten gewünschten Status aufweisen. Im zweiten Modus können Sie die MOF-Dateien ausführen und die Konfiguration Ihrer Knoten basierend auf den Ressourcen und ihren in den MOF-Dateien definierten Werten ändern. Mit dem `AWS-ApplyDSCMofs`-Dokument können Sie MOF-Konfigurationsdateien von Amazon Simple Storage Service (Amazon S3), einem lokal freigegebenen Verzeichnis, oder einer sicheren Website mit einer HTTPS-Domain herunterladen und ausführen.

State Manager protokolliert und meldet den Status der einzelnen MOF-Dateiausführungen bereits während die Zuordnungen ausgeführt werden. Darüber hinaus meldet State Manager die Ausgabe der MOF-Dateiausführungen als Compliance-Ereignis, das Sie auf der AWS Systems Manager Compliance-Seite anzeigen können.

Die Ausführung von MOF-Dateien basiert auf der Windows PowerShell Desired State Configuration (PowerShell DSC). PowerShell DSC ist eine deklarative Plattform, die für die Konfiguration, Bereitstellung und Verwaltung von Windows-Systemen verwendet wird. PowerShell DSC ermöglicht es Administratoren, in einfachen Textdokumenten, den sogenannten DSC-Konfigurationen, zu beschreiben, wie ein Server konfiguriert werden soll. Eine PowerShell DSC-Konfiguration ist ein spezielles PowerShell Skript, das angibt, was zu tun ist, aber nicht, wie es zu tun ist. Bei der Ausführung der Konfiguration wird eine MOF-Datei erzeugt. Die MOF-Datei kann auf einen oder mehrere Server angewendet werden, um die gewünschte Konfiguration für diese Server zu erreichen. PowerShell DSC-Ressourcen übernehmen die eigentliche Aufgabe, die Konfiguration durchzusetzen. Weitere Informationen finden Sie unter Übersicht über die Konfiguration des PowerShell gewünschten Windows-Zustands.

Themen

- Verwenden von Amazon S3 zum Speichern von Artefakten
- Auflösen von Anmeldeinformationen in MOF-Dateien
- Verwenden von Token in MOF-Dateien
- Voraussetzungen zum Erstellen von Zuordnungen, die MOF-Dateien ausführen
- Erstellen einer Zuordnung, die MOF-Dateien ausführt
- Problembehandlung bei der Erstellung von Zuordnungen, die MOF-Dateien ausführen
- Anzeigen von Details zur DSC-Ressourcen-Compliance

Verwenden von Amazon S3 zum Speichern von Artefakten

Wenn Sie Amazon S3 zum Speichern von PowerShell Modulen, MOF-Dateien, Compliance-Berichten oder Statusberichten verwenden, AWS Systems Manager SSM Agent müssen die von verwendete AWS Identity and Access Management (IAM) -Rolle `GetObject` und die `ListBucket` Berechtigungen für den Bucket vorhanden sein. Ohne diese Berechtigungen gibt das System einen Zugriff verweigert Fehler zurück. Unten finden Sie wichtige Informationen zum Speichern von Artefakten in Amazon S3.

- Wenn sich der Bucket in einem anderen Bucket befindet AWS-Konto, erstellen Sie eine Bucket-Ressourcenrichtlinie, die dem Konto (oder der IAM-Rolle) `GetObject` und Berechtigungen gewährt. `ListBucket`
- Wenn Sie benutzerdefinierte DSC-Ressourcen verwenden möchten, können Sie diese Ressourcen aus einem Amazon S3-Bucket herunterladen. Sie können sie auch automatisch aus der PowerShell Galerie installieren.
- Wenn Sie Amazon S3 als Modulquelle verwenden, laden Sie das Modul als Zip-Datei im folgenden Format mit Groß- und Kleinschreibung hoch: *`ModuleName`* _ *`ModuleVersion`* .zip. Zum Beispiel: _1.0.0.zip MyModule.
- Alle Dateien müssen im sich im Stammverzeichnis des Buckets befinden. Ordnerstrukturen werden nicht unterstützt.

Auflösen von Anmeldeinformationen in MOF-Dateien

Anmeldeinformationen werden mithilfe von AWS Secrets Manager oder AWS Systems Manager Parameter Store aufgelöst. Auf diese Weise können Sie eine automatische Rotation der Anmeldeinformationen einrichten. Auf diese Weise kann DSC auch Anmeldeinformationen automatisch an Ihre Server weitergeben, ohne sie erneut bereitstellen zu müssen. MOFs

Um ein AWS Secrets Manager Geheimnis in einer Konfiguration zu verwenden, erstellen Sie ein PSCredential Objekt, bei dem der Benutzername der SecretId oder SecretARN des Geheimnisses ist, das die Anmeldeinformationen enthält. Sie können für das Passwort einen beliebigen Wert angeben. Der Wert wird ignoriert. Im Folgenden sehen Sie ein Beispiel.

```
Configuration MyConfig
{
   $ss = ConvertTo-SecureString -String 'a_string' -AsPlaintext -Force
   $credential = New-Object PSCredential('a_secret_or_ARN', $ss)
```

```
    Node localhost
    {
       File file_name
       {
           DestinationPath = 'C:\MyFile.txt'
           SourcePath = '\\FileServer\Share\MyFile.txt'
           Credential = $credential
       }
    }
}
```

Kompilieren Sie Ihr MOF mithilfe der PsAllowPlaintextPassword Einstellung in den Konfigurationsdaten. Dies ist kein besonderes Risiko, weil die Anmeldeinformationen nur einen Bezeichner enthalten.

Stellen Sie in Secrets Manager sicher, dass der Knoten in einer von IAM verwalteten Richtlinie und optional in der Secret Resource Policy, falls vorhanden, GetSecretValue Zugriff hat. Für die Arbeit mit DSC muss das Geheimnis das folgende Format aufweisen.

```
{ 'Username': 'a_name', 'Password': 'a_password' }
```

Das Secret kann weitere Eigenschaften (z. B. Eigenschaften für die Rotation) haben, aber es muss mindestens den Benutzernamen und das Passwort enthalten.

Es wird empfohlen, eine Rotationsmethode für mehrere Benutzer zu verwenden, bei der Sie zwei verschiedene Benutzernamen und Kennwörter verwenden und die AWS Lambda Rotationsfunktion zwischen diesen hin und her wechselt. Diese Methode ermöglicht Ihnen, mehrere aktive Konten zu haben, ohne in Gefahr zu laufen, dass Benutzer bei einer Rotation ausgesperrt werden.

Verwenden von Token in MOF-Dateien

Token bieten Ihnen die Möglichkeit, Eigenschaftswerte von Ressourcen zu ändern, nachdem die MOF-Datei kompiliert wurde. Auf diese Weise können Sie häufig verwendete MOF-Dateien auf mehreren Servern mit ähnlichen Konfigurationen wiederverwenden.

Die Ersetzung der Token funktioniert nur für Ressourceneigenschaften des Typs `String`. Wenn Ihre Ressource jedoch eine eingebettete CIM-Knoten-Eigenschaft hat, löst sie auch Token von `String`-Eigenschaften in diesem CIM-Knoten auf. Sie können die Token-Ersetzung nicht für Zahlen oder Arrays verwenden.

Stellen Sie sich beispielsweise ein Szenario vor, in dem Sie die xComputerManagement Ressource verwenden und den Computer mithilfe von DSC umbenennen möchten. Normalerweise benötigen Sie eine dedizierte MOF-Datei für diesen Computer. Mit der Token-Unterstützung können Sie eine MOF-Datei erstellen und diese auf alle Ihre Knoten anwenden. Sie können in der `ComputerName`-Eigenschaft in der MOF-Datei anstelle des festkodierten Computernamens ein Token vom Typ Instance-Tag verwenden. Der Wert wird während beim Parsing der MOF-Datei aufgelöst. Sehen Sie sich das folgende Beispiel an.

```
Configuration MyConfig
{
    xComputer Computer
    {
        ComputerName = '{tag:ComputerName}'
    }
}
```

Anschließend legen Sie entweder ein Tag für den verwalteten Knoten in der Systems Manager Manager-Konsole oder ein Amazon Elastic Compute Cloud (Amazon EC2) -Tag in der EC2 Amazon-Konsole fest. Wenn Sie das Dokument ausführen, ersetzt das Skript den Wert des Instance-Tags durch das Token {tag:ComputerName}.

Sie können auch mehrere Tags in einer einzigen Eigenschaft kombinieren, wie im folgenden Beispiel gezeigt.

```
Configuration MyConfig
{
    File MyFile
    {
        DestinationPath = '{env:TMP}\{tag:ComputerName}'
        Type = 'Directory'
    }
}
```

Es gibt fünf verschiedene Arten von Token, die Sie verwenden können:

- Tag: Amazon EC2 - oder verwaltete Node-Tags.
- tagb64: Dies ist das gleiche wie tag, aber das System verwendet base64, um den Wert zu dekodieren. Auf diese Weise können Sie in Tag-Werten Sonderzeichen verwenden.
- env: Löst Umgebungsvariablen auf.

- ssm: Parameter Store-Werte. Es werden nur die Typen String und Secure String unterstützt.
- tagssm: Dies ist dasselbe wie Tag, aber wenn das Tag auf dem Knoten nicht festgelegt ist, versucht das System, den Wert aus einem Systems Manager-Parameter mit demselben Namen aufzulösen. Dies ist nützlich, wenn Sie einen „globalen Standardwert“ benötigen, den Sie auf einzelnen Knoten außer Kraft setzen möchten (z. B. bei One-Box-Bereitstellungen).

Es folgt ein Parameter Store-Beispiel, bei dem der Token-Typ `ssm` verwendet wird.

```
File MyFile
{
    DestinationPath = "C:\ProgramData\ConnectionData.txt"
    Content = "{ssm:%servicePath%/ConnectionData}"
}
```

Token spielen eine wichtige Rolle bei der Reduzierung von redundantem Code, weil MOF-Dateien generisch und wiederverwendbar werden. Wenn Sie serverspezifische MOF-Dateien vermeiden können, benötigen Sie auch keinen Service, um die MOF-Datei zu erstellen. Ein MOF-Building-Service erhöht die Kosten, verlangsamt die Bereitstellungszeit und erhöht das Risiko von Konfigurationsabweichungen zwischen gruppierten Knoten aufgrund unterschiedlicher Modulversionen, die bei der Kompilierung auf dem Build-Server installiert MOFs wurden.

Voraussetzungen zum Erstellen von Zuordnungen, die MOF-Dateien ausführen

Bevor Sie eine Assoziation erstellen, die MOF-Dateien ausführt, überprüfen Sie, ob Ihre verwalteten Knoten die folgenden Voraussetzungen erfüllen:

- Windows PowerShell Version 5.0 oder höher. Weitere Informationen finden Sie unter PowerShell Systemanforderungen für Windows auf Microsoft.com.
- AWS Tools for Windows PowerShell Version 3.3.261.0 oder höher
- SSM Agent, Version 2.2 oder höher.

Erstellen einer Zuordnung, die MOF-Dateien ausführt

So erstellen Sie eine Zuordnung, die MOF-Dateien ausführt

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich State Manager aus.

3. Wählen Sie State Manager und dann Zuordnung wählen aus.
4. Geben Sie im Feld Name einen Namen an. Dies ist zwar optional, wird aber empfohlen. Ein Name kann Ihnen helfen, den Zweck der Zuordnung zu verstehen, nachdem Sie sie erstellt haben. Der Name darf keine Leerzeichen enthalten.
5. Wählen Sie in der Liste Dokument die Option **`AWS-ApplyDSCMofs`** aus.
6. Geben Sie im Abschnitt Parameters (Parameter) die benötigten Angaben für die erforderlichen und die optionalen Eingabeparameter ein.

   a. Mofs To Apply (Anzuwendende MOF-Dateien): Geben Sie eine oder mehrere MOF-Dateien an, die mit dieser Zuordnung ausgeführt werden sollen. Um eine Liste von MOF-Dateien anzugeben, trennen Sie die Dateinamen mit Kommas. Systems Manager durchläuft die Liste der MOF-Dateien und führt sie in der Reihenfolge aus, die in der kommagetrennten Liste angegeben ist.

   - Eine Amazon S3-Bucket-Bezeichnung. Bucketnamen müssen Kleinbuchstaben angegeben werden. Geben Sie diese Informationen in dem folgenden Format an.

     ```
     s3:amzn-s3-demo-bucket:MOF_file_name.mof
     ```

     Wenn Sie ein angeben möchten AWS-Region, verwenden Sie das folgende Format.

     ```
     s3:bucket_Region:amzn-s3-demo-bucket:MOF_file_name.mof
     ```

   - Eine sichere Website. Geben Sie diese Informationen in dem folgenden Format an.

     ```
     https://domain_name/MOF_file_name.mof
     ```

     Ein Beispiel.

     ```
     https://www.example.com/TestMOF.mof
     ```

   - Ein Dateisystem auf einer lokalen Freigabe. Geben Sie diese Informationen in dem folgenden Format an.

     ```
     \server_name\shared_folder_name\MOF_file_name.mof
     ```

     Ein Beispiel.

```
\StateManagerAssociationsBox\MOFs_folder\MyMof.mof
```

b. Service Path (Service-Pfad): (Optional) Ein Service-Pfad ist entweder das Präfix eines Amazon S3-Buckets, in den Sie Berichte und Statusinformationen schreiben möchten, oder ein Service-Pfad ist ein Pfad für Parameter Store-Parameter-basierte Tags. Bei der Auflösung parameterbasierter Tags verwendet das System {ssm: %ServicePath %/*parameter_name*}, um den ServicePath-Wert in den Parameternamen einzufügen. Zum Beispiel, WebServers/Production" then the systems resolves the parameter as: WebServers/Production wenn *parameter_name* Ihr Dienstpfad "/ist. Dies ist nützlich, wenn Sie in einem Konto mehrere Umgebungen ausführen.

c. Report Bucket Name (Bucket-Name für Berichte): (Optional) Geben Sie den Namen eines Amazon S3-Buckets ein, in den Sie Compliance-Daten schreiben möchten. Berichte werden in diesem Bucket im JSON-Format gespeichert.

> **Note**
>
> Sie können dem Bucketnamen ein Präfix voranstellen, um die Region anzugeben, in der sich der Bucket befindet. Hier ist ein Beispiel: US-West-2:MYMOFBucket. Wenn Sie einen Proxy für Amazon S3-Endpunkte in einer bestimmten Region verwenden, die us-east-1 nicht enthält, stellen Sie dem Bucket-Namen eine Region voran. Wenn dem Bucketname kein Präfix vorangestellt ist, wird die Bucketregion unter Verwendung des Endpunkts us-east-1 automatisch erkannt.

d. Mof Operation Mode (MOF-Betriebsmodus): Wählen Sie das State Manager-Verhalten beim Ausführen der Zuordnung **`AWS-ApplyDSCMofs`** aus:

- Apply (Anwenden): Korrigiert Knoten-Konfigurationen, die nicht konform sind.
- ReportOnly: Korrigieren Sie keine Knotenkonfigurationen, sondern protokollieren Sie stattdessen alle Compliance-Daten und melden Sie Knoten, die nicht konform sind.

e. Status Bucket Name (Bucket-Name für Status): (Optional) Geben Sie den Namen eines Amazon S3-Buckets ein, in den Sie den MOF-Ausführungsstatus schreiben möchten. Diese Statusberichte sind Singleton-Zusammenfassungen des letzten Compliance-Laufs eines Knotens. Dies bedeutet, dass der Bericht überschrieben wird, wenn die Zuordnung das nächste Mal MOF-Dateien ausführt.

> **Note**
>
> Sie können dem Bucketnamen ein Präfix voranstellen, um die Region anzugeben, in der sich der Bucket befindet. Ein Beispiel: `us-west-2:amzn-s3-demo-bucket`. Wenn Sie einen Proxy für Amazon S3-Endpunkte in einer bestimmten Region verwenden, die us-east-1 nicht enthält, stellen Sie dem Bucket-Namen eine Region voran. Wenn dem Bucketname kein Präfix vorangestellt ist, wird die Bucketregion unter Verwendung des Endpunkts us-east-1 automatisch erkannt.

f. Name des Quell-Buckets für das Modul: (Optional) Geben Sie den Namen eines Amazon S3 S3-Buckets ein, der PowerShell Moduldateien enthält. Wenn Sie None (Keine) angeben, wählen Sie True (Wahr) für die nächste Option, Allow PS Gallery Module Source.

> **Note**
>
> Sie können dem Bucketnamen ein Präfix voranstellen, um die Region anzugeben, in der sich der Bucket befindet. Ein Beispiel: `us-west-2:amzn-s3-demo-bucket`. Wenn Sie einen Proxy für Amazon S3-Endpunkte in einer bestimmten Region verwenden, die us-east-1 nicht enthält, stellen Sie dem Bucket-Namen eine Region voran. Wenn dem Bucketname kein Präfix vorangestellt ist, wird die Bucketregion unter Verwendung des Endpunkts us-east-1 automatisch erkannt.

g. Quelle des PS Gallery-Moduls zulassen: (Optional) Wählen Sie True, um PowerShell Module von https://www.powershellgallery.com/herunterzuladen. Wenn Sie Falsch wählen, geben Sie eine Quelle für die vorherige Option an ModuleSourceBucketName.

h. Proxy-URI: (Optional) Verwenden Sie diese Option, um MOF-Dateien von einem Proxy-Server herunterzuladen.

i. Reboot Behavior (Neustart-Verhalten): (Optional) Geben Sie eine der folgenden Neustart-Verhaltensweisen an, wenn die Ausführung Ihrer MOF-Datei einen Neustart erfordert:

- AfterMof: Startet den Knoten neu, nachdem alle MOF-Ausführungen abgeschlossen sind. Selbst wenn mehrere MOF-Ausführungen einen Neustart anfordern, wartet das System mit dem Neustart, bis alle MOF-Ausführungen abgeschlossen sind.
- Immediately (Sofort): Startet den Knoten neu, sobald eine MOF-Ausführung dies anfordert. Wenn mehrere MOF-Dateien ausgeführt werden, die einen Neustart anfordern, wird der Knoten mehrmals neu gestartet.

- Never (Nie): Knoten werden selbst dann nicht neu gestartet, wenn die MOF-Ausführung explizit einen Neustart anfordert.

j. Use Computer Name For Reporting (Computername für Berichte verwenden): (Optional) Aktivieren Sie diese Option, um in gemeldeten Compliance-Informationen den Namen des Computers aufzuführen. Der Standardwert ist false (falsch). Das bedeutet, dass das System bei der Meldung von Compliance-Informationen die Knoten-ID verwendet.

k. Turn on Verbose Logging (Verbose-Protokollierung aktivieren): (Optional) Wir empfehlen, die Verbose-Protokollierung zu aktivieren, wenn Sie MOF-Dateien zum ersten Mal bereitstellen.

> **Important**
>
> Wenn diese Option aktiviert ist, schreibt die ausführliche Protokollierung mehr Daten in Ihren Amazon S3-Bucket als die Standardprotokollierung für die Zuordnungsausführung. Dies kann dazu führen, dass die Leistung beeinträchtigt wird und etwas höhere Speichergebühren für Amazon S3 anfallen. Um diese Probleme beim Speichern großer Datenvolumen abzumildern, empfehlen wir, die Lebenszyklusrichtlinien für Ihren Amazon S3-Bucket zu aktivieren. Weitere Informationen finden Sie unter Wie erstelle ich eine Lebenszyklus-Richtlinie für einen S3-Bucket? im Benutzerhandbuch zu Amazon Simple Storage Service.

l. Turn on Debug Logging (Debug-Protokollierung aktivieren): (Optional) Es wird empfohlen, die Debug-Protokollierung zu aktivieren, um MOF-Fehler zu beheben. Wir empfehlen außerdem, diese Option bei normaler Nutzung zu deaktivieren.

> **Important**
>
> Wenn diese Option aktiviert ist, schreibt die Debug-Protokollierung mehr Daten in Ihren Amazon S3-Bucket als die Standardprotokollierung für die Zuordnungsausführung. Dies kann dazu führen, dass die Leistung beeinträchtigt wird und etwas höhere Speichergebühren für Amazon S3 anfallen. Um diese Probleme beim Speichern großer Datenvolumen abzumildern, empfehlen wir, die Lebenszyklusrichtlinien für Ihren Amazon S3-Bucket zu aktivieren. Weitere Informationen finden Sie unter Wie erstelle ich eine Lebenszyklus-Richtlinie für einen S3-Bucket? im Benutzerhandbuch zu Amazon Simple Storage Service.

m. Compliance Type (Compliance-Typ): (Optional) Geben Sie den Compliance-Typ für die Meldung von Compliance-Informationen an. Der Standard-Compliance-Typ lautet Custom:DSC. Wenn Sie mehrere Zuordnungen erstellen, die MOF-Dateien ausführen, müssen Sie für jede Zuordnung einen anderen Compliance-Typ angeben. Wenn Sie dies nicht tun, überschreiben die zusätzlichen Zuordnungen mit Custom:DSC jeweils die bereits zusammengestellten Compliance-Daten.

n. Pre Reboot Script (Skript vor dem Neustart): (Optional) Geben Sie ein Skript an, das ausgeführt werden soll, wenn die Konfiguration einen Neustart anfordert. Das Skript wird vor dem Neustart ausgeführt. Das Skript muss aus einer einzelnen Zeile bestehen. Trennen Sie zusätzliche Zeilen mithilfe von Semikolons.

7. Wählen Sie im Abschnitt Targets (Ziele) entweder Specifying tags (Angaben von Tags) oder Manually Selecting Instance (Manuelles Auswählen einer Instance) aus. Wenn Sie die Zielressourcen mithilfe von Tags ausgewählt haben, geben Sie einen Tag-Schlüssel und den Tag-Wert in die entsprechenden Felder ein. Weitere Informationen zur Verwendung von Zielen finden Sie unter Verstehen von Zielen und Ratensteuerungen in State Manager Zuordnungen.

8. Wählen Sie im Abschnitt Zeitplan angeben entweder Nach Zeitplan oder Kein Zeitplan aus. Wenn Sie On Schedule (Nach Zeitplan) auswählen, verwenden Sie die verfügbaren Schaltflächen zum Erstellen eines cron- oder rate-Zeitplans für die Zuordnung.

9. Führen Sie im Abschnitt Advanced options (Erweiterte Optionen) Folgendes durch:

   - Wählen Sie unter Compliance severity (Compliance -Schweregrad), einen Schweregrad für die Zuordnung aus. In den Compliance-Berichten finden Sie Informationen dazu, ob die Zuordnung konform ist, zusammen mit dem Schweregrad, den Sie hier angeben. Weitere Informationen finden Sie unter Informationen zu State Manager-Zuordnungs-Compliance.

10. Konfigurieren Sie im Abschnitt Rate control (Ratensteuerung) Optionen für die Ausführung von State Manager-Zuordnungen in der Flotte der verwalteten Knoten. Weitere Informationen zu diesen Optionen finden Sie unter Verstehen von Zielen und Ratensteuerungen in State Manager Zuordnungen.

    Wählen Sie im Abschnitt Gleichzeitigkeit eine Option aus:

    - Wählen Sie Ziele aus, um eine absolute Anzahl von Zielen einzugeben, die die Zuordnung gleichzeitig ausführen können.
    - Wählen Sie Prozentsatz aus, um einen Prozentsatz der Ziele anzugeben, die die Zuordnung gleichzeitig ausführen können.

Wählen Sie im Abschnitt Fehlerschwellenwert eine Option aus:

- Wählen Sie Errors (Fehler) aus und geben Sie die absolute Anzahl erlaubter Fehler an, bis State Manager die Ausführung von Zuordnungen für weitere Ziele beendet.
- Wählen Sie Percentage (Prozentsatz) aus und geben Sie den Prozentsatz erlaubter Fehler an, bis State Manager die Ausführung von Zuordnungen für weitere Ziele beendet.

11. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

12. Wählen Sie Zuordnung erstellen.

State Manager erstellt die Assoziation und führt sie sofort auf den angegebenen Knoten aus. Nach der ersten Ausführung wird die Zuordnung gemäß dem festgelegten Zeitplan und entsprechend den folgenden Regeln in Intervallen ausgeführt:

- State Manager führt Assoziationen für Knoten aus, die online sind, wenn das Intervall gestartet wird und überspringt Offline-Knoten.
- State Manager versucht, die Assoziation während eines Intervalls auf allen konfigurierten Knoten auszuführen.
- Wenn eine Assoziation in einem Intervall nicht ausgeführt wird (weil beispielsweise die Anzahl der Knoten, die die Assoziation gleichzeitig ausführen können, durch einen Gleichzeitigkeitswert begrenzt wird), versucht State Manager, die Assoziation im nächsten Intervall auszuführen.

- State Manager zeichnet einen Verlauf für alle übersprungenen Datensätze an. Sie können den Verlauf auf der Registerkarte Execution History (Ausführungsverlauf) anzeigen.

> **Note**
>
> Das `AWS-ApplyDSCMofs` ist ein Systems Manager Befehlsdokument. Dies bedeutet, dass dieses Dokument auch mit Run Command ausgeführt werden kann, einem Tool von AWS Systems Manager. Weitere Informationen finden Sie unter AWS Systems Manager Run Command.

Problembehandlung bei der Erstellung von Zuordnungen, die MOF-Dateien ausführen

Dieser Abschnitt enthält Informationen zur Unterstützung bei der Behebung von Problemen, die möglicherweise beim Erstellen von Zuordnungen zur Ausführung von MOF-Dateien auftreten.

Aktivieren der erweiterten Protokollierung

Aktivieren Sie als ersten Schritt zur Fehlerbehebung die erweiterte Protokollierung. Führen Sie dazu die folgenden Schritte aus:

1. Stellen Sie sicher, dass die Zuordnung so konfiguriert ist, dass sie Befehlsausgaben entweder in Amazon S3 oder Amazon CloudWatch Logs (CloudWatch) schreibt.
2. Setzen Sie den Parameter Enable Verbose Logging auf „True“ fest.
3. Setzen Sie den Parameter Enable Debug Logging auf „True“ fest.

Wenn die Verbose- und die Debug-Protokollierung aktiviert ist, enthält die Stdout-Ausgabedatei Details über die Skriptausführung. Diese Ausgabedatei kann Sie bei der Suche, an welcher Stelle das Skript fehlgeschlagen ist, unterstützen. Die Stderr-Ausgabedatei enthält Fehler, die während der Skriptausführung aufgetreten sind.

Häufige Probleme beim Erstellen von Zuordnungen, die MOF-Dateien ausführen

Dieser Abschnitt enthält Informationen über häufige Probleme, die beim Erstellen von Zuordnungen für die Ausführung von MOF-Dateien auftreten können, sowie Schritte, um diese Probleme zu beheben.

Meine MOF-Datei wurde nicht angewendet.

Wenn State Manager die Assoziation nicht auf Ihre Knoten anwenden konnte, überprüfen Sie zunächst die Stderr-Ausgabedatei. Diese Datei kann Ihnen helfen, die Ursache des Problems zu verstehen. Überprüfen Sie auch Folgendes:

- Der Knoten hat die erforderlichen Zugriffsberechtigungen für alle mit der MOF-Datei verbundenen Amazon S3-Buckets. Das heißt:
  - s3: GetObject Berechtigungen: Dies ist für MOF-Dateien in privaten Amazon S3 S3-Buckets und benutzerdefinierte Module in Amazon S3 S3-Buckets erforderlich.
  - s3: PutObject Berechtigung: Dies ist erforderlich, um Compliance-Berichte und den Compliance-Status in Amazon S3 S3-Buckets zu schreiben.
- Wenn Sie Tags verwenden, stellen Sie sicher, dass der Knoten die erforderliche IAM-Richtlinie hat. Die Verwendung von Tags erfordert, dass die Instance-IAM-Rolle über eine Richtlinie verfügt, die die Aktionen `ec2:DescribeInstances` und `ssm:ListTagsForResource` ermöglicht.
- Stellen Sie sicher, dass dem Knoten die erwarteten Tags oder SSM-Parameter zugewiesen wurden.
- Stellen Sie sicher, dass alle Tags und SSM-Parameter richtig geschrieben sind.
- Versuchen Sie, die MOF-Datei lokal auf dem Knoten auszuführen, um sicherzustellen, dass kein Problem bei der MOF-Datei selbst vorliegt.

Meine MOF-Datei schien fehlzuschlagen, aber die Systems Manager-Ausführung war erfolgreich.

Wenn das Dokument `AWS-ApplyDSCMofs` erfolgreich ausgeführt wurde, wird der Systems Manager-Ausführungsstatus als Success (Erfolg) angezeigt. Dieser Status sagt nichts über den Compliance-Status Ihres Knotens, gemessen an den Konfigurationsanforderungen in der MOF-Datei, aus. Um den Compliance-Status Ihrer Knoten anzuzeigen, zeigen Sie die Compliance-Berichte an. Sie können einen JSON-Bericht im Amazon S3-Bericht-Bucket anzeigen. Dies gilt nur für Run Command- und für State Manager-Ausführungen. Darüber hinaus können Sie Compliance-Details für State Manager auf der Systems Manager-Compliance-Seite anzeigen.

In der Stderr-Ausgabedatei finden sich Hinweise, dass bei dem Versuch, den Service zu erreichen, ein Fehler bei der Namensauflösung aufgetreten ist.

Dieser Fehler weist darauf hin, dass das Skript einen Remoteservice nicht erreichen kann. In den meisten Fällen dürfte das Skript Probleme haben, Amazon S3 zu erreichen. Dieses Problem tritt am häufigsten auf, wenn das Skript versucht, Compliance-Berichte oder den Compliance-Status in den Amazon S3-Bucket zu schreiben, der in den Dokumentparametern angegeben ist. In der Regel tritt

dieser Fehler auf, wenn eine Datenverarbeitungsumgebung eine Firewall oder einen transparenten Proxy mit einer Zulassungsliste verwendet. So beheben Sie dieses Problem

- Verwenden Sie die regionsspezifische Bucket-Syntax für alle Amazon S3-Bucket-Parameter. Beispiel: Die Mofs to Apply-Parameter sollten in dem folgenden Format angegeben werden:

  s3:: *bucket-regionamzn-s3-demo-bucket*: *mof-file-name* .mof.

  Ein Beispiel: `s3:us-west-2:amzn-s3-demo-bucket:my-mof.mof`

  Die Bucketnamen für Berichte, Statusinformationen und Modulquellen sollten in dem folgenden Format angegeben werden.

  bucket-region:. *amzn-s3-demo-bucket* Hier ist ein Beispiel: `us-west-1:amzn-s3-demo-bucket;`

- Wenn sich das Problem nicht durch Verwendung einer regionsspezifischen Syntax beheben lässt, stellen Sie sicher, dass die Ziel-Knoten in der gewünschten Region auf Simple Storage Service (Amazon S3) zugreifen können. So können Sie dies überprüfen

  1. Suchen Sie den Endpunktnamen für Amazon S3 in der entsprechenden Amazon S3-Region. Weitere Informationen finden Sie unter Amazon-S3-Service-Endpunkte im Allgemeine Amazon Web Services-Referenz.
  2. Melden Sie sich am Ziel-Knoten an und führen Sie den folgenden Ping-Befehl aus.

     ```
     ping s3.s3-region.amazonaws.com
     ```

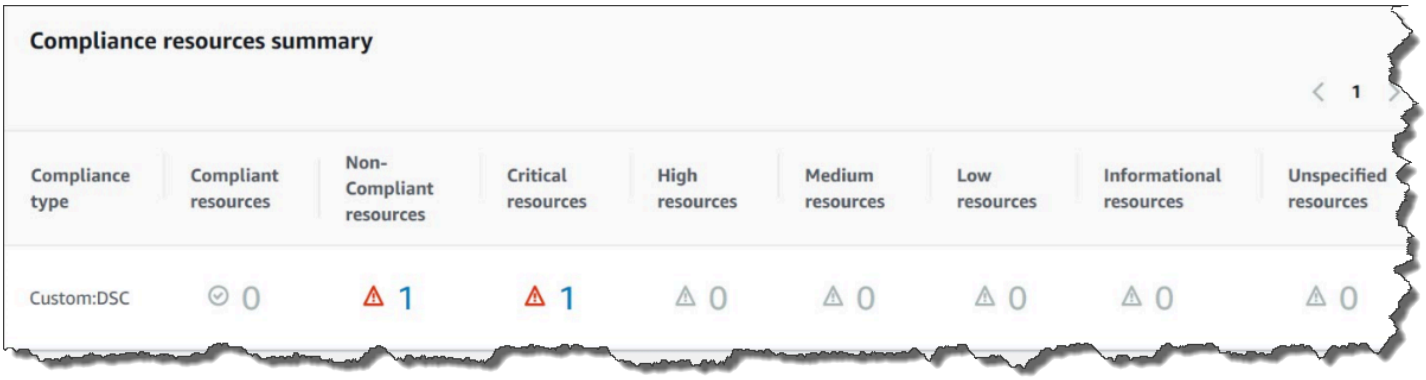
     Wenn der Ping fehlschlägt, bedeutet dies, dass entweder Amazon S3 ausgefallen ist oder ein firewall/transparent Proxy den Zugriff auf die Amazon S3 S3-Region blockiert oder der Knoten nicht auf das Internet zugreifen kann.

Anzeigen von Details zur DSC-Ressourcen-Compliance

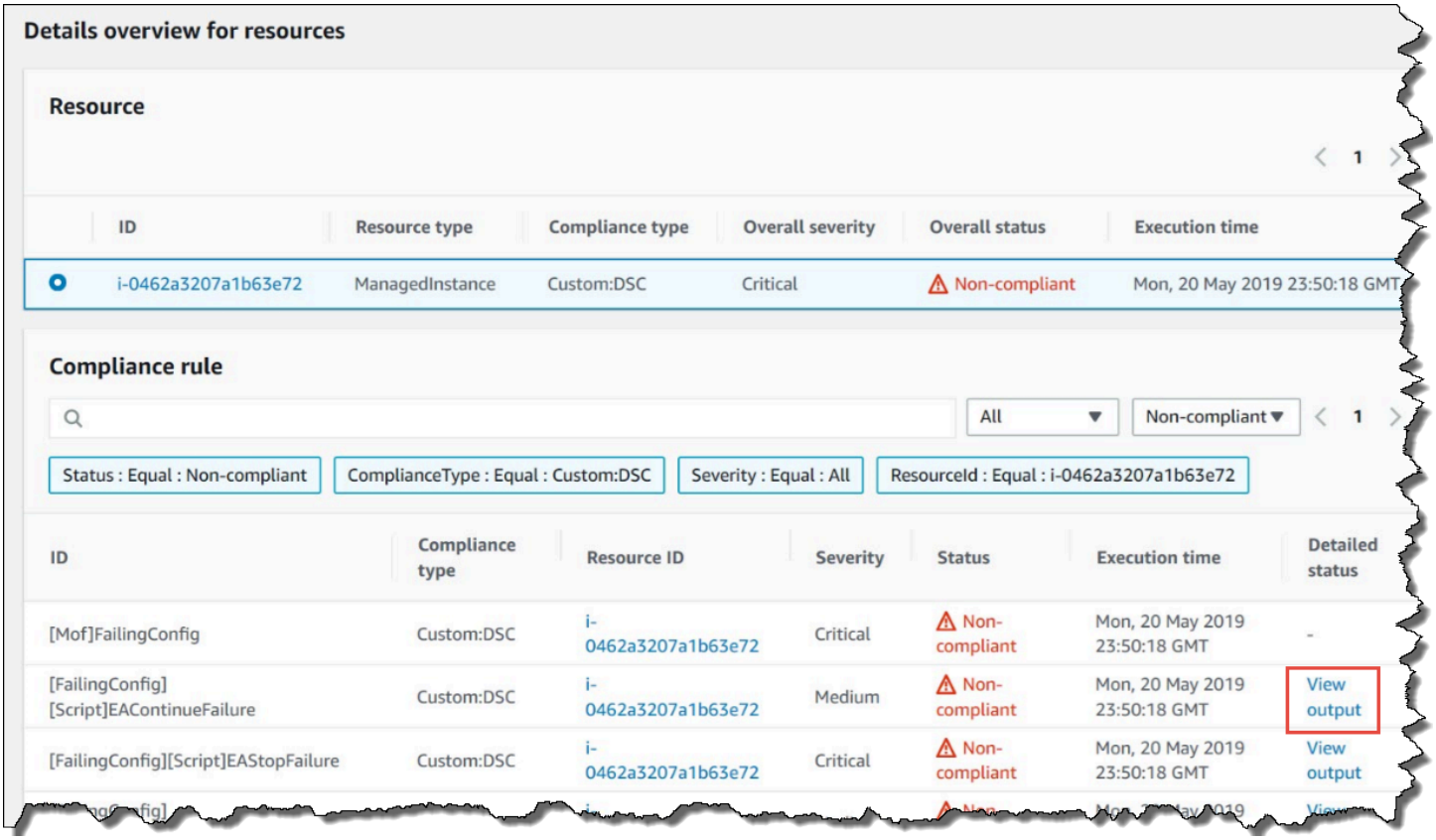
Systems Manager erfasst Compliance-Informationen zu DSC-Ressourcenfehlern im Amazon S3 Status-Bucket, den Sie bei der Ausführung des `AWS-ApplyDSCMofs`-Dokuments angegeben haben. Die Suche nach Informationen zu DSC-Ressourcenfehlern in einem Amazon S3-Bucket kann zeitaufwendig sein. Stattdessen können Sie diese Informationen auf der Systems Manager-Seite Compliance anzeigen.

Der Bereich Compliance-Ressourcen-Zusammenfassung zeigt die Anzahl der Ressourcen an, die fehlgeschlagen sind. Im folgenden Beispiel ComplianceTypeist das custom:DSC und eine Ressource ist nicht konform.

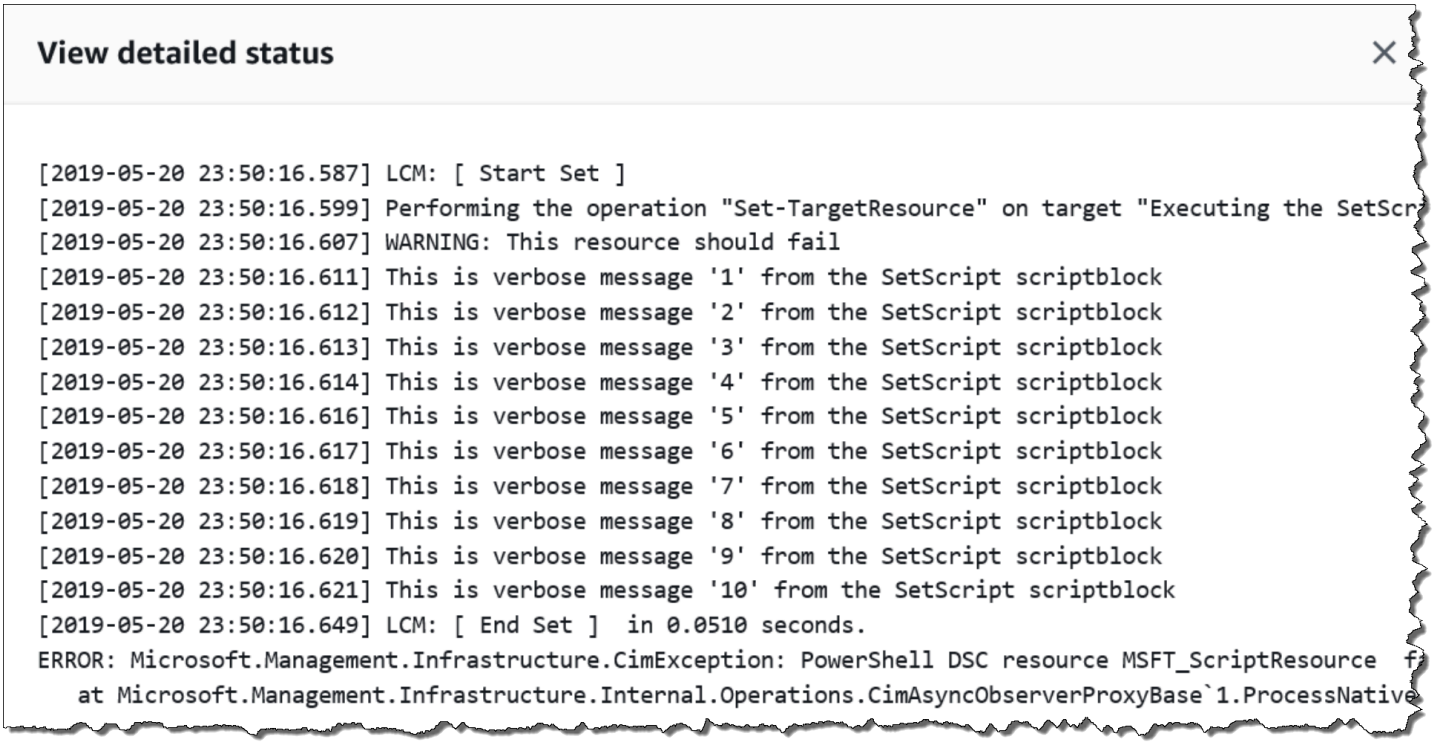
> **Note**
>
> custom:DSC ist der Standardwert im Dokument. ComplianceType`AWS-ApplyDSCMofs` Dieser Wert ist anpassbar.



Im Abschnitt „Detailübersicht für Ressourcen“ werden Informationen über die AWS Ressource mit der nicht konformen DSC-Ressource angezeigt. Dieser Bereich enthält auch den MOF-Namen, Skript-Ausführungsschritte und (falls zutreffend) den Link View output (Ausgabe anzeigen) zur Ansicht detaillierter Statusinformationen.



Der Link View output zeigt die letzten 4.000 Zeichen des detaillierten Status an. Systems Manager beginnt mit der Ausnahme als erstem Element und scannt dann durch die ausführlichen Nachrichten und stellt so viele wie möglich voran, bis das Kontingent von 4.000 Zeichen erreicht wird. Dieser Vorgang zeigt die Protokollmeldungen an, die vor dem Auslösen der Ausnahme ausgegeben wurden. Dabei handelt es sich um die relevantesten Nachrichten für die Fehlerbehebung.



Weitere Informationen zum Anzeigen von Compliance-Informationen finden Sie unter AWS Systems Manager-Compliance.

Situationen, die die Compliance-Berichterstellung beeinflussen

Wenn die State Manager-Zuordnung fehlschlägt, werden keine Compliance-Daten gemeldet. Genauer gesagt, meldet Systems Manager doesn't keine Compliance-Elemente, wenn eine MOF-Datei nicht verarbeitet werden kann, da die Zuordnungen fehlschlagen. Beispiel: Wenn Systems Manager versucht, eine MOF-Datei von einem Amazon S3-Bucket herunterzuladen, und der Knoten keine Berechtigung zum Zugriff besitzt, schlägt die Zuordnung fehl und es werden keine Compliance-Daten gemeldet.

Wenn eine Ressource in einer zweiten MOF-Datei fehlschlägt, dann meldet Systems Manager Bericht-Compliance-Daten. Beispiel: Wenn ein MOF versucht, eine Datei auf einem nicht vorhandenen Laufwerk zu erstellen, dann meldet Systems Manager Compliance, da das `AWS-ApplyDSCMofs`-Dokument vollständig verarbeitet werden kann. Dies bedeutet, dass die Zuordnung erfolgreich ausgeführt wird.

Erstellen von Zuordnungen, die Ansible-Playbooks ausführen

Sie können State Manager-Zuordnungen erstellen, die Ansible-Playbooks ausführen, indem Sie das `AWS-ApplyAnsiblePlaybooks`-SSM-Dokument verwenden. State Manager ist ein Tool von AWS Systems Manager. Dieses Dokument bietet die folgenden Vorteile für die Ausführung von Playbooks:

- Unterstützung für die Ausführung komplexer Playbooks
- Support für das Herunterladen von Playbooks von GitHub und Amazon Simple Storage Service (Amazon S3)
- Unterstützung der komprimierten Playbook-Struktur
- Erweiterte Protokollierung
- Möglichkeit, anzugeben, welches Playbook ausgeführt werden soll, wenn Playbooks gebündelt werden

> **Note**
>
> Systems Manager enthält zwei SSM-Dokumente, mit denen Sie State Manager-Zuordnungen erstellen können, die Ansible-Playbooks ausführen: `AWS-RunAnsiblePlaybook` und `AWS-ApplyAnsiblePlaybooks`. Das `AWS-RunAnsiblePlaybook`-Dokument ist veraltet. Es bleibt für Legacy-Zwecke in Systems Manager verfügbar. Wir empfehlen, dass Sie das `AWS-ApplyAnsiblePlaybooks`-Dokument aufgrund der hier beschriebenen Verbesserungen verwenden.
>
> Zuordnungen, die Ansible-Playbooks ausführen, werden in macOS nicht unterstützt.

Unterstützung für die Ausführung komplexer Playbooks

Das `AWS-ApplyAnsiblePlaybooks`-Dokument unterstützt gebündelte, komplexe Playbooks, da es die gesamte Dateistruktur vor der Ausführung des angegebenen Haupt-Playbooks in ein lokales Verzeichnis kopiert. Sie können Quell-Playbooks in Zip-Dateien oder in einer Verzeichnisstruktur bereitstellen. Die Zip-Datei oder das Verzeichnis kann in GitHub oder Amazon S3 gespeichert werden.

Unterstützung für das Herunterladen von Playbooks von GitHub

Das `AWS-ApplyAnsiblePlaybooks`-Dokument verwendet das `aws:downloadContent`-Plugin zum Herunterladen von Playbook-Dateien. Dateien können in GitHub in einer einzigen Datei oder als

kombinierte Gruppe von Playbook-Dateien gespeichert werden. Geben Sie Informationen über Ihr GitHub-Repository im JSON-Format an, um Inhalte von GitHub herunterzuladen. Ein Beispiel.

```
{
   "owner":"TestUser",
   "repository":"GitHubTest",
   "path":"scripts/python/test-script",
   "getOptions":"branch:master",
   "tokenInfo":"{{ssm-secure:secure-string-token}}"
}
```

Unterstützung für das Herunterladen von Playbooks von Amazon S3

Sie können Ansible-Playbooks auch in Amazon S3 als einzelne ZIP-Datei oder als Verzeichnisstruktur speichern und herunterladen. Geben Sie den Pfad zur Datei an, um Inhalte von Amazon S3 herunterzuladen. Nachfolgend finden Sie zwei Beispiele.

Beispiel 1: Herunterladen einer bestimmten Playbook-Datei

```
{
   "path":"https://s3.amazonaws.com/amzn-s3-demo-bucket/playbook.yml"
}
```

Beispiel 2: Herunterladen des Inhalts eines Verzeichnisses

```
{
   "path":"https://s3.amazonaws.com/amzn-s3-demo-bucket/ansible/webservers/"
}
```

Important

Wenn Sie Amazon S3 angeben, muss das AWS Identity and Access Management (IAM) - Instance-Profil auf Ihren verwalteten Knoten Berechtigungen für den S3-Bucket enthalten. Weitere Informationen finden Sie unter Erforderliche Instance-Berechtigungen für Systems Manager konfigurieren.

Unterstützung der komprimierten Playbook-Struktur

Mit dem AWS-ApplyAnsiblePlaybooks-Dokument können Sie komprimierte ZIP-Dateien im heruntergeladenen Paket ausführen. Das Dokument prüft, ob die heruntergeladenen Dateien eine komprimierte Datei im ZIP-Format enthalten. Wenn eine ZIP-Datei gefunden wird, dekomprimiert das Dokument die Datei automatisch und führt dann die angegebene Ansible-Automatisierung aus.

Erweiterte Protokollierung

Das AWS-ApplyAnsiblePlaybooks-Dokument enthält einen optionalen Parameter für die Angabe verschiedener Protokollierungsebenen. Geben Sie -v für niedrige Ausführlichkeit, -vv oder -vvv für mittlere Ausführlichkeit und -vvvv für die Protokollierung auf Debug-Ebene an. Diese Optionen werden direkt den Ansible-Ausführlichkeitsoptionen zugeordnet.

Möglichkeit, anzugeben, welches Playbook ausgeführt werden soll, wenn Playbooks gebündelt werden

Das AWS-ApplyAnsiblePlaybooks-Dokument enthält einen erforderlichen Parameter, um anzugeben, welches Playbook ausgeführt werden soll, wenn mehrere Playbooks gebündelt werden. Diese Option bietet Flexibilität für die Ausführung von Playbooks, um verschiedene Anwendungsfälle zu unterstützen.

Grundlegendes zu installierten Abhängigkeiten

Wenn Sie True für den InstallDependenciesParameter angeben, überprüft Systems Manager, ob auf Ihren Knoten die folgenden Abhängigkeiten installiert sind:

- Ubuntu Server/Debian Server: Apt-get (Paketverwaltung), Python 3, Ansible, Unzip
- Von Amazon Linux unterstützte Versionen: Ansible
- RHEL: Python 3, Ansible, Entpacken

Wenn eine oder mehrere dieser Abhängigkeiten nicht gefunden werden, installiert Systems Manager sie automatisch.

Erstellen einer Zuordnung zur Ausführung von Ansible-Playbooks (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie mit der Systems-Manager-Konsole eine State Manager-Zuordnung erstellen, die Ansible-Playbooks mithilfe des AWS-ApplyAnsiblePlaybooks-Dokuments ausführt.

So erstellen Sie eine Zuordnung, die Ansible-Playbooks ausführt (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich State Manager aus.
3. Wählen Sie State Manager und dann Zuordnung wählen aus.
4. Geben Sie unter Name einen Namen an, der Ihnen hilft, sich an den Zweck der Zuordnung zu erinnern.
5. Wählen Sie in der Liste Dokument die Option **AWS-ApplyAnsiblePlaybooks** aus.
6. Wählen Sie im Abschnitt Parameter für Source Type entweder GitHuboder S3 aus.

   GitHub

   Wenn Sie GitHub auswählen, geben Sie Repository-Informationen im folgenden Format ein.

   ```
   {
      "owner":"user_name",
      "repository":"name",
      "path":"path_to_directory_or_playbook_to_download",
      "getOptions":"branch:branch_name",
      "tokenInfo":"{{(Optional)_token_information}}"
   }
   ```

   S3

   Wenn Sie S3 auswählen, geben Sie Pfadinformationen im folgenden Format ein.

   ```
   {
      "path":"https://s3.amazonaws.com/path_to_directory_or_playbook_to_download"
   }
   ```
7. Wählen Sie unter Install Dependencies (Abhängigkeiten installieren) eine Option aus.
8. (Optional) Geben Sie unter Playbook File (Playbook-Datei) einen Dateinamen ein. Wenn eine Zip-Datei das Playbook enthält, geben Sie einen relativen Pfad zur Zip-Datei an.
9. (Optional) Geben Sie unter Zusätzliche Variablen Variablen ein, die State Manager zur Laufzeit an Ansible senden soll.
10. (Optional) Wählen Sie unter Check (Prüfen) eine Option aus.
11. (Optional) Wählen Sie für Verbose eine Option aus.

12. Wählen Sie für Ziele eine Option aus. Weitere Informationen zur Verwendung von Zielen finden Sie unter Verstehen von Zielen und Ratensteuerungen in State Manager Zuordnungen.
13. Wählen Sie im Abschnitt Specify schedule (Zeitplan angeben) entweder On schedule (Nach Zeitplan) oder No schedule (Kein Zeitplan) aus. Wenn Sie On schedule (Nach Zeitplan) auswählen, verwenden Sie die verfügbaren Schaltflächen zum Erstellen eines cron- oder rate-Zeitplans für die Zuordnung.
14. Wählen Sie im Abschnitt Advanced options (Erweiterte Optionen) für Compliance severity (Compliance-Schweregrad) einen Schweregrad für die Zuordnung aus. In den Compliance-Berichten finden Sie Informationen dazu, ob die Zuordnung konform ist, zusammen mit dem Schweregrad, den Sie hier angeben. Weitere Informationen finden Sie unter Informationen zu State Manager-Zuordnungs-Compliance.
15. Konfigurieren Sie im Abschnitt Rate control (Ratensteuerung) Optionen für die Ausführung von State Manager-Zuordnungen in der Flotte von verwalteten Knoten. Weitere Informationen über Ratensteuerungen finden Sie unter Verstehen von Zielen und Ratensteuerungen in State Manager Zuordnungen.

    Wählen Sie im Abschnitt Gleichzeitigkeit eine Option aus:

    - Wählen Sie Ziele aus, um eine absolute Anzahl von Zielen einzugeben, die die Zuordnung gleichzeitig ausführen können.
    - Wählen Sie Prozentsatz aus, um einen Prozentsatz der Ziele anzugeben, die die Zuordnung gleichzeitig ausführen können.

    Wählen Sie im Abschnitt Fehlerschwellenwert eine Option aus:

    - Wählen Sie Fehler aus und geben Sie die absolute Anzahl erlaubter Fehler an, bis State Manager die Ausführung von Zuordnungen für weitere Ziele beendet.
    - Wählen Sie Prozentsatz aus und geben Sie den Prozentsatz erlaubter Fehler an, bis State Manager die Ausführung von Zuordnungen für weitere Ziele beendet.
16. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

17. Wählen Sie Zuordnung erstellen.

> **Note**
>
> Wenn Sie auf einer oder mehreren Knoten eine Assoziation anhand von Tags erstellen und von einem dieser Knoten die Tags entfernen, wird die Assoziation auf diesem Knoten nicht mehr ausgeführt. Die Assoziazion zwischen dem Knoten und dem State Manager-Dokument ist aufgehoben.

Erstellen einer Zuordnung zur Ausführung von Ansible-Playbooks (CLI)

Im folgenden Verfahren wird beschrieben, wie Sie mit AWS Command Line Interface (AWS CLI) eine State Manager Assoziation erstellen, die Ansible Playbooks mithilfe des `AWS-ApplyAnsiblePlaybooks` Dokuments ausführt.

So erstellen Sie eine Zuordnung, die Ansible-Playbooks ausführt (CLI)

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie einen der folgenden Befehle aus, um eine Zuordnung zu erstellen, die Ansible-Playbooks ausführt, indem Knoten mithilfe von Tags angesprochen werden. Ersetzen Sie

jeden *example resource placeholder* durch Ihre Informationen. Befehl (A) gibt GitHub als Quelltyp an. Befehl (B) gibt Amazon S3 als Quelltyp an.

(Eine) GitHub Quelle

Linux & macOS

```
aws ssm create-association --name "AWS-ApplyAnsiblePlaybooks" \
    --targets Key=tag:TagKey,Values=TagValue \
    --parameters '{"SourceType":["GitHub"],"SourceInfo":
["{\"owner\":\"owner_name\", \"repository\": \"name\",
 \"getOptions\": \"branch:master\"}"],"InstallDependencies":
["True_or_False"],"PlaybookFile":["file_name.yml"],"ExtraVariables":["key/
value_pairs_separated_by_a_space"],"Check":["True_or_False"],"Verbose":["-v,-
vv,-vvv, or -vvvv"],"TimeoutSeconds":["3600"]}' \
    --association-name "name" \
    --schedule-expression "cron_or_rate_expression"
```

Windows

```
aws ssm create-association --name "AWS-ApplyAnsiblePlaybooks" ^
    --targets Key=tag:TagKey,Values=TagValue ^
    --parameters '{"SourceType":["GitHub"],"SourceInfo":
["{\"owner\":\"owner_name\", \"repository\": \"name\",
 \"getOptions\": \"branch:master\"}"],"InstallDependencies":
["True_or_False"],"PlaybookFile":["file_name.yml"],"ExtraVariables":["key/
value_pairs_separated_by_a_space"],"Check":["True_or_False"],"Verbose":["-v,-
vv,-vvv, or -vvvv"], "TimeoutSeconds":["3600"]}' ^
    --association-name "name" ^
    --schedule-expression "cron_or_rate_expression"
```

Ein Beispiel.

```
aws ssm create-association --name "AWS-ApplyAnsiblePlaybooks" \
    --targets "Key=tag:OS,Values=Linux" \
    --parameters '{"SourceType":["GitHub"],"SourceInfo":["{\"owner\":
\"ansibleDocumentTest\", \"repository\": \"Ansible\", \"getOptions\":
 \"branch:master\"}"],"InstallDependencies":["True"],"PlaybookFile":["hello-world-
playbook.yml"],"ExtraVariables":["SSM=True"],"Check":["False"],"Verbose":["-v"]}' \
    --association-name "AnsibleAssociation" \
```

```
    --schedule-expression "cron(0 2 ? * SUN *)"
```

(B) S3-Quelle

Linux & macOS

```
aws ssm create-association --name "AWS-ApplyAnsiblePlaybooks" \
    --targets Key=tag:TagKey,Values=TagValue \
    --parameters '{"SourceType":["S3"],"SourceInfo":["{\"path\":\"https://
s3.amazonaws.com/
path_to_Zip_file,_directory,_or_playbook_to_download\"}"],"InstallDependencies":
["True_or_False"],"PlaybookFile":["file_name.yml"],"ExtraVariables":["key/
value_pairs_separated_by_a_space"],"Check":["True_or_False"],"Verbose":["-v,-
vv,-vvv, or -vvvv"]}' \
    --association-name "name" \
    --schedule-expression "cron_or_rate_expression"
```

Windows

```
aws ssm create-association --name "AWS-ApplyAnsiblePlaybooks" ^
    --targets Key=tag:TagKey,Values=TagValue ^
    --parameters '{"SourceType":["S3"],"SourceInfo":["{\"path\":\"https://
s3.amazonaws.com/
path_to_Zip_file,_directory,_or_playbook_to_download\"}"],"InstallDependencies":
["True_or_False"],"PlaybookFile":["file_name.yml"],"ExtraVariables":["key/
value_pairs_separated_by_a_space"],"Check":["True_or_False"],"Verbose":["-v,-
vv,-vvv, or -vvvv"]}' ^
    --association-name "name" ^
    --schedule-expression "cron_or_rate_expression"
```

Ein Beispiel.

```
aws ssm create-association --name "AWS-ApplyAnsiblePlaybooks" \
    --targets "Key=tag:OS,Values=Linux" \
    --parameters '{"SourceType":["S3"],"SourceInfo":["{\"path\":\"https://
s3.amazonaws.com/amzn-s3-demo-bucket/playbook.yml\"}"],"InstallDependencies":
["True"],"PlaybookFile":["playbook.yml"],"ExtraVariables":["SSM=True"],"Check":
["False"],"Verbose":["-v"]}' \
    --association-name "AnsibleAssociation" \
    --schedule-expression "cron(0 2 ? * SUN *)"
```

> **Note**
>
> State Manager-Zuordnungen unterstützen nicht alle Cron- und Rate-Ausdrücke. Weitere Informationen zum Erstellen von Cron- und Rate-Ausdrücken für Zuordnungen finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

Das System versucht, die Assoziation auf den Knoten zu erstellen und den Status sofort anzuwenden.

3. Führen Sie den folgenden Befehl aus, um einen aktualisierten Status der soeben erstellten Zuordnung anzuzeigen.

```
aws ssm describe-association --association-id "ID"
```

Erstellen von Zuordnungen, die Chef-Rezepte ausführen

Sie können State Manager-Zuordnungen erstellen, die Chef-Rezepte ausführen, indem Sie das `AWS-ApplyChefRecipes`-SSM-Dokument verwenden. State Manager ist ein Tool von AWS Systems Manager. Sie können mit dem `AWS-ApplyChefRecipes` SSM-Dokument eine Ausrichtung auf Linux-basierte verwaltete Systems Manager-Knoten verwenden. Dieses Dokument bietet die folgenden Vorteile für die Ausführung von Chef-Rezepten:

- Unterstützt mehrere Versionen von Chef (Chef 11 bis Chef 18).
- Installiert die Chef-Clientsoftware automatisch auf Ziel-Knoten.
- Führt optional Systems Manager-Compliance-Prüfungen für Ziel-Knoten aus und speichert die Ergebnisse der Compliance-Prüfungen in einem Amazon Simple Storage Service (Amazon S3)-Bucket.
- Führt mehrere Cookbooks und Rezepte in einem einzigen Durchlauf des Dokuments aus.
- Führt optional Rezepte im `why-run`-Modus aus, um anzuzeigen, welche Rezepte sich auf Ziel-Knoten ändern, ohne Änderungen vorzunehmen.
- Wendet optional benutzerdefinierte JSON-Attribute auf `chef-client`-Durchläufe an.
- Wendet optional benutzerdefinierte JSON-Attribute aus einer Quelldatei an, die an einem von Ihnen angegebenen Ort gespeichert ist.

Sie können Git, GitHub, HTTP oder Amazon-S3-Buckets als Download-Quellen für Chef Cookbooks und Rezepte verwenden, die Sie in einem `AWS-ApplyChefRecipes`-Dokument angeben.

> **Note**
>
> Zuordnungen, die Chef-Rezepte ausführen, werden in macOS nicht unterstützt.

Erste Schritte

Bevor Sie ein `AWS-ApplyChefRecipes`-Dokument erstellen, bereiten Sie Ihre Chef-Cookbooks und das Cookbook-Repository vor. Wenn Sie noch kein Chef Kochbuch haben, das Sie verwenden möchten, können Sie zunächst ein `HelloWorld` Testkochbuch verwenden, das für Sie vorbereitet AWS wurde. Das `AWS-ApplyChefRecipes`-Dokument verweist bereits standardmäßig auf dieses Cookbook. Ihre Cookbooks sollten ähnlich wie die folgende Verzeichnisstruktur eingerichtet werden. Im folgenden Beispiel sind `jenkins` und `nginx` Beispiele für Chef-Cookbooks, die im Chef Supermarket auf der Chef-Website verfügbar sind.

Kochbücher auf der Chef SupermarketWebsite AWS können zwar nicht offiziell unterstützt werden, aber viele von ihnen funktionieren mit dem Dokument. `AWS-ApplyChefRecipes` Im Folgenden finden Sie Beispiele für Kriterien, die Sie bestimmen müssen, wenn Sie ein Community-Cookbook testen:

- Das Cookbook sollte die Linux-basierten Betriebssysteme der Systems Manager-verwalteten Knoten unterstützen, auf die Sie zielen.
- Das Cookbook sollte für die Chef-Client-Version (Chef 11 bis Chef 18) gültig sein, die Sie verwenden.
- Das Cookbook ist kompatibel mit Chef Infra Client und erfordert keinen Chef-Server.

Stellen Sie sicher, dass Sie die `Chef.io`-Website erreichen können, damit alle Cookbooks, die Sie in der Ausführungsliste angeben, installiert werden können, wenn das Systems-Manager-Dokument (SSM-Dokument) ausgeführt wird. Die Verwendung eines eingebetteten `cookbooks`-Ordners wird zwar unterstützt, ist aber nicht erforderlich. Sie können Cookbooks direkt unter der Root-Ebene speichern.

```
<Top-level directory, or the top level of the archive file (ZIP or tgz or tar.gz)>
    ### cookbooks (optional level)
```

```
        ### jenkins
        #   ### metadata.rb
        #   ### recipes
        ### nginx
            ### metadata.rb
            ### recipes
```

> **Important**
>
> Bevor Sie eine State Manager-Zuordnung erstellen, in der Chef-Rezepte ausgeführt werden, beachten Sie, dass die Dokumentausführung die Chef-Clientsoftware auf den Systems-Manager-verwalteten Knoten installiert, es sei denn, Sie setzen den Wert der Chef-Client-Version auf `None`. Diese Operation verwendet ein Installationsskript von Chef, um Chef-Komponenten in Ihrem Namen zu installieren. Bevor Sie ein `AWS-ApplyChefRecipes`-Dokument ausführen, stellen Sie sicher, dass Ihr Unternehmen alle geltenden gesetzlichen Anforderungen einhalten kann, einschließlich der Lizenzbedingungen, die für die Verwendung von Chef-Software gelten. Weitere Informationen finden Sie auf der Chef-Website.

Systems Manager kann Compliance-Berichte an einen S3-Bucket oder die Systems Manager-Konsole übermitteln oder Compliance-Ergebnisse als Antwort auf Systems Manager-API-Befehle zur Verfügung stellen. Zum Ausführen von Systems Manager-Compliance-Berichten muss das Instance-Profil, das an Systems Manager-verwaltete Knoten angefügt ist, über Berechtigungen zum Schreiben in den S3-Bucket verfügen. Das Instance-Profil muss über die Berechtigung zur Nutzung der Systems Manager `PutComplianceItem`-API verfügen. Weitere Informationen zur Systems Manager-Compliance finden Sie unter AWS Systems Manager-Compliance.

Protokollieren der Dokumentausführung

Wenn Sie ein Systems Manager Manager-Dokument (SSM-Dokument) mithilfe einer State Manager Zuordnung ausführen, können Sie die Zuordnung so konfigurieren, dass die Ausgabe des Dokumentenlaufs ausgewählt wird, und Sie können die Ausgabe an Amazon S3 oder Amazon CloudWatch Logs (CloudWatch Logs) senden. Um die Problembehebung zu vereinfachen, wenn die Ausführung einer Zuordnung abgeschlossen ist, stellen Sie sicher, dass die Zuordnung so konfiguriert ist, dass die Befehlsausgabe entweder in einen Amazon S3 S3-Bucket oder in CloudWatch Logs geschrieben wird. Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager.

Anwenden von JSON-Attributen auf Ziele bei der Ausführung eines Rezepts

Sie können JSON-Attribute angeben, die Ihr Chef-Client während einer Zuordnung auf die Zielknoten anwenden soll. Beim Einrichten der Zuordnung können Sie unformatiertes JSON oder den Pfad zu einer in Amazon S3 gespeicherten JSON-Datei angeben.

Verwenden Sie JSON-Attribute, wenn Sie beispielsweise die Art und Weise, wie das Rezept ausgeführt wird, anpassen möchten, ohne das Rezept selbst ändern zu müssen:

- Überschreiben einer kleinen Anzahl von Attributen

  Verwenden Sie benutzerdefiniertes JSON, um zu vermeiden, dass Sie mehrere Versionen eines Rezepts verwalten müssen, um kleine Unterschiede zu berücksichtigen.
- Bereitstellung variabler Werte

  Verwenden Sie benutzerdefiniertes JSON, um Werte anzugeben, die sich von ändern können run-to-run. Wenn Ihre Chef-Cookbooks beispielsweise eine Drittanbieter-Anwendung konfigurieren, die Zahlungen akzeptiert, können Sie möglicherweise benutzerdefiniertes JSON verwenden, um die URL des Zahlungsendpunkts anzugeben.

Angeben von Attributen in unformatiertem JSON

Im Folgenden finden Sie ein Beispiel für das Format, das Sie verwenden können, um benutzerdefinierte JSON-Attribute für Ihr Chef-Rezept anzugeben.

```
{"filepath":"/tmp/example.txt", "content":"Hello, World!"}
```

Angabe eines Pfads zu einer JSON-Datei

Im Folgenden finden Sie ein Beispiel für das Format, das Sie verwenden können, um den Pfad zu benutzerdefinierten JSON-Attributen für Ihr Chef-Rezept anzugeben.

```
{"sourceType":"s3", "sourceInfo":"someS3URL1"}, {"sourceType":"s3",
 "sourceInfo":"someS3URL2"}
```

Git als Quelle für Cookbooks verwenden

Das `AWS-ApplyChefRecipes`-Dokument verwendet das aws:downloadContent-Plugin, um Chef-Cookbooks herunterzuladen. Um Inhalte aus Git herunterzuladen, geben Sie Informationen über

Ihr Git-Repository im JSON-Format an, wie im folgenden Beispiel. Ersetzen Sie jeden *example-resource-placeholder* durch Ihre Informationen.

```
{
   "repository":"GitCookbookRepository",
   "privateSSHKey":"{{ssm-secure:ssh-key-secure-string-parameter}}",
   "skipHostKeyChecking":"false",
   "getOptions":"branch:refs/head/main",
   "username":"{{ssm-secure:username-secure-string-parameter}}",
   "password":"{{ssm-secure:password-secure-string-parameter}}"
}
```

Verwenden von GitHub als Quelle für Cookbooks

Das `AWS-ApplyChefRecipes`-Dokument verwendet das aws:downloadContent-Plugin, um Cookbooks herunterzuladen. Um Inhalte aus GitHub herunterzuladen, geben Sie Informationen über Ihr GitHub-Repository im JSON-Format an, wie im folgenden Beispiel. Ersetzen Sie jeden *example-resource-placeholder* durch Ihre Informationen.

```
{
   "owner":"TestUser",
   "repository":"GitHubCookbookRepository",
   "path":"cookbooks/HelloWorld",
   "getOptions":"branch:refs/head/main",
   "tokenInfo":"{{ssm-secure:token-secure-string-parameter}}"
}
```

HTTP als Quelle für Cookbooks verwenden

Sie können Chef Cookbooks an einem benutzerdefinierten HTTP-Speicherort entweder als einzelne `.zip` oder `tar.gz`-Datei oder als Verzeichnisstruktur speichern. Um Inhalte über HTTP herunterzuladen, geben Sie den Pfad zu der Datei oder dem Verzeichnis im JSON-Format wie im folgenden Beispiel an. Ersetzen Sie jeden *example-resource-placeholder* durch Ihre Informationen.

```
{
   "url":"https://my.website.com/chef-cookbooks/HelloWorld.zip",
   "allowInsecureDownload":"false",
   "authMethod":"Basic",
   "username":"{{ssm-secure:username-secure-string-parameter}}",
   "password":"{{ssm-secure:password-secure-string-parameter}}"
```

```
}
```

Verwenden von Amazon S3 als Quelle für Cookbooks

Sie können Chef-Cookbooks auch in Amazon S3 als einzelne `.zip`- oder `tar.gz`-Datei oder als Verzeichnisstruktur speichern und herunterladen. Um Inhalte von Amazon S3 herunterzuladen, geben Sie den Pfad zu der Datei im JSON-Format wie in den folgenden Beispielen an. Ersetzen Sie jeden *`example-resource-placeholder`* durch Ihre Informationen.

Beispiel 1: Herunterladen eines bestimmten Cookbooks

```
{
   "path":"https://s3.amazonaws.com/chef-cookbooks/HelloWorld.zip"
}
```

Beispiel 2: Herunterladen des Inhalts eines Verzeichnisses

```
{
   "path":"https://s3.amazonaws.com/chef-cookbooks-test/HelloWorld"
}
```

> **Important**
>
> Wenn Sie Amazon S3 angeben, muss das Instance-Profil AWS Identity and Access Management (IAM) auf Ihren verwalteten Knoten mit der `AmazonS3ReadOnlyAccess` Richtlinie konfiguriert werden. Weitere Informationen finden Sie unter Erforderliche Instance-Berechtigungen für Systems Manager konfigurieren.

Erstellen einer Zuordnung, die Chef-Rezepte ausführt (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie mit der Systems-Manager-Konsole eine State Manager-Zuordnung erstellen, die Chef-Cookbooks mithilfe des `AWS-ApplyChefRecipes`-Dokuments ausführt.

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich State Manager aus.

3. Wählen Sie State Manager und dann Zuordnung wählen aus.
4. Geben Sie unter Name einen Namen ein, der Ihnen hilft, sich an den Zweck der Zuordnung zu erinnern.
5. Wählen Sie in der Liste Dokument die Option **`AWS-ApplyChefRecipes`** aus.
6. Wählen Sie unter Parameter für Quelltyp entweder Git, GitHub, HTTP, oder S3 aus.
7. Geben Sie unter Quelleninfo die Informationen zur Cookbook-Quelle in dem Format ein, das dem in Schritt 6 ausgewählten Quellentyp entspricht. Weitere Informationen finden Sie unter den folgenden Themen:

   - the section called “Git als Quelle für Cookbooks verwenden”
   - the section called “Verwenden von GitHub als Quelle für Cookbooks”
   - the section called “HTTP als Quelle für Cookbooks verwenden”
   - the section called “Verwenden von Amazon S3 als Quelle für Cookbooks”
8. Listen Sie in der Run list (Ausführungsliste) die auszuführenden Rezepte im folgenden Format auf. Trennen Sie jedes Rezept durch ein Komma wie gezeigt. Geben Sie kein Leerzeichen nach dem Komma ein. Ersetzen Sie jeden *`example-resource-placeholder`* durch Ihre Informationen.

   ```
   recipe[cookbook-name1::recipe-name],recipe[cookbook-name2::recipe-name]
   ```
9. (Optional) Geben Sie benutzerdefinierte JSON-Attribute an, die der Chef-Client an Ihre Zielknoten weitergeben soll.

   a. Fügen Sie unter Inhalt von JSON-Attributen alle Attribute hinzu, die der Chef-Client an Ihre Zielknoten weitergeben soll.

   b. Fügen Sie in JSON-Attributquellen die Pfade zu allen Attributen hinzu, die der Chef-Client an Ihre Zielknoten weitergeben soll.

   Weitere Informationen finden Sie unter the section called “Anwenden von JSON-Attributen auf Ziele bei der Ausführung eines Rezepts”.
10. Geben Sie für die Chef-Client-Version eine Chef-Version an. Gültige Werte sind 11 bis 18 oder `None`. Wenn Sie eine Zahl zwischen 11 und 18 (einschließlich) angeben, installiert Systems Manager die richtige Chef-Client-Version auf Ihren Zielknoten. Wenn Sie `None` angeben, installiert Systems Manager den Chef-Client nicht auf Ziel-Knoten, bevor Sie die Rezepte des Dokuments ausführen.

11. (Optional) Geben Sie für Chef-Client-Argumente zusätzliche Argumente an, die für die von Ihnen verwendete Version von Chef unterstützt werden. Um mehr über unterstützte Argumente zu erfahren, führen Sie `chef-client -h` auf einem Knoten aus, auf dem der Chef-Client ausgeführt wird.
12. (Optional) Aktivieren Sie Why-run, um Änderungen anzuzeigen, die bei der Ausführung der Rezepte an Ziel-Knoten vorgenommen wurden, ohne dass die Ziel-Knoten tatsächlich geändert werden.
13. Wählen Sie für Compliance severity (Schweregrad der Compliance) den Schweregrad der Systems Manager-Compliance-Ergebnisse aus, die gemeldet werden sollen. In den Compliance-Berichten finden Sie Informationen dazu, ob die Zuordnung konform ist, zusammen mit dem festgelegten Schweregrad. Compliance-Berichte werden in einem S3-Bucket gespeichert, den Sie als Wert des Parameters Compliance report bucket (Compliance-Berichts-Bucket) angeben (Schritt 14). Weitere Informationen zur Compliance finden Sie unter Erfahren Sie mehr über Compliance in dieser Anleitung.

    Compliance-Scans messen die Abweichung zwischen der Konfiguration, die in Ihren Chef-Rezepten und Knoten-Ressourcen angegeben ist. Gültige Werte sind `Critical`, `High`, `Medium`, `Low`, `Informational`, `Unspecified` oder `None`. Um die Compliance-Berichterstattung zu überspringen, wählen Sie `None`.
14. Geben Sie unter Compliance type (Compliance-Typ) den Compliance-Typ an, für den die Ergebnisse gemeldet werden sollen. Gültige Werte sind `Association` für State Manager Assoziationen oder `Custom:`*`custom-type`*. Der Standardwert ist `Custom:Chef`.
15. Geben Sie unter Compliance-Berichts-Bucket den Namen eines S3-Buckets ein, in dem Informationen zu jeder Chef-Ausführung dieses Dokuments gespeichert werden sollen, einschließlich der Ressourcenkonfiguration und der Ergebnisse der Compliance.
16. Konfigurieren Sie in Rate control (Ratensteuerung) Optionen für die Ausführung von State Manager-Zuordnungen in der Flotte von verwalteten Knoten. Weitere Informationen über Ratensteuerungen finden Sie unter Verstehen von Zielen und Ratensteuerungen in State Manager Zuordnungen.

    Wählen Sie unter Concurrency (Gleichzeitigkeit) eine Option aus:

    - Wählen Sie Ziele aus, um eine absolute Anzahl von Zielen einzugeben, die die Zuordnung gleichzeitig ausführen können.
    - Wählen Sie Prozentsatz aus, um einen Prozentsatz der Ziele anzugeben, die die Zuordnung gleichzeitig ausführen können.

Wählen Sie unter Error threshold (Fehlerschwelle) eine Option aus:

- Wählen Sie Fehler aus und geben Sie die absolute Anzahl erlaubter Fehler an, bis State Manager die Ausführung von Zuordnungen für weitere Ziele beendet.
- Wählen Sie Prozentsatz aus und geben Sie den Prozentsatz erlaubter Fehler an, bis State Manager die Ausführung von Zuordnungen für weitere Ziele beendet.

17. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

18. Wählen Sie Zuordnung erstellen.

Erstellen einer Zuordnung, die Chef-Rezepte ausführt (CLI)

Das folgende Verfahren beschreibt, wie Sie mit AWS Command Line Interface (AWS CLI) eine State Manager Assoziation erstellen, die Chef-Kochbücher mithilfe des `AWS-ApplyChefRecipes` Dokuments ausführt.

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie einen der folgenden Befehle aus, um eine Zuordnung zu erstellen, die Chef Cookbooks auf Zielknoten ausführt, die die angegebenen Tags haben. Verwenden Sie den

Befehl, der für Ihren Quellentyp des Cookbooks und Ihr Betriebssystem geeignet ist. Ersetzen Sie jeden *example-resource-placeholder* durch Ihre Informationen.

a. Git-Quelle

Linux & macOS

```
aws ssm create-association --name "AWS-ApplyChefRecipes" \
    --targets Key=tag:TagKey,Values=TagValue \
    --parameters '{"SourceType":["Git"],"SourceInfo":["{\"repository\":
\"repository-name\", \"getOptions\": \"branch:branch-name\", \"username
\": \"{{ ssm-secure:username-secure-string-parameter }}\", \"password\":
 \"{{ ssm-secure:password-secure-string-parameter }}\"}"], "RunList":
["{\"recipe[cookbook-name-1::recipe-name]\", \"recipe[cookbook-
name-2::recipe-name]\"}"], "JsonAttributesContent": ["{custom-json-
content}"], "JsonAttributesSources": "{\"sourceType\":\"s3\", \"sourceInfo
\":\"s3-bucket-endpoint-1\"}, {\"sourceType\":\"s3\", \"sourceInfo\":
\"s3-bucket-endpoint-2\"}", "ChefClientVersion": ["version-number"],
 "ChefClientArguments":["{chef-client-arguments}"], "WhyRun": boolean,
 "ComplianceSeverity": ["severity-value"], "ComplianceType":
 ["Custom:Chef"], "ComplianceReportBucket": ["s3-bucket-name"]}' \
    --association-name "name" \
    --schedule-expression "cron-or-rate-expression"
```

Windows

```
aws ssm create-association --name "AWS-ApplyChefRecipes" ^
    --targets Key=tag:TagKey,Values=TagValue ^
    --parameters '{"SourceType":["Git"],"SourceInfo":["{\"repository\":
\"repository-name\", \"getOptions\": \"branch:branch-name\", \"username
\": \"{{ ssm-secure:username-secure-string-parameter }}\", \"password\":
 \"{{ ssm-secure:password-secure-string-parameter }}\"}"], "RunList":
["{\"recipe[cookbook-name-1::recipe-name]\", \"recipe[cookbook-
name-2::recipe-name]\"}"], "JsonAttributesContent": ["{custom-json}"],
 "JsonAttributesSources": "{\"sourceType\":\"s3\", \"sourceInfo\":
\"s3-bucket-endpoint-1\"}, {\"sourceType\":\"s3\", \"sourceInfo\":
\"s3-bucket-endpoint-2\"}", "ChefClientVersion": ["version-number"],
 "ChefClientArguments":["{chef-client-arguments}"], "WhyRun": boolean,
 "ComplianceSeverity": ["severity-value"], "ComplianceType":
 ["Custom:Chef"], "ComplianceReportBucket": ["s3-bucket-name"]}' ^
    --association-name "name" ^
    --schedule-expression "cron-or-rate-expression"
```

b. GitHub source

Linux & macOS

```
aws ssm create-association --name "AWS-ApplyChefRecipes" \
    --targets Key=tag:TagKey,Values=TagValue \
    --parameters '{"SourceType":["GitHub"],"SourceInfo":["{\"owner\":
\"owner-name\", \"repository\": \"name\", \"path\": \"path-to-directory-
or-cookbook-to-download\", \"getOptions\": \"branch:branch-name\"}"],
 "RunList":["{\"recipe[cookbook-name-1::recipe-name]\", \"recipe[cookbook-
name-2::recipe-name]\"}"], "JsonAttributesContent": ["{custom-json}"],
 "ChefClientVersion": ["version-number"], "ChefClientArguments":["{chef-
client-arguments}"], "WhyRun": boolean, "ComplianceSeverity": ["severity-
value"], "ComplianceType": ["Custom:Chef"], "ComplianceReportBucket": ["s3-
bucket-name"]}' \
    --association-name "name" \
    --schedule-expression "cron-or-rate-expression"
```

Windows

```
aws ssm create-association --name "AWS-ApplyChefRecipes" ^
    --targets Key=tag:TagKey,Values=TagValue \
    --parameters '{"SourceType":["GitHub"],"SourceInfo":["{\"owner\":
\"owner-name\", \"repository\": \"name\", \"path\": \"path-to-directory-
or-cookbook-to-download\", \"getOptions\": \"branch:branch-name\"}"],
 "RunList":["{\"recipe[cookbook-name-1::recipe-name]\", \"recipe[cookbook-
name-2::recipe-name]\"}"], "JsonAttributesContent": ["{custom-json}"],
 "ChefClientVersion": ["version-number"], "ChefClientArguments":["{chef-
client-arguments}"], "WhyRun": boolean, "ComplianceSeverity": ["severity-
value"], "ComplianceType": ["Custom:Chef"], "ComplianceReportBucket": ["s3-
bucket-name"]}' ^
    --association-name "name" ^
    --schedule-expression "cron-or-rate-expression"
```

Ein Beispiel.

Linux & macOS

```
aws ssm create-association --name "AWS-ApplyChefRecipes" \
    --targets Key=tag:OS,Values=Linux \
```

```
    --parameters '{"SourceType":["GitHub"],"SourceInfo":["{\"owner
\":\"ChefRecipeTest\", \"repository\": \"ChefCookbooks\", \"path
\": \"cookbooks/HelloWorld\", \"getOptions\": \"branch:master
\"}"], "RunList":["{\"recipe[HelloWorld::HelloWorldRecipe]\",
 \"recipe[HelloWorld::InstallApp]\"}"], "JsonAttributesContent":
 ["{\"state\": \"visible\",\"colors\": {\"foreground\": \"light-blue
\",\"background\": \"dark-gray\"}}"], "ChefClientVersion": ["14"],
 "ChefClientArguments":["{--fips}"], "WhyRun": false, "ComplianceSeverity":
 ["Medium"], "ComplianceType": ["Custom:Chef"], "ComplianceReportBucket":
 ["ChefComplianceResultsBucket"]}' \
    --association-name "MyChefAssociation" \
    --schedule-expression "cron(0 2 ? * SUN *)"
```

Windows

```
aws ssm create-association --name "AWS-ApplyChefRecipes" ^
    --targets Key=tag:OS,Values=Linux ^
    --parameters '{"SourceType":["GitHub"],"SourceInfo":["{\"owner
\":\"ChefRecipeTest\", \"repository\": \"ChefCookbooks\", \"path
\": \"cookbooks/HelloWorld\", \"getOptions\": \"branch:master
\"}"], "RunList":["{\"recipe[HelloWorld::HelloWorldRecipe]\",
 \"recipe[HelloWorld::InstallApp]\"}"], "JsonAttributesContent":
 ["{\"state\": \"visible\",\"colors\": {\"foreground\": \"light-blue
\",\"background\": \"dark-gray\"}}"], "ChefClientVersion": ["14"],
 "ChefClientArguments":["{--fips}"], "WhyRun": false, "ComplianceSeverity":
 ["Medium"], "ComplianceType": ["Custom:Chef"], "ComplianceReportBucket":
 ["ChefComplianceResultsBucket"]}' ^
    --association-name "MyChefAssociation" ^
    --schedule-expression "cron(0 2 ? * SUN *)"
```

c. HTTP-Quelle

Linux & macOS

```
aws ssm create-association --name "AWS-ApplyChefRecipes" \
    --targets Key=tag:TagKey,Values=TagValue \
    --parameters '{"SourceType":["HTTP"],"SourceInfo":["{\"url\":\"url-
to-zip-file|directory|cookbook\", \"authMethod\": \"auth-method\",
 \"username\": \"{{ ssm-secure:username-secure-string-parameter }}\",
 \"password\": \"{{ ssm-secure:password-secure-string-parameter }}\"}"],
 "RunList":["{\"recipe[cookbook-name-1::recipe-name]\", \"recipe[cookbook-
name-2::recipe-name]\"}"], "JsonAttributesContent": ["{custom-json-
content}"], "JsonAttributesSources": "{\"sourceType\":\"s3\", \"sourceInfo
```

```
\":\"s3-bucket-endpoint-1\"}, {\"sourceType\":\"s3\", \"sourceInfo\":
\"s3-bucket-endpoint-2\"}", "ChefClientVersion": ["version-number"],
 "ChefClientArguments":["{chef-client-arguments}"], "WhyRun": boolean,
 "ComplianceSeverity": ["severity-value"], "ComplianceType":
 ["Custom:Chef"], "ComplianceReportBucket": ["s3-bucket-name"]}' \
    --association-name "name" \
    --schedule-expression "cron-or-rate-expression"
```

Windows

```
aws ssm create-association --name "AWS-ApplyChefRecipes" ^
    --targets Key=tag:TagKey,Values=TagValue ^
    --parameters '{"SourceType":["HTTP"],"SourceInfo":["{\"url\":\"url-
to-zip-file|directory|cookbook\", \"authMethod\": \"auth-method\",
 \"username\": \"{{ ssm-secure:username-secure-string-parameter }}\",
 \"password\": \"{{ ssm-secure:password-secure-string-parameter }}\"}"],
 "RunList":["{\"recipe[cookbook-name-1::recipe-name]\", \"recipe[cookbook-
name-2::recipe-name]\"}"], "JsonAttributesContent": ["{custom-json-
content}"], "JsonAttributesSources": "{\"sourceType\":\"s3\", \"sourceInfo
\":\"s3-bucket-endpoint-1\"}, {\"sourceType\":\"s3\", \"sourceInfo\":
\"s3-bucket-endpoint-2\"}", "ChefClientVersion": ["version-number"],
 "ChefClientArguments":["{chef-client-arguments}"], "WhyRun": boolean,
 "ComplianceSeverity": ["severity-value"], "ComplianceType":
 ["Custom:Chef"], "ComplianceReportBucket": ["s3-bucket-name"]}' \
    --association-name "name" ^
    --schedule-expression "cron-or-rate-expression"
```

d. Amazon-S3-Quelle

Linux & macOS

```
aws ssm create-association --name "AWS-ApplyChefRecipes" \
    --targets Key=tag:TagKey,Values=TagValue \
    --parameters '{"SourceType":["S3"],"SourceInfo":["{\"path\":\"https://
s3.amazonaws.com/path_to_Zip_file,_directory,_or_cookbook_to_download\"}"],
 "RunList":["{\"recipe[cookbook_name1::recipe_name]\",
 \"recipe[cookbook_name2::recipe_name]\"}"], "JsonAttributesContent":
 ["{Custom_JSON}"], "ChefClientVersion": ["version_number"],
 "ChefClientArguments":["{chef_client_arguments}"], "WhyRun": true_or_false,
 "ComplianceSeverity": ["severity_value"], "ComplianceType":
 ["Custom:Chef"], "ComplianceReportBucket": ["amzn-s3-demo-bucket"]}' \
    --association-name "name" \
```

```
    --schedule-expression "cron_or_rate_expression"
```

Windows

```
aws ssm create-association --name "AWS-ApplyChefRecipes" ^
    --targets Key=tag:TagKey,Values=TagValue ^
    --parameters '{"SourceType":["S3"],"SourceInfo":["{\"path\":\"https://
s3.amazonaws.com/path_to_Zip_file,_directory,_or_cookbook_to_download\"}"],
 "RunList":["{\"recipe[cookbook_name1::recipe_name]\",
 \"recipe[cookbook_name2::recipe_name]\"}"], "JsonAttributesContent":
 ["{Custom_JSON}"], "ChefClientVersion": ["version_number"],
 "ChefClientArguments":["{chef_client_arguments}"], "WhyRun": true_or_false,
 "ComplianceSeverity": ["severity_value"], "ComplianceType":
 ["Custom:Chef"], "ComplianceReportBucket": ["amzn-s3-demo-bucket"]}' ^
    --association-name "name" ^
    --schedule-expression "cron_or_rate_expression"
```

Ein Beispiel.

Linux & macOS

```
aws ssm create-association --name "AWS-ApplyChefRecipes" \
    --targets "Key=tag:OS,Values= Linux" \
    --parameters '{"SourceType":["S3"],"SourceInfo":["{\"path
\":\"https://s3.amazonaws.com/amzn-s3-demo-bucket/HelloWorld
\"}"], "RunList":["{\"recipe[HelloWorld::HelloWorldRecipe]\",
 \"recipe[HelloWorld::InstallApp]\"}"], "JsonAttributesContent":
 ["{\"state\": \"visible\",\"colors\": {\"foreground\": \"light-blue
\",\"background\": \"dark-gray\"}}"], "ChefClientVersion": ["14"],
 "ChefClientArguments":["{--fips}"], "WhyRun": false, "ComplianceSeverity":
 ["Medium"], "ComplianceType": ["Custom:Chef"], "ComplianceReportBucket":
 ["ChefComplianceResultsBucket"]}' \
    --association-name "name" \
    --schedule-expression "cron(0 2 ? * SUN *)"
```

Windows

```
aws ssm create-association --name "AWS-ApplyChefRecipes" ^
    --targets "Key=tag:OS,Values= Linux" ^
    --parameters '{"SourceType":["S3"],"SourceInfo":["{\"path
\":\"https://s3.amazonaws.com/amzn-s3-demo-bucket/HelloWorld
```

```
\"}"], "RunList":["{\"recipe[HelloWorld::HelloWorldRecipe]\",
 \"recipe[HelloWorld::InstallApp]\"}"], "JsonAttributesContent":
 ["{\"state\": \"visible\",\"colors\": {\"foreground\": \"light-blue
\",\"background\": \"dark-gray\"}}"], "ChefClientVersion": ["14"],
 "ChefClientArguments":["{--fips}"], "WhyRun": false, "ComplianceSeverity":
 ["Medium"], "ComplianceType": ["Custom:Chef"], "ComplianceReportBucket":
 ["ChefComplianceResultsBucket"]}' ^
    --association-name "name" ^
    --schedule-expression "cron(0 2 ? * SUN *)"
```

Das System erstellt die Zuordnung und führt die Zuordnung auf den Zielknoten aus, es sei denn, Ihr angegebener cron- oder rate-Ausdruck verhindert dies.

> **Note**
>
> State Manager-Zuordnungen unterstützen nicht alle Cron- und Rate-Ausdrücke. Weitere Informationen zum Erstellen von Cron- und Rate-Ausdrücken für Zuordnungen finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

3. Führen Sie den folgenden Befehl aus, um den Status der Zuordnung, die Sie gerade erstellt haben, anzuzeigen.

```
aws ssm describe-association --association-id "ID"
```

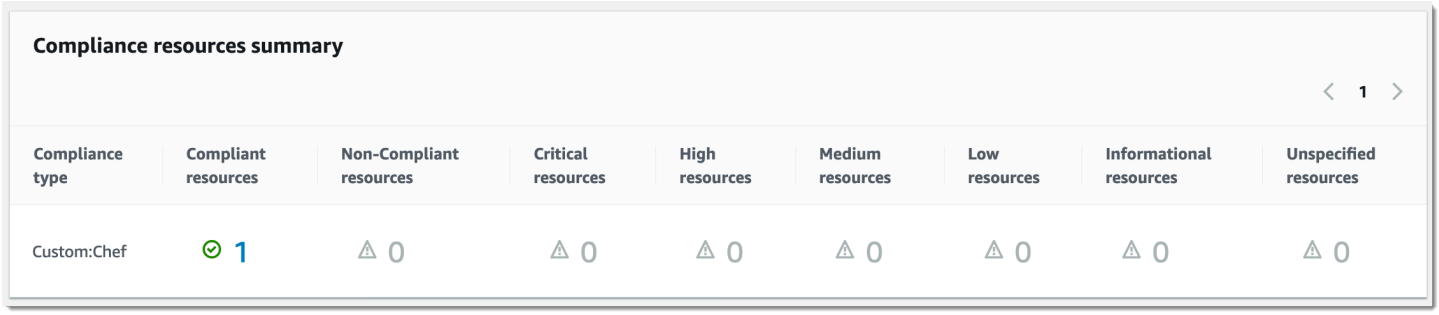
Anzeigen von Details zur Chef-Ressourcen-Compliance

Systems Manager erfasst Compliance-Informationen über von Chef verwaltete Ressourcen im Compliance-Berichts-Bucket-Wert von Amazon S3, den Sie bei der Ausführung des `AWS-ApplyChefRecipes`-Dokuments angegeben haben. Die Suche nach Informationen zu Chef-Ressourcenfehlern in einem S3-Bucket kann zeitaufwendig sein. Stattdessen können Sie diese Informationen auf der Systems Manager-Seite Compliance anzeigen.

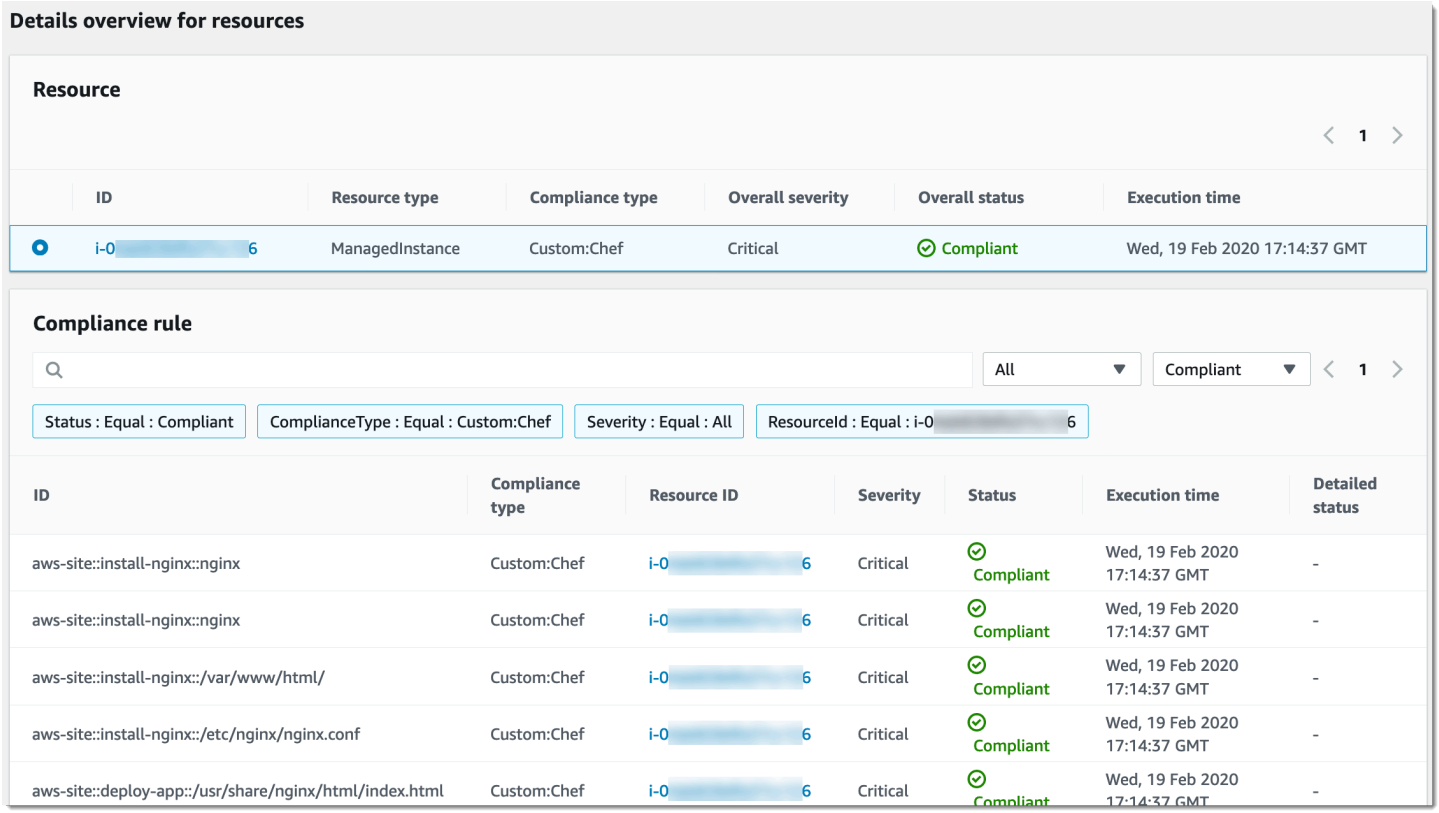
Ein Systems-Manager-Compliance-Scan sammelt Informationen zu Ressourcen auf Ihren verwalteten Knoten, die in der letzten Chef-Ausführung erstellt oder überprüft wurden. Die Ressourcen können unter anderem Dateien, Verzeichnisse, `systemd`-Services, `yum`-Pakete, Vorlagendateien, `gem`-Pakete und abhängige Cookbooks umfassen.

Der Bereich Compliance-Ressourcen-Zusammenfassung zeigt die Anzahl der Ressourcen an, die fehlgeschlagen sind. Im folgenden Beispiel ComplianceTypelautet Custom: Chef und eine Ressource ist nicht konform.

> **Note**
>
> `Custom:Chef`ist der ComplianceTypeStandardwert im `AWS-ApplyChefRecipes` Dokument. Dieser Wert ist anpassbar.



Der Abschnitt „Detailübersicht für Ressourcen“ enthält Informationen über die AWS Ressource, die nicht richtlinientreu ist. Dieser Abschnitt enthält auch den Chef-Ressourcentyp, für den die Compliance ausgeführt wurde, den Schweregrad des Problems, den Compliance-Status und Links zu weiteren Informationen, falls vorhanden.



View output zeigt die letzten 4.000 Zeichen des detaillierten Status an. Systems Manager beginnt mit der Ausnahme als erstem Element, sucht nach ausführlichen Meldungen und zeigt diese an, bis das Kontingent von 4.000 Zeichen erreicht ist. Dieser Vorgang zeigt die Protokollmeldungen an, die vor dem Auslösen der Ausnahme ausgegeben wurden. Dabei handelt es sich um die relevantesten Nachrichten für die Fehlerbehebung.

Weitere Informationen zum Anzeigen von Compliance-Informationen finden Sie unter AWS Systems Manager-Compliance.

> **Important**
>
> Wenn die State Manager-Zuordnung fehlschlägt, werden keine Compliance-Daten gemeldet. Beispiel: Wenn Systems Manager versucht, ein Chef-Cookbook aus einem S3-Bucket herunterzuladen, und der Knoten keine Berechtigung zum Zugriff besitzt, schlägt die Zuordnung fehl und Systems Manager meldet keine Compliance-Daten.

Exemplarische Vorgehensweise: Automatisches Update SSM Agent mit AWS CLI

Mit den folgenden Schritten können Sie eine State Manager-Zuordnung mithilfe der AWS Command Line Interface erstellen. Durch die Zuordnung wird der SSM Agent automatisch entsprechend einem von Ihnen angegebenen Zeitplan aktualisiert. Mehr über SSM Agent erfahren Sie unter Arbeiten mit SSM Agent. Informationen zum Anpassen des Aktualisierungszeitplans für SSM Agent mithilfe der Konsole finden Sie unter Automatische Aktualisierung von SSM Agent.

Wenn Sie über SSM Agent-Aktualisierungen benachrichtigt werden möchten, abonnieren Sie die Seite SSM Agent-Versionshinweise auf GitHub.

Bevor Sie beginnen

Bevor Sie das folgende Verfahren ausführen, stellen Sie sicher, dass mindestens eine Amazon Elastic Compute Cloud (Amazon EC2) -Instance für Linux läuft oder Windows Server dass diese für Systems Manager konfiguriert ist. macOS Weitere Informationen finden Sie unter Einrichten von verwalteten Knoten für AWS Systems Manager.

Wenn Sie eine Zuordnung mithilfe von AWS CLI oder erstellen AWS Tools for Windows PowerShell, verwenden Sie den `--Targets` Parameter, um Instances als Ziel zu verwenden, wie im folgenden Beispiel gezeigt. Verwenden Sie nicht den Parameter `--InstanceID`. Der Parameter `--InstanceID` ist veraltet.

So erstellen Sie eine Zuordnung für die automatische Aktualisierung von SSM Agent

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus, um eine Zuordnung zu erstellen, indem Sie auf Instances abzielen, die Amazon Elastic Compute Cloud (Amazon EC2) -Tags verwenden. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen. Der `Schedule`-Parameter legt einen Zeitplan für die Ausführung der Zuordnung an jedem Sonntagmorgen um 2:00 Uhr (UTC) fest.

   State Manager-Zuordnungen unterstützen nicht alle Cron- und Rate-Ausdrücke. Weitere Informationen zum Erstellen von Cron- und Rate-Ausdrücken für Zuordnungen finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

Linux & macOS

```
aws ssm create-association \
--targets Key=tag:tag_key,Values=tag_value \
--name AWS-UpdateSSMAgent \
--schedule-expression "cron(0 2 ? * SUN *)"
```

Windows

```
aws ssm create-association ^
--targets Key=tag:tag_key,Values=tag_value ^
--name AWS-UpdateSSMAgent ^
--schedule-expression "cron(0 2 ? * SUN *)"
```

Sie können mehrere Instances als Ziel angeben, indem Sie Instances IDs in einer durch Kommas getrennten Liste angeben.

Linux & macOS

```
aws ssm create-association \
--targets Key=instanceids,Values=instance_ID,instance_ID,instance_ID \
--name AWS-UpdateSSMAgent \
--schedule-expression "cron(0 2 ? * SUN *)"
```

Windows

```
aws ssm create-association ^
--targets Key=instanceids,Values=instance_ID,instance_ID,instance_ID ^
--name AWS-UpdateSSMAgent ^
--schedule-expression "cron(0 2 ? * SUN *)"
```

Sie können die Version des SSM Agent angeben, den Sie aktualisieren möchten.

Linux & macOS

```
aws ssm create-association \
--targets Key=instanceids,Values=instance_ID,instance_ID,instance_ID \
--name AWS-UpdateSSMAgent \
```

```
--schedule-expression "cron(0 2 ? * SUN *)" \
--parameters version=ssm_agent_version_number
```

Windows

```
aws ssm create-association ^
--targets Key=instanceids,Values=instance_ID,instance_ID,instance_ID ^
--name AWS-UpdateSSMAgent ^
--schedule-expression "cron(0 2 ? * SUN *)" ^
--parameters version=ssm_agent_version_number
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "AssociationDescription": {
        "ScheduleExpression": "cron(0 2 ? * SUN *)",
        "Name": "AWS-UpdateSSMAgent",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "AssociationId": "123.............",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1504034257.98,
        "Date": 1504034257.98,
        "AssociationVersion": "1",
        "Targets": [
            {
                "Values": [
                    "TagValue"
                ],
                "Key": "tag:TagKey"
            }
        ]
    }
}
```

Das System versucht, die Zuordnung für die Instances zu erstellen und wendet den Status nach der Erstellung an. Der Zuordnungsstatus lautet `Pending` (Schwebend).

3. Führen Sie den folgenden Befehl aus, um einen aktualisierten Status der erstellten Zuordnung anzuzeigen.

```
aws ssm list-associations
```

   Wenn auf Ihren Instances nicht die neueste Version von SSM Agent ausgeführt wird, wird der Status `Failed` angezeigt. Bei Veröffentlichung einer neuen SSM Agent-Version wird der neue Agent automatisch von der Zuordnung installiert und der Status zeigt `Success` an.

Exemplarische Vorgehensweise: Automatisches Aktualisieren von PV-Treibern auf EC2 Instanzen für Windows Server

Amazon Windows Server Amazon Machine Images (AMIs) enthalten eine Reihe von Treibern, die den Zugriff auf virtualisierte Hardware ermöglichen. Diese Treiber werden von Amazon Elastic Compute Cloud (Amazon EC2) verwendet, um Instance-Speicher- und Amazon Elastic Block Store (Amazon EBS) -Volumes ihren Geräten zuzuordnen. Wir empfehlen Ihnen, die neuesten Treiber zu installieren, um die Stabilität und Leistung Ihrer EC2 Instances für Windows Server zu verbessern. Weitere Informationen zu PV-Treibern finden Sie unter AWS PV-Treiber.

Die folgende exemplarische Vorgehensweise zeigt Ihnen, wie Sie eine State Manager Zuordnung so konfigurieren, dass neue AWS PV-Treiber automatisch heruntergeladen und installiert werden, sobald die Treiber verfügbar sind. State Managerist ein Tool in AWS Systems Manager.

Bevor Sie beginnen

Bevor Sie das folgende Verfahren ausführen, stellen Sie sicher, dass Sie mindestens eine EC2 Amazon-Instance Windows Server ausführen, die für Systems Manager konfiguriert ist. Weitere Informationen finden Sie unter Einrichten von verwalteten Knoten für AWS Systems Manager.

So erstellen Sie eine State Manager-Zuordnung, die automatisch PV-Treiber aktualisiert

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager aus.
3. Wählen Sie Create association (Zuordnung erstellen) aus.
4. Geben Sie im Feld Name einen beschreibenden Namen für die Zuordnung ein.
5. Wählen Sie in der Liste Dokument die Option `AWS-ConfigureAWSPackage` aus.

6. Gehen Sie im Abschnitt Parameter wie folgt vor:

   - Wählen Sie für Action (Aktion) die Option Install (Installieren).
   - Wählen Sie für Installation type (Art der Installation) Uninstall and reinstall (Deinstallieren und neu installieren).

   > **Note**
   >
   > Direkte Upgrades werden für dieses Paket nicht unterstützt. Es muss deinstalliert und neu installiert werden.

   - Geben Sie unter Name **`AWSPVDriver`** ein.

   Sie müssen nichts für Version und Zusätzliche Argumente eingeben.

7. Identifizieren Sie für den Abschnitt Ziele die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

   > **Note**
   >
   > Wenn Sie Ziel-Instances mittels Tags auswählen und Tags angeben, die Linux-Instances zugeordnet sind, ist die Zuordnung zwar auf der Windows Server-Instance erfolgreich, schlägt jedoch auf den Linux-Instances fehl. Der Gesamtstatus der Zuordnung zeigt Failed (Fehler) an.

8. Wählen Sie im Bereich Zeitplan angeben aus, ob die Zuordnung nach einem von Ihnen konfigurierten Zeitplan oder nur einmal ausgeführt werden soll. Aktualisierte PV-Treiber werden mehrere Male pro Jahr veröffentlicht. Wenn Sie möchten, können Sie die Zuordnung einmal pro Monat ausführen lassen.

9. Wählen Sie unter Erweiterte Optionen für Compliance-Schweregrad einen Schweregrad für die Zuordnung aus. In den Compliance-Berichten finden Sie Informationen dazu, ob die Zuordnung konform ist, zusammen mit dem Schweregrad, den Sie hier angeben. Weitere Informationen finden Sie unter Informationen zu State Manager-Zuordnungs-Compliance.
10. Für Ratenregelung:

    - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

      > **Note**
      >
      > Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

    - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.
11. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

    > **Note**
    >
    > Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

12. (Optional) Wählen Sie im Bereich CloudWatch Alarm unter Alarmname einen CloudWatch Alarm aus, der auf Ihre Assoziation zur Überwachung angewendet werden soll.

    > **Note**
    >
    > Bitte beachten Sie die folgenden Informationen über diesen Schritt.
    >
    > - Die Liste der Alarme zeigt maximal 100 Alarme. Wenn Sie Ihren Alarm nicht in der Liste sehen, verwenden Sie den, AWS Command Line Interface um die Zuordnung zu erstellen. Weitere Informationen finden Sie unter Erstellen einer Zuordnung (Befehlszeile).
    > - Um Ihrem Befehl einen CloudWatch Alarm anzuhängen, muss der IAM-Principal, der die Zuordnung erstellt, über die entsprechende Berechtigung für die `iam:createServiceLinkedRole` Aktion verfügen. Weitere Informationen zu CloudWatch Alarmen finden Sie unter CloudWatch Amazon-Alarme verwenden.
    > - Ausstehende Befehlsaufrufe oder Automatisierungen werden nicht ausgeführt, wenn Ihr Alarm aktiviert wird.

13. Wählen Sie Create association (Zuordnung erstellen) und dann Close (Schließen) aus. Das System versucht, die Zuordnung auf den Instances zu erstellen und den Status sofort anzuwenden.

    Wenn Sie die Zuordnung auf einer oder mehreren EC2 Amazon-Instances für erstellt habenWindows Server, ändert sich der Status in Success. Wenn Ihre Instances nicht ordnungsgemäß für Systems Manager konfiguriert sind, oder wenn Sie versehentlich Linux-Instances ausgewählt haben, wird der Status Failed (Fehler) angezeigt.

    Wenn der Status Fehlgeschlagen lautet, wählen Sie die Zuordnungs-ID aus, wählen Sie den Tab Ressourcen und überprüfen Sie dann, ob die Zuordnung auf Ihren EC2 Instances für erfolgreich erstellt wurdeWindows Server. Wenn EC2 Instances für den Status Fehlgeschlagen Windows Server anzeigen, stellen Sie sicher, dass die auf der Instance ausgeführt SSM Agent wird, und stellen Sie sicher, dass die Instance mit einer AWS Identity and Access Management (IAM-) Rolle für Systems Manager konfiguriert ist. Weitere Informationen finden Sie unter Einrichten der einheitlichen Systems-Manager-Konsole für eine Organisation.

AWS Systems Manager Tools für das Änderungsmanagement

AWS Systems Manager stellt die folgenden Tools bereit, mit denen Sie Änderungen an Ihren AWS Ressourcen vornehmen können.

Themen

- AWS Systems Manager-Automatisierung
- AWS Systems Manager Change Calendar
- AWS Systems Manager Change Manager
- AWS Systems Manager-Documents
- AWS Systems Manager Maintenance Windows
- AWS Systems Manager Quick Setup

AWS Systems Manager-Automatisierung

Automation, ein Tool in AWS Systems Manager, vereinfacht allgemeine Wartungs-, Bereitstellungs- und Problembehebungsaufgaben für AWS-Services Amazon Elastic Compute Cloud (Amazon EC2), Amazon Relational Database Service (Amazon RDS), Amazon Redshift, Amazon Simple Storage Service (Amazon S3) und viele mehr. Um mit der Automatisierung zu beginnen, öffnen Sie die Systems-Manager-Konsole. Klicken Sie im Navigationsbereich auf Automation.

Automation hilft Ihnen, automatisierte Lösungen für die Bereitstellung, Konfiguration und Verwaltung von AWS -Ressourcen im großen Umfang zu entwickeln. Mit Automation haben Sie eine detaillierte Kontrolle über die Nebenläufigkeit der Automatisierungen. So können Sie zum Beispiel angeben, wie viele Ressourcen gleichzeitig verarbeitet werden sollen und wie viele Fehler auftreten können, bevor eine Automatisierung gestoppt wird.

Um Ihnen den Einstieg in die Automatisierung zu erleichtern, AWS entwickelt und verwaltet das Unternehmen mehrere vordefinierte Runbooks. Je nach Anwendungsfall können Sie diese vordefinierten Runbooks nutzen, die verschiedene Aufgaben ausführen, oder eigene benutzerdefinierte Runbooks erstellen, die Ihren Anforderungen besser entsprechen. Um den Fortschritt und den Status der Automatisierungen zu überwachen, können Sie die Systems-Manager-Automation-Konsole oder Ihr bevorzugtes Befehlszeilen-Tool nutzen. Automation lässt sich auch in Amazon integrieren EventBridge , um Sie beim Aufbau einer ereignisgesteuerten Architektur in großem Maßstab zu unterstützen.

> **Note**
>
> Für Kunden, die seit dem 14. August 2025 neu bei Systems Manager Automation sind, ist das kostenlose Kontingent Automation nicht verfügbar. Für Kunden, die Automation bereits nutzen, endet das kostenlose Nutzungskontingent am 31. Dezember 2025. Informationen zu den Servicekosten finden Sie unter AWS Systems Manager Preise.

Wie kann meine Organisation von Automation profitieren?

Automation bietet die folgenden Vorteile:

- Unterstützung der Skripterstellung in Runbook-Inhalten

  Mit der `aws:executeScript` Aktion können Sie benutzerdefiniertes Python und PowerShell Funktionen direkt von Ihren Runbooks aus ausführen. Das bietet Ihnen eine größere Flexibilität beim Erstellen eigener Runbooks, da Sie verschiedene Aufgaben ausführen können, die andere Automation-Aktionen nicht unterstützen. Zudem haben Sie eine bessere Kontrolle über die Logik des Runbooks. Ein Beispiel dafür, wie diese Aktion genutzt werden kann und wie sie zur Verbesserung einer bestehenden automatisierten Lösung beitragen kann, finden Sie unter Erstellen von Automation-Runbooks.
- Führen Sie Automatisierungen für mehrere Benutzer AWS-Konten und AWS-Regionen von einem zentralen Ort aus aus

  Administratoren können über die Systems-Manager-Konsole Automatisierungen für Ressourcen in mehreren Konten und Regionen ausführen.
- Verbesserte Betriebssicherheit

  Administratoren verfügen über eine zentrale Stelle zum Erteilen und Widerrufen des Zugriffs auf Runbooks. Mithilfe von reinen AWS Identity and Access Management (IAM-) Richtlinien können Sie steuern, welche einzelnen Benutzer oder Gruppen in Ihrer Organisation Automation verwenden können und auf welche Runbooks sie zugreifen können.
- Automatisieren von häufigen IT-Aufgaben

  Die Automatisierung häufiger Aufgaben kann dazu beitragen, die betriebliche Effizienz zu verbessern, organisatorische Standards durchzusetzen und Bedienfehler zu reduzieren. Sie können das `AWS-UpdateCloudFormationStackWithApproval` Runbook beispielsweise verwenden, um Ressourcen zu aktualisieren, die mithilfe einer Vorlage bereitgestellt wurden.

AWS CloudFormation Die Aktualisierung wendet eine neue Vorlage an. Sie können Automation so konfigurieren, dass es eine Genehmigung von einem oder mehreren -Benutzer anfordert, bevor die Aktualisierung beginnt.

- Sichere Ausführung störender Aufgaben auf einmal

  Automation umfasst Funktionen wie etwa Ratensteuerelemente, mit deren Hilfe Sie die Bereitstellung einer Automatisierung in der gesamten Flotte durch Angabe eines Nebenläufigkeits- und eines Fehlerschwellenwerts steuern können. Weitere Informationen zum Arbeiten mit Ratensteuerelementen finden Sie unter Automatisierte Abläufe in großem Umfang ausführen.

- Optimieren komplexer Aufgaben

  Automation bietet vordefinierte Runbooks, die komplexe und zeitaufwendige Aufgaben wie das Erstellen von goldenen Amazon Machine Images (AMIs) optimieren. Mit den Runbooks `AWS-UpdateLinuxAmi` und `AWS-UpdateWindowsAmi` können Sie zum Beispiel goldene AMIs aus einem Quell-AMI erstellen. Mithilfe dieser Runbooks können Sie benutzerdefinierte Skripte ausführen, bevor und nachdem Updates angewendet werden. Zudem können Sie bestimmte Softwarepakete in die Installation einbeziehen oder daraus ausschließen. Beispiele für die Ausführung dieser Runbooks finden Sie unter Tutorials.

- Definieren von Einschränkungen für Eingaben

  In benutzerdefinierten Runbooks können Einschränkungen definiert werden, um die Werte einzugrenzen, die Automation für einen bestimmten Eingabeparameter akzeptiert. `allowedPattern` zum Beispiel akzeptiert für einen Eingabeparameter nur Werte, die dem von Ihnen definierten regulären Ausdruck entsprechen. Wenn Sie `allowedValues` für einen Eingabeparameter angeben, werden nur die Werte akzeptiert, die Sie im Runbook angegeben haben.

- Ausgabe von Aktionen zur Protokollautomatisierung in Amazon CloudWatch Logs

  Zur Erfüllung betriebs- oder sicherheitsbezogener Anforderungen in Ihrer Organisation müssen Sie möglicherweise eine Aufzeichnung der während eines Runbooks ausgeführten Skripte bereitstellen. Mit CloudWatch Logs können Sie Protokolldateien aus verschiedenen Quellen überwachen, speichern und darauf zugreifen AWS-Services. Sie können die Ausgabe der `aws:executeScript` Aktion zu Debugging- und Fehlerbehebungszwecken an eine CloudWatch Protokollgruppe „Logs“ senden. Protokolldaten können mit oder ohne AWS KMS Verschlüsselung mithilfe Ihres KMS-Schlüssels an Ihre Protokollgruppe gesendet werden. Weitere Informationen finden Sie unter Protokollierung der Automation-Aktionsausgabe mit CloudWatch Logs.

- EventBridge Amazon-Integration

Automatisierung wird in EventBridge Amazon-Regeln als Zieltyp unterstützt. Das bedeutet, dass Sie Runbooks mithilfe von Ereignissen auslösen können. Weitere Informationen erhalten Sie unter Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge und Referenz: EventBridge Amazon-Ereignistypen und -muster für Systems Manager.

- Gemeinsame Nutzung von bewährten Methoden für Organisationen

  Sie können bewährte Methoden u. a. für das Ressourcenmanagement und Betriebsaufgaben in Runbooks definieren, die Sie in mehreren Konten und Regionen nutzen.

Wer sollte Automation nutzen?

- Jeder AWS Kunde, der seine betriebliche Effizienz in großem Umfang verbessern, Fehler im Zusammenhang mit manuellen Eingriffen reduzieren und die Zeit bis zur Lösung häufig auftretender Probleme verkürzen möchte.
- Infrastrukturexperten, die Bereitstellungs- und Konfigurationsaufgaben automatisieren möchten.
- Administratoren, die häufig auftretende Probleme zuverlässig lösen, die Effizienz bei der Fehlerbehebung verbessern und die Anzahl sich wiederholender Vorgänge reduzieren möchten.
- Benutzer, die eine Aufgabe automatisieren möchten, die sie normalerweise manuell ausführen.

Was ist eine Automatisierung?

Eine Automatisierung besteht aus allen Aufgaben, die in einem Runbook definiert sind und vom Automation-Service ausgeführt werden. Automation nutzt die folgenden Komponenten zur Ausführung von Automatisierungen.

Konzept	Details
Automation-Runbook	Ein Systems Manager Automation-Runbook definiert die Automatisierung (die Aktionen, die Systems Manager auf Ihren verwalteten Knoten und AWS Ressourcen ausführt). Die Automatisierung umfasst mehrere vordefinierte Runbooks, mit denen Sie allgemeine Aufgaben wie den Neustart einer oder mehrerer EC2 Amazon-Instances oder das Erstellen einer

Konzept	Details
	Amazon Machine Image () ausführen können. AMI Sie können auch eigene Runbooks erstellen. Die Runbooks liegen im YAML- oder JSON-Format vor und enthalten die von Ihnen angegebenen Schritte und Parameter. Die Schritte werden nacheinander ausgeführ t. Weitere Informationen finden Sie unter Erstellen Ihrer eigenen Runbooks. Runbooks sind Systems Manager-Dokumente vom Typ `Automation` , im Gegensatz zu `Command`, `Policy`, `Session`-Dokumenten. Runbooks unterstützen die Schemaversion 0.3. Befehlsdokumente mit Schema-Version 1.2, 2.0 oder 2.2. Richtliniendokumente verwenden die Schemaversion 2.0 oder höher.
Automation-Aktion	Die in einem Runbook definierte Automatis ierung umfasst einen oder mehrere Schritte. Jeder Schritt ist einer bestimmten Aktion zugeordnet. Die Aktion bestimmt die Eingaben, das Verhalten und die Ausgaben des Schritts. Die Schritte sind im `mainSteps` -Bereich Ihres Runbooks definiert. Die Automatisierung unterstützt 20 verschiedene Aktionstypen. Weitere Informationen hierzu finden Sie unter Systems Manager Automation Aktionen-Referenz.

Konzept	Details
Automation-Kontingente	Jeder AWS-Konto kann 100 Automatisierungen gleichzeitig ausführen. Dazu gehören untergeordnete Automatisierungen (Automatisierungen, die durch eine andere Automatisierung gestartet werden) und Automatisierungen der Ratenregelung. Wenn Sie versuchen, mehr Automatisierungen auszuführen, fügt Systems Manager die zusätzlichen Automatisierungen zu einer Warteschlange hinzu und zeigt den Status „Pending“ an. Dieses Kontingent kann mithilfe von adaptiver Nebenläufigkeit angepasst werden. Weitere Informationen finden Sie unterZulassen, dass sich Automation an Ihre Nebenläufigkeitsanforderungen anpasst. Informationen zur Ausführung von Automatisierungen finden Sie unter Führen Sie einen automatisierten Vorgang aus, der von Systems Manager Automation unterstützt wird.
Automatisierungs-Warteschlange	Wenn Sie versuchen, mehr Automatisierungen als das gleichzeitige Automatisierungslimit auszuführen, werden nachfolgende Automatisierungen zu einer Warteschlange hinzugefügt. Jedes AWS-Konto kann 5 000 Automatisierungen in die Warteschlange stellen. Sobald eine Automatisierung abgeschlossen ist (oder einen Terminalstatus erreicht), beginnt die erste Automatisierung in der Warteschlange.

Konzept	Details
Kontingent für Automatisierungen der Ratenregelung	Jeder AWS-Konto kann 25 Automationen zur Ratensteuerung gleichzeitig ausführen. Wenn Sie versuchen, mehr Automatisierungen der Ratenregelung als das gleichzeitige Limit durchzuführen, fügt Systems Manager der Warteschlange die nachfolgenden Automatisierungen der Ratenregelung hinzu und zeigt den Status „Pending“ . Weitere Informationen über die Ratenregelung-Automatisierungen finden Sie unter Automatisierte Abläufe in großem Umfang ausführen.
Kontingent für Automatisierungs-Warteschlange der Ratenregelung	Wenn Sie versuchen, mehr Automatisierungen als das Limit für gleichzeitige Automatisierungen der Ratenregelung auszuführen, werden nachfolgende Automatisierungen zu einer Warteschlange hinzugefügt. Jeder AWS-Konto kann 1.000 Automationen zur Ratensteuerung in die Warteschlange stellen. Sobald eine Automatisierung abgeschlossen ist (oder einen Terminalstatus erreicht), beginnt die erste Automatisierung in der Warteschlange.

Themen

- Einrichten der Automatisierung
- Führen Sie einen automatisierten Vorgang aus, der von Systems Manager Automation unterstützt wird
- Wiederholtes Ausführen von Automatisierungsausführungen
- Eine Automatisierung ausführen, für die Genehmigungen erforderlich sind
- Automatisierte Abläufe in großem Umfang ausführen
- Automatisierungen in mehreren Konten AWS-Regionen ausführen
- Führen Sie Automatisierungen auf EventBridge der Grundlage von Ereignissen aus

- Ausführen einer Automatisierung Schritt für Schritt
- Planen von Automatisierungen mit State Manager-Zuordnungen
- Planen von Automatisierungen mit Wartungsfenstern
- Systems Manager Automation Aktionen-Referenz
- Erstellen Ihrer eigenen Runbooks
- Runbook-Referenz für Systems Manager Automation
- Tutorials
- Erfahren Sie mehr über die von Systems Manager Automation zurückgegebenen Status
- Fehlerbehebung für Systems Manager Automation.

Einrichten der Automatisierung

Um Automation, ein Tool in, einzurichten AWS Systems Manager, müssen Sie den Benutzerzugriff auf den Automationsdienst überprüfen und Rollen situationsabhängig konfigurieren, damit der Dienst Aktionen an Ihren Ressourcen ausführen kann. Außerdem empfiehlt es sich, in den Automation-Einstellungen den adaptiven Nebenläufigkeitsmodus zu aktivieren. Die adaptive Nebenläufigkeit passt Ihr Automatisierungskontingent automatisch an Ihre Anforderungen an. Weitere Informationen finden Sie unter Zulassen, dass sich Automation an Ihre Nebenläufigkeitsanforderungen anpasst.

Um einen ordnungsgemäßen Zugriff auf AWS Systems Manager Automation sicherzustellen, sollten Sie die folgenden Anforderungen an Benutzer- und Servicerollen überprüfen.

Überprüfen des Benutzerzugriffs für Runbooks

Stellen Sie sicher, dass Sie berechtigt sind, Runbooks zu verwenden. Wenn Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle Administratorrechte zugewiesen sind, haben Sie Zugriff auf Systems Manager Automation. Wenn Sie nicht über Administratorrechte verfügen, muss ein Administrator Ihnen die Berechtigung gewähren, indem er die von `AmazonSSMFullAccess` verwaltete Richtlinie oder eine Richtlinie, die vergleichbare Berechtigungen bereitstellt, Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle zuweist.

> **Important**
>
> Die IAM-Richtlinie `AmazonSSMFullAccess` erteilt Berechtigungen für Systems Manager Aktionen. Einige Runbooks erfordern jedoch Berechtigungen für andere Services, z. B. das

Runbook `AWS-ReleaseElasticIP`, das IAM-Berechtigungen für `ec2:ReleaseAddress` erfordert. Daher müssen Sie die in einem Runbook ausgeführten Aktionen überprüfen, um sicherzustellen, dass Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle die erforderlichen Berechtigungen zum Ausführen der im Runbook enthaltenen Aktionen zugewiesen sind.

Konfigurieren eines Service-Rollenzugriffs (Rolle übernehmen) für Automatisierungen

Automation kann im Kontext einer Service-Rolle initiiert werden (oder Übernahmerolle). Auf diese Weise kann der Service Aktionen in Ihrem Namen ausführen. Wenn Sie keine Übernahmerolle angeben, verwendet Automation den Kontext des Benutzers, der die Automatisierung aufgerufen hat.

In den folgenden Situationen müssen Sie jedoch eine Servicerolle für Automation angeben:

- Wenn Sie die Zugriffsberechtigungen eines Benutzers für eine Ressource einschränken, aber dem Benutzer die Ausführung einer Automatisierung gestatten möchten, der höhere Berechtigungen erfordert. In diesem Szenario können Sie eine Servicerolle mit höheren Berechtigungen erstellen und dem Benutzer das Ausführen der Automatisierung gestatten.
- Wenn Sie eine Systems Manager State Manager-Zuordnung zum Ausführen eines Runbooks erstellen.
- Wenn Sie Vorgänge haben, die voraussichtlich länger als 12 Stunden ausgeführt werden.
- Wenn Sie ein Runbook ausführen, das nicht Amazon gehört und die `aws:executeScript` Aktion verwendet, um eine AWS API-Operation aufzurufen oder auf eine AWS Ressource zu reagieren. Weitere Informationen finden Sie unter Berechtigungen für die Verwendung von Runbooks.

Wenn Sie eine Servicerolle für Automation erstellen müssen, können Sie eine der folgenden Methoden anwenden.

Themen

- Erstellen Sie Servicerollen für die Automatisierung mithilfe von CloudFormation
- Erstellen Sie die Servicerollen für Automation mithilfe der Konsole
- Beispiele für die Einrichtung identitätsbasierter Richtlinien
- Zulassen, dass sich Automation an Ihre Nebenläufigkeitsanforderungen anpasst
- Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge
- Implementieren von Änderungskontrollen für Automatisierung

Erstellen Sie Servicerollen für die Automatisierung mithilfe von CloudFormation

Sie können eine Servicerolle für Automation, ein Tool in AWS Systems Manager, aus einer AWS CloudFormation Vorlage erstellen. Nachdem Sie die Servicerolle erstellt haben, können Sie die Servicerolle in Runbooks mit dem Parameter `AutomationAssumeRole` angeben.

Erstellen der Servicerolle mit CloudFormation

Gehen Sie wie folgt vor, um die erforderliche Rolle AWS Identity and Access Management (IAM) für Systems Manager Automation zu erstellen, indem Sie CloudFormation.

Erstellen der erforderlichen IAM-Rolle

1. Laden Sie die `AWS-SystemsManager-AutomationServiceRole.zip`-Datei herunter und entpacken Sie diese. Diese Datei enthält die `AWS-SystemsManager-AutomationServiceRole.yaml` CloudFormation Vorlagendatei.
2. Öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
3. Wählen Sie Stapel erstellen aus.
4. Wählen Sie im Abschnitt Specify template (Vorlage angeben) die Option Upload a template file (Vorlagendatei hochladen) aus.
5. Wählen Sie Durchsuchen und wählen Sie dann die `AWS-SystemsManager-AutomationServiceRole.yaml` CloudFormation Vorlagendatei aus.
6. Wählen Sie Weiter aus.
7. Geben Sie auf der Seite Stack-Details angeben im Feld Stack-Name einen Namen ein.
8. Auf der Seite Configure stack options (Stack-Optionen konfigurieren) müssen Sie keine Auswahl treffen. Wählen Sie Weiter aus.
9. Scrollen Sie auf der Seite „Überprüfen“ nach unten und wählen Sie die Option Ich bestätige, dass CloudFormation möglicherweise IAM-Ressourcen erstellt werden.
10. Wählen Sie Erstellen aus.

CloudFormation zeigt den Status CREATE_IN_PROGRESS für ungefähr drei Minuten an. Der Status wird in CREATE_COMPLETE geändert, sobald der Stack erstellt wurde und die Rollen verwendet werden können.

Important

Wenn Sie einen automatisierten Workflow ausführen, der andere Services mithilfe einer AWS Identity and Access Management -(IAM)-Servicerolle aufruft, muss die Servicerolle mit der Berechtigung zum Aufrufen dieser Services konfiguriert sein. Diese Anforderung gilt für alle AWS Automatisierungs-Runbooks (`AWS-*`Runbooks) wie die`AWS-ConfigureS3BucketLogging`,, und `AWS-RestartEC2Instance` Runbooks`AWS-CreateDynamoDBBackup`, um nur einige zu nennen. Diese Anforderung gilt auch für alle benutzerdefinierten Automatisierungs-Runbooks, die Sie erstellen und die andere mithilfe AWS-Services von Aktionen aufrufen, die andere Dienste aufrufen. Wenn Sie unter anderem `aws:executeAwsApi`-, `aws:createStack`- oder `aws:copyImage`-Aktionen verwenden, konfigurieren Sie die Dienstrolle mit der Berechtigung zum Aufrufen solcher Services.
Sie können anderen AWS-Services Berechtigungen erteilen, indem Sie der Rolle eine eingebundene IAM-Richtlinie hinzufügen. Weitere Informationen finden Sie unter (Optional) Fügen Sie eine Inline-Automatisierungsrichtlinie oder eine vom Kunden verwaltete Richtlinie hinzu, um andere aufzurufen AWS-Services.

Kopieren von Rolleninformationen für Automation

Gehen Sie wie folgt vor, um Informationen über die Automations-Servicerolle aus der CloudFormation Konsole zu kopieren. Sie müssen diese Rollen beim Verwenden eines Runbooks festlegen.

Note

Sie müssen keine Rolleninformationen mit diesen Schritten kopieren, wenn Sie die Runbooks `AWS-UpdateLinuxAmi` oder `AWS-UpdateWindowsAmi` ausführen. In diesen Runbook sind die erforderlichen Rollen bereits als Standardwerte festgelegt. Die Rollen in diesen Runbooks verwenden von IAM verwaltete Richtlinien.

Kopieren der Rollennamen

1. Öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie den Stack name (Stack-Name) der Automation aus, den Sie im vorherigen Verfahren erstellt haben.
3. Wählen Sie die Registerkarte Resources (Ressourcen) aus.

4. Wählen Sie den Link Physical ID für. AutomationServiceRole Beim Öffnen der IAM-Konsole wird eine Zusammenfassung der Servicerolle für die Automatisierung angezeigt.
5. Kopieren Sie den Amazon-Ressourcennamen (ARN) neben Role ARN (Rollen-ARN). Der ARN ist ähnlich wie der folgende: `arn:aws:iam::12345678:role/AutomationServiceRole`
6. Kopieren Sie den ARN zur späteren Verwendung in eine Textdatei.

Sie haben die Konfiguration der Automation-Servicerolle abgeschlossen. Sie können jetzt den ARN der Automation-Servicerolle in Ihren Runbooks verwenden.

Erstellen Sie die Servicerollen für Automation mithilfe der Konsole

Wenn Sie eine Servicerolle für Automation, ein Tool in, erstellen müssen AWS Systems Manager, führen Sie die folgenden Aufgaben aus. Weitere Informationen darüber, wann eine Servicerolle für Automation erforderlich ist, finden Sie unter Einrichten der Automatisierung.

Aufgaben

- Aufgabe 1: Erstellen einer Servicerolle für Automation
- Aufgabe 2: Hängen Sie die iam: PassRole -Richtlinie an Ihre Automatisierungsrolle an

Aufgabe 1: Erstellen einer Servicerolle für Automation

Führen Sie die folgenden Schritte zum Erstellen einer Service-Rolle (oder Übernahmerolle) für Systems Manager Automation.

> **Note**
>
> Sie können diese Rolle auch in Runbooks, wie dem `AWS-CreateManagedLinuxInstance`-Runbook, verwenden. Wenn Sie diese Rolle oder den Amazon-Ressourcennamen (ARN) einer AWS Identity and Access Management (IAM) -Rolle in Runbooks verwenden, kann Automation Aktionen in Ihrer Umgebung ausführen, z. B. neue Instances starten und Aktionen in Ihrem Namen ausführen.

Erstellen einer IAM-Rolle und Gestatten der Automatisierung

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).

3. Wählen Sie unter Select type of trusted entity (Typ der vertrauenswürdigen Entität auswählen) die Option AWS -Service aus.
4. Wählen Sie im Abschnitt Choose a use case (Anwendungsfall auswählen) die Option Systems Manager und wählen Sie dann Next: Permissions (Weiter: Berechtigungen).
5. Suchen Sie auf der Seite Angehängte Berechtigungsrichtlinie nach der SSMAutomationAmazon-Rollenrichtlinie, wählen Sie sie aus und klicken Sie dann auf Weiter: Überprüfen.
6. Geben Sie auf der Seite Review im Feld Role name einen Namen und anschließend eine Beschreibung ein.
7. Wählen Sie Create role (Rolle erstellen) aus. Das System leitet Sie zur Seite Rollen zurück.
8. Wählen Sie auf der Seite Roles (Rollen) die gerade erstellte Rolle aus, um die Seite Summary (Übersicht) zu öffnen. Notieren Sie sich den Role Name (Rollenname) und Role ARN (Rollen-ARN). Sie geben den Rollen-ARN an, wenn Sie im nächsten Verfahren die iam: PassRole - Richtlinie an Ihr IAM-Konto anhängen. Sie können den Rollennamen und den ARN in Runbooks festlegen.

> **Note**
>
> Die `AmazonSSMAutomationRole` Richtlinie weist die Automatisierungs-Rollenberechtigung einer Teilmenge von AWS Lambda Funktionen in Ihrem Konto zu. Diese Funktionen beginnen mit „Automation“ (Automatisierung). Wenn Sie die Automatisierung mit Lambda-Funktionen verwenden möchten, muss der Lambda-ARN das folgende Format verwenden: `"arn:aws:lambda:*:*:function:Automation*"`
> Wenn Sie über bestehende Lambda-Funktionen verfügen, die dieses Format ARNs nicht verwenden, müssen Sie Ihrer Automatisierungsrolle auch eine zusätzliche Lambda-Richtlinie hinzufügen, z. B. die AWSLambdaRollenrichtlinie. Die zusätzliche Richtlinie oder Rolle muss umfassendere Zugriffsberechtigungen für Lambda-Funktionen im AWS-Konto bieten.

Nachdem Sie Ihre Servicerolle erstellt haben, sollten Sie die Vertrauensrichtlinie bearbeiten, um das serviceübergreifende Confused-Deputy-Problem zu vermeiden. Das Confused-Deputy-Problem ist ein Sicherheitsproblem, bei dem eine Entität, die nicht über die Berechtigung zum Ausführen einer Aktion verfügt, eine Entität mit größeren Rechten zwingen kann, die Aktion auszuführen. In der AWS Tat kann ein dienstübergreifender Identitätswechsel zu einem Problem mit dem verwirrten Stellvertreter führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der Anruf-Dienst kann so

manipuliert werden, dass er seine Berechtigungen verwendet, um auf die Ressourcen eines anderen Kunden zu reagieren, auf die er sonst nicht zugreifen dürfte. Um dies zu verhindern, AWS bietet Tools, mit denen Sie Ihre Daten für alle Dienste mit Dienstprinzipalen schützen können, denen Zugriff auf Ressourcen in Ihrem Konto gewährt wurde.

Wir empfehlen die Verwendung der globalen Bedingungskontext-Schlüssel `aws:SourceArn` und `aws:SourceAccount` in ressourcenbasierten Richtlinien, um die Berechtigungen, die Automation einem anderen Service erteilt, auf eine bestimmte Ressource zu beschränken. Wenn der `aws:SourceArn`-Wert nicht die Konto-ID enthält, z. B. den ARN eines Amazon-S3-Buckets, müssen Sie beide globalen Bedingungskontext-Schlüssel verwenden, um Berechtigungen einzuschränken. Wenn Sie beide globale Bedingungskontextschlüssel verwenden und der `aws:SourceArn`-Wert die Konto-ID enthält, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in der gleichen Richtlinienanweisung verwendet wird. Verwenden Sie `aws:SourceArn`, wenn Sie nur eine Ressource mit dem betriebsübergreifenden Zugriff verknüpfen möchten. Verwenden Sie `aws:SourceAccount`, wenn Sie zulassen möchten, dass Ressourcen in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft werden. Der Wert von `aws:SourceArn` muss für Automatisierungsausführungen der ARN sein. Wenn Sie den vollständigen ARN der Ressource nicht kennen oder wenn Sie mehrere Ressourcen angeben, verwenden Sie den globalen Bedingungskontext-Schlüssel `aws:SourceArn` mit Platzhaltern (*) für die unbekannten Teile des ARN. Beispiel, `arn:aws:ssm:*:123456789012:automation-execution/*`.

Das folgende Beispiel zeigt, wie Sie die `aws:SourceArn` und `aws:SourceAccount` globale Bedingungskontext-Schlüssel für Automatisierung verwenden können, um das Confused-Deputy-Problem zu verhindern.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "ssm.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
```

```
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:ssm:*:123456789012:automation-execution/*"
        }
      }
    }
  ]
}
```

So ändern Sie die Vertrauensrichtinie einer Rolle

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen.
3. Wählen Sie in der Rollenliste in Ihrem Konto den Namen der Automation-Servicerolle aus.
4. Klicken Sie auf der Registerkarte Trust Relationships (Vertrauensbeziehungen) auf Edit Trust Relationship (Vertrauensbeziehungen bearbeiten).
5. Bearbeiten Sie die Vertrauensrichtlinie mit den globalen Bedingungskontext-Schlüsseln `aws:SourceArn` und `aws:SourceAccount` für Automation, um das Confused-Deputy-Problem zu verhindern.
6. Wählen Sie Update Trust Policy (Vertrauensrichtlinie aktualisieren) aus, um die Änderungen zu speichern.

(Optional) Fügen Sie eine Inline-Automatisierungsrichtlinie oder eine vom Kunden verwaltete Richtlinie hinzu, um andere aufzurufen AWS-Services

Wenn Sie eine Automatisierung ausführen, die andere Dienste AWS-Services mithilfe einer IAM-Servicerolle aufruft, muss die Servicerolle so konfiguriert sein, dass sie berechtigt ist, diese Dienste aufzurufen. Diese Anforderung gilt für alle AWS Automatisierungs-Runbooks (`AWS-*`Runbooks) wie, und `AWS-RestartEC2Instance` Runbooks `AWS-ConfigureS3BucketLoggingAWS-CreateDynamoDBBackup`, um nur einige zu nennen. Diese Anforderung gilt auch für alle von Ihnen erstellten benutzerdefinierten Runbooks, die andere AWS-Services aufrufen, indem sie Aktionen verwenden, die andere Services aufrufen. Wenn Sie unter anderem `aws:executeAwsApi`-, `aws:CreateStack`- oder `aws:copyImage`-Aktionen verwenden, dann müssen Sie die Servicerolle mit der Berechtigung zum Aufrufen solcher Services konfigurieren. Sie können anderen Benutzern

Berechtigungen erteilen, AWS-Services indem Sie der Rolle eine IAM-Inline-Richtlinie oder eine vom Kunden verwaltete Richtlinie hinzufügen.

So betten Sie eine eingebundene Richtlinie für eine Servicerolle ein (IAM-Konsole)

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im Navigationsbereich Rollen.
3. Wählen Sie in der Liste den Namen der Rolle aus, die Sie bearbeiten möchten.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie in der Dropdown-Liste Berechtigungen hinzufügen die Option Richtlinien anhängen oder Inline-Richtlinie erstellen.
6. Wenn Sie die Option Richtlinien anhängen wählen, aktivieren Sie das Kontrollkästchen neben der Richtlinie, die Sie hinzufügen möchten, und wählen Sie Berechtigungen hinzufügen.
7. Wenn Sie Inline-Richtlinie erstellen wählen, wählen Sie die Registerkarte JSON.
8. Geben Sie ein JSON-Richtliniendokument für das Dokument ein AWS-Services , das Sie aufrufen möchten. Nachfolgend sind zwei Beispiele für JSON-Richtliniendokumente aufgeführt.

   Amazon S3 PutObject und GetObject Beispiel

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Amazon EC2 CreateSnapshot und DescribeSnapShots Beispiel

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":"ec2:DescribeSnapshots",
         "Resource":"*"
      }
   ]
}
```

Details zur IAM-Richtliniensprache und finden Sie in der IAM JSON Policy Reference im IAM-Benutzerhandbuch.

9. Wählen Sie, wenn Sie fertig sind, Review policy (Richtlinie überprüfen) aus. Die Richtlinienvalidierung meldet mögliche Syntaxfehler.
10. Geben Sie auf der Seite Review Policy (Richtlinie überprüfen) im Feld Name (Name) einen Namen für die zu erstellende Richtlinie ein. Überprüfen Sie unter Summary die Richtlinienzusammenfassung, um die Berechtigungen einzusehen, die von Ihrer Richtlinie gewährt werden. Wählen Sie dann Create policy aus, um Ihre Eingaben zu speichern.
11. Nachdem Sie eine Inline-Richtlinie erstellt haben, wird sie automatisch in Ihre Rolle eingebettet.

Aufgabe 2: Hängen Sie die iam: PassRole -Richtlinie an Ihre Automatisierungsrolle an

Fügen Sie mit den folgenden Schritten die Richtlinie `iam:PassRole` Ihrer Automation-Servicerolle hinzu. Dies erlaubt dem Automation-Service, die Rolle anderen Services oder Systems-Manager-Tools zu übergeben, wenn Automatisierungen ausgeführt werden.

So hängen Sie die iam: PassRole -Richtlinie an Ihre Automatisierungsrolle an

1. Wählen Sie auf der Seite Summary für die gerade erstellte Rolle die Registerkarte Permissions.
2. Wählen Sie Inline-Richtlinie hinzufügen.
3. Wählen Sie auf der Seite Richtlinie erstellen die Registerkarte Visueller Editor aus.
4. Wählen Sie Service (Service) und anschließend die Option IAM aus.
5. Wählen Sie Select actions (Aktionen auswählen) aus.
6. Geben Sie in das Textfeld Aktionen filtern die PassRoleOption ein**`PassRole`**, und wählen Sie sie dann aus.
7. Wählen Sie Resources aus. Stellen Sie sicher, dass Specific ausgewählt ist und wählen Sie dann Add ARN aus.
8. Fügen Sie im Feld Specify ARN for role (ARN für die Rolle angeben) den ARN der Automation-Rolle ein, den Sie am Ende von Aufgabe 1 kopiert haben. Das System füllt die Felder Account (Konto) und Role name with path (Rollenname mit Pfad) automatisch aus.

   > **Note**
   >
   > Wenn Sie möchten, dass die Automatisierungsdienst-Rolle einer Instanz eine IAM-Instanzprofilrolle anhängt, müssen Sie den ARN der IAM-Instanzprofilrolle hinzufügen. EC2 Auf diese Weise kann die Automatisierungsdienst-Rolle die IAM-Instanzprofilrolle an die Zielinstanz übergeben. EC2

9. Wählen Sie Hinzufügen aus.
10. Wählen Sie Richtlinie prüfen.
11. Geben Sie auf der Seite Review Policy einen Namen ein und wählen Sie anschließend Create Policy aus.

Beispiele für die Einrichtung identitätsbasierter Richtlinien

Die folgenden Abschnitte enthalten Beispiele für identitätsbasierte IAM-Richtlinien für den Automationsdienst. AWS Systems Manager Weitere Informationen zum Erstellen einer identitätsbasierten IAM-Richtlinie mithilfe dieser Beispieldokumente zu JSON-Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

> **Note**
>
> Alle Beispiele enthalten ein fiktives Konto. IDs Die Konto-ID sollte nicht im Amazon-Ressourcennamen (ARN) für AWS eigene öffentliche Dokumente angegeben werden.

Beispiele

- Beispiel 1: Erlauben Sie einem Benutzer, ein Automatisierungsdokument auszuführen und sich die Ausführung der Automatisierung anzusehen
- Beispiel 2: Erlauben Sie einem Benutzer, eine bestimmte Version eines Automatisierungsdokuments auszuführen
- Beispiel 3: Erlauben Sie einem Benutzer, Automatisierungsdokumente mit einem bestimmten Tag auszuführen
- Beispiel 4: Erlauben Sie einem Benutzer, ein Automatisierungsdokument auszuführen, wenn ein bestimmter Tag-Parameter für die Automatisierungsausführung bereitgestellt wird

Beispiel 1: Erlauben Sie einem Benutzer, ein Automatisierungsdokument auszuführen und sich die Ausführung der Automatisierung anzusehen

Mit der folgenden IAM-Beispielrichtlinie kann ein Benutzer folgende Aktionen ausführen:

- Führen Sie das in der Richtlinie angegebene Automatisierungsdokument aus. Der Name des Dokuments wird durch den folgenden Eintrag bestimmt.

```
arn:aws:ssm:*:111122223333:document/{{DocumentName}}
```

- Stoppen und senden Sie Signale an eine Automatisierungsausführung.
- Sehen Sie sich Details zur Automatisierungsausführung an, nachdem sie gestartet wurde.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": "ssm:StartAutomationExecution",
```

```
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ssm:*:111122223333:document/{{DocumentName}}",
                "arn:aws:ssm:*:111122223333:automation-execution/*"
            ]
        },
        {
            "Action": [
                "ssm:StopAutomationExecution",
                "ssm:GetAutomationExecution",
                "ssm:DescribeAutomationExecutions",
                "ssm:DescribeAutomationStepExecutions",
                "ssm:SendAutomationSignal"
            ],
            "Resource": [
                "arn:aws:ssm:*:111122223333:automation-execution/*"
            ],
            "Effect": "Allow"
        }
    ]
}
```

Beispiel 2: Erlauben Sie einem Benutzer, eine bestimmte Version eines Automatisierungsdokuments auszuführen

Das folgende Beispiel für eine IAM-Richtlinie gewährt einem Benutzer die Erlaubnis, die Automatisierungsausführung mithilfe einer bestimmten Version eines Automatisierungsdokuments auszuführen:

- Der Name des Automatisierungsdokuments wird durch den folgenden Eintrag bestimmt.

```
arn:aws:ssm:*:111122223333:document/{{DocumentName}}
```

- Die Version des Automatisierungsdokuments wird durch den folgenden Eintrag bestimmt.

```
"ssm:DocumentVersion": "5"
```

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": "ssm:StartAutomationExecution",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ssm:*:111122223333:document/{{DocumentName}}"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                   "ssm:DocumentVersion": ["5"]
                }
            }
        },
        {
            "Action": [
                "ssm:StartAutomationExecution"
            ],
            "Resource": [
                "arn:aws:ssm:*:111122223333:automation-execution/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "ssm:StopAutomationExecution",
                "ssm:GetAutomationExecution",
                "ssm:DescribeAutomationExecutions",
                "ssm:DescribeAutomationStepExecutions",
                "ssm:SendAutomationSignal"
            ],
            "Resource": [
                "arn:aws:ssm:*:111122223333:automation-execution/*"
            ],
            "Effect": "Allow"
        }
    ]
}
```

Beispiel 3: Erlauben Sie einem Benutzer, Automatisierungsdokumente mit einem bestimmten Tag auszuführen

Das folgende Beispiel für eine IAM-Richtlinie ermöglicht es einem Benutzer, die Automatisierungsausführung mithilfe eines beliebigen Automatisierungsdokuments auszuführen, das über ein bestimmtes Tag verfügt:

- Der Name des Automatisierungsdokuments wird durch den folgenden Eintrag bestimmt.

```
arn:aws:ssm:*:111122223333:document/{{DocumentName}}
```

- Das Tag des Automatisierungsdokuments wird durch den folgenden Eintrag bestimmt.

```
"ssm:DocumentVersion": "5"
```

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": "ssm:StartAutomationExecution",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ssm:*:111122223333:document/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/stage": "production"
                }
            }
        },
        {
            "Action": [
                "ssm:StartAutomationExecution"
            ],
            "Resource": [
                "arn:aws:ssm:*:111122223333:automation-execution/*"
            ],
            "Effect": "Allow"
        },
```

```
        {
            "Action": [
                "ssm:StopAutomationExecution",
                "ssm:GetAutomationExecution",
                "ssm:DescribeAutomationExecutions",
                "ssm:DescribeAutomationStepExecutions",
                "ssm:SendAutomationSignal"
            ],
            "Resource": [
                "arn:aws:ssm:*:111122223333:automation-execution/*"
            ],
            "Effect": "Allow"
        }
    ]
}
```

Beispiel 4: Erlauben Sie einem Benutzer, ein Automatisierungsdokument auszuführen, wenn ein bestimmter Tag-Parameter für die Automatisierungsausführung bereitgestellt wird

Das folgende Beispiel für eine IAM-Richtlinie gewährt einem Benutzer die Erlaubnis, die Automatisierungsausführung mithilfe von Automatisierungsdokumenten auszuführen, wenn ein bestimmter Tag-Parameter für die Automatisierungsausführung bereitgestellt wird:

- Führen Sie das in der Richtlinie angegebene Automatisierungsdokument aus. Der Name des Dokuments wird durch den folgenden Eintrag bestimmt.

  ```
  arn:aws:ssm:*:111122223333:document/{{DocumentName}}
  ```

- Muss einen bestimmten Tag-Parameter für die Automatisierungsausführung angeben. Der Tag-Parameter für die Automatisierungsausführungsressource wird durch den folgenden Eintrag bestimmt.

  ```
  "aws:ResourceTag/stage": "production"
  ```

- Stoppt und sendet Signale an Automatisierungsausführungen, die über das angegebene Tag verfügen.
- Zeigt Details zu den Automatisierungsausführungen an, die das angegebene Tag haben.
- Fügt das angegebene Tag zu den SSM-Ressourcen hinzu.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": "ssm:StartAutomationExecution",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ssm:*:111122223333:document/{{DocumentName}}"
            ]
        },
        {
            "Action": [
                "ssm:StartAutomationExecution",
                "ssm:StopAutomationExecution",
                "ssm:GetAutomationExecution",
                "ssm:DescribeAutomationExecutions",
                "ssm:DescribeAutomationStepExecutions",
                "ssm:SendAutomationSignal"
            ],
            "Resource": [
                "arn:aws:ssm:*:111122223333:automation-execution/*"
            ],
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/environment": "beta"
                }
            }
        },
        {
            "Action": "ssm:AddTagsToResource",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ssm:*:111122223333:automation-execution/*"
            ]
        }
    ]
}
```

Zulassen, dass sich Automation an Ihre Nebenläufigkeitsanforderungen anpasst

Standardmäßig können Sie mit Automation bis zu 100 nebenläufige Automatisierungen gleichzeitig ausführen. Automation bietet zudem eine optionale Einstellung, mit der Sie Ihr Kontingent für nebenläufige Automatisierungen automatisch anpassen können. Mit dieser Einstellung kann Ihr Kontingent je nach verfügbaren Ressourcen bis zu 500 nebenläufige Automatisierungen umfassen.

> **Note**
>
> Wenn Ihre Automatisierung API-Vorgänge aufruft, kann eine adaptive Skalierung entsprechend Ihren Zielen zu Drosselungsausnahmen führen. Wenn beim Ausführen von Automatisierungen mit aktivierter adaptiver Nebenläufigkeit wiederholt Drosselungsausnahmen auftreten, müssen Sie möglicherweise Kontingenterhöhungen für den API-Vorgang anfordern, sofern verfügbar.

So aktivieren Sie die adaptive Nebenläufigkeit mit der AWS-Managementkonsole

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation.
3. Wählen Sie die Registerkarte Präferenzen und anschließend Bearbeiten aus.
4. Aktivieren Sie das Kontrollkästchen neben Enable adaptive concurrency (Adaptive Nebenläufigkeit aktivieren).
5. Wählen Sie Speichern.

So aktivieren Sie die adaptive Nebenläufigkeit mit der Befehlszeile

- Öffnen Sie die AWS CLI oder die Tools für Windows PowerShell und führen Sie den folgenden Befehl aus, um die adaptive Nebenläufigkeit für Ihr Konto in der anfordernden Region zu aktivieren.

  Linux & macOS

  ```
  aws ssm update-service-setting \
      --setting-id /ssm/automation/enable-adaptive-concurrency \
      --setting-value True
  ```

Windows

```
aws ssm update-service-setting ^
    --setting-id /ssm/automation/enable-adaptive-concurrency ^
    --setting-value True
```

PowerShell

```
Update-SSMServiceSetting `
    -SettingId "/ssm/automation/enable-adaptive-concurrency" `
    -SettingValue "True"
```

Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge

Die Anzahl der gleichzeitigen Automatisierungen, die in jedem Konto ausgeführt werden können. Wenn Sie versuchen, mehrere Automatisierungen in einem Konto gleichzeitig auszuführen, kann dies zu Drosselungsproblemen führen. Sie können die automatische Drosselungswiederholung verwenden, um das Wiederholungsverhalten für gedrosselte Automatisierungsschritte zu konfigurieren.

Die automatische Drosselungswiederholung für Automatisierungsaktionen bietet eine zuverlässigere Ausführungsumgebung für umfangreiche Vorgänge. Die Drosselungswiederholung unterstützt alle Automatisierungsaktionen mit Ausnahme von `aws:executeScript`.

Die Drosselungswiederholung kann zusätzlich zur Schritteigenschaft `maxAttempts` verwendet werden. Wenn beide Funktionen konfiguriert sind, versucht das System zunächst, Drosselungswiederholungen innerhalb des angegebenen Zeitlimits einzuschränken, und wendet dann die `maxAttempts`-Einstellung an, wenn der Schritt weiterhin fehlschlägt.

Um die Drosselung zu konfigurieren, versuchen Sie es erneut mit dem AWS-Managementkonsole

1. Öffnen Sie die Konsole unter AWS Systems Manager . https://console.aws.amazon.com/systems-manager/
2. Klicken Sie im Navigationsbereich auf Automation.
3. Wählen Sie die Registerkarte Präferenzen und anschließend Bearbeiten aus.
4. Geben Sie im Feld Zeitlimit für die Drosselungswiederholung einen Wert zwischen 0 und 3 600 Sekunden ein. Dies gibt an, wie lange einer gedrosselter Schritt maximal wiederholt wird.

5. Wählen Sie Speichern.

Konfigurieren Sie die Drosselungswiederholung mit der Befehlszeile wie folgt:

- Öffnen Sie AWS CLI oder Tools für Windows PowerShell und führen Sie den folgenden Befehl aus, um die Drosselung der Wiederholungsversuche für Ihr Konto in der anfragenden Region zu konfigurieren.

  Linux & macOS

  ```
  aws ssm update-service-setting \
      --setting-id /ssm/automation/throttling-retry-time-limit \
      --setting-value 3600
  ```

  Windows

  ```
  aws ssm update-service-setting ^
      --setting-id /ssm/automation/throttling-retry-time-limit ^
      --setting-value 3600
  ```

  PowerShell

  ```
  Update-SSMServiceSetting `
      -SettingId "/ssm/automation/throttling-retry-time-limit" `
      -SettingValue "3600"
  ```

Implementieren von Änderungskontrollen für Automatisierung

Standardmäßig ermöglicht Automatisierung die Verwendung von Runbooks ohne Datums- und Zeitbeschränkungen. Durch die Integration der Automatisierung mit Change Calendar können Sie Änderungskontrollen für alle Automatisierungen in Ihrem AWS-Konto implementieren. Mit dieser Einstellung können AWS Identity and Access Management (IAM)-Prinzipale in Ihrem Konto Automatisierungen nur während der von Ihrem Änderungskalender zugelassenen Zeiträume ausführen. Weitere Informationen zum Arbeiten mit Change Calendar finden Sie unter Arbeiten mit Change Calendar.

So aktivieren Sie Änderungskontrollen (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation.
3. Wählen Sie die Registerkarte Präferenzen und anschließend Bearbeiten aus.
4. Aktivieren Sie das Kontrollkästchen neben Change Calendar-Integration aktivieren.
5. Wählen Sie in der Dropdown-Liste Änderungskalender auswählen den Änderungskalender aus, dem die Automatisierung folgen soll.
6. Wählen Sie Speichern.

Führen Sie einen automatisierten Vorgang aus, der von Systems Manager Automation unterstützt wird

Wenn Sie eine Automatisierung ausführen, wird die Automatisierung standardmäßig im Kontext des Benutzers ausgeführt, der die Automatisierung initiiert hat. Das bedeutet beispielsweise, wenn Ihr Benutzer über Administratorrechte verfügt, wird die Automatisierung mit Administratorrechten und vollständigem Zugriff auf die von der Automatisierung konfigurierten Ressourcen ausgeführt. Aus Sicherheitsgründen empfehlen wir, dass Sie die Automatisierung mithilfe einer IAM-Servicerolle ausführen, die in diesem Fall als Übernahmerolle bezeichnet wird und mit der Amazon SSMAutomation Role Managed Policy konfiguriert ist. Möglicherweise müssen Sie Ihrer angenommenen Rolle zusätzliche IAM-Richtlinien hinzufügen, um verschiedene Runbooks verwenden zu können. Die Verwendung einer IAM-Servicerolle zur Ausführung der Automatisierung wird als delegierte Administration bezeichnet.

Wenn Sie eine Servicerolle verwenden, darf die Automatisierung zwar für AWS -Ressourcen laufen, aber der Benutzer, der die Automatisierung ausgeführt hat, verfügt über einen eingeschränkten Zugriff (oder besitzt keinen Zugriff) auf diese Ressourcen. Sie können beispielsweise eine Servicerolle konfigurieren und sie mit Automation verwenden, um eine oder mehrere Amazon Elastic Compute Cloud (Amazon EC2) -Instances neu zu starten. Automation ist ein Tool in AWS Systems Manager. Die Automatisierung startet die Instances neu, aber die Servicerolle gibt dem Benutzer nicht die Berechtigung, auf diese Instances zuzugreifen.

Sie können eine Servicerolle zur Laufzeit angeben, wenn Sie eine Automatisierung ausführen, oder Sie können benutzerdefinierte Runbooks erstellen und die Servicerolle direkt im Runbook angeben. Wenn Sie zur Laufzeit oder in einem Runbook eine Servicerolle angeben, dann wird der Service im

Kontext der angegebenen Servicerolle ausgeführt. Wenn Sie keine Servicerolle angeben, dann legt das System im Kontext des Benutzers eine temporäre Sitzung an und führt die Automatisierung aus.

> **Note**
>
> Für Automatisierungen, die voraussichtlich länger als 12 Stunden laufen, müssen Sie eine Servicerolle angeben. Wenn Sie eine lang laufende Automatisierung im Kontext eines Benutzers starten, läuft die temporäre Sitzung des Benutzers nach 12 Stunden ab.

Delegierte Administration sorgt für mehr Sicherheit und Kontrolle Ihrer AWS -Ressourcen. Sie erlaubt auch eine verbesserte Prüfungserfahrung, da Aktionen für Ihre Ressourcen von einer zentralen Servicerolle statt von mehreren IAM-Konten ausgeführt werden.

Bevor Sie beginnen

Bevor Sie die folgenden Verfahren ausführen, müssen Sie die IAM-Servicerolle erstellen und eine Vertrauensbeziehung für Automation, ein Tool in AWS Systems Manager, konfigurieren. Weitere Informationen finden Sie unter Aufgabe 1: Erstellen einer Servicerolle für Automation.

In den folgenden Verfahren wird beschrieben, wie Sie die Systems-Manager-Konsole oder Ihr bevorzugtes Befehlszeilen-Tool zum Ausführen einer einfachen Automatisierung verwenden.

Ausführen einer einfachen Automatisierung (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie mithilfe der Systems Manager-Konsole eine einfache Automatisierung ausführen.

Ausführen einer einfachen Automatisierung

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Automatisierung und Automatisierung ausführen aus.
3. Wählen Sie in der Liste Automation-Dokument ein Runbook. Wählen Sie eine oder mehrere Optionen im Bereich Dokumentkategorien, um SSM-Dokumente nach ihrem Zweck zu filtern. Um ein Runbook anzuzeigen, das Sie besitzen, wählen Sie die Im Besitz von mir-Registerkarte. Um ein Runbook anzuzeigen, das für Ihr Konto freigegeben ist, wählen Sie die Registerkarte Mit mir geteilt. Um alle Runbooks anzuzeigen, wählen Sie die Registerkarte Alle Dokumente.

> **Note**
>
> Sie können Informationen zu einem Runbook einsehen, indem Sie den Runbook-Namen auswählen.

4. Überprüfen Sie im Abschnitt Dokument-Details, ob Dokumentversion auf die Version gesetzt ist, die Sie ausführen möchten. Das System bietet die folgenden Versionsoptionen:
   - Standardversion zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird und eine neue Standardversion zugewiesen ist.
   - Letzte Version zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird, und Sie die Version auszuführen möchten, die zuletzt aktualisiert wurde.
   - 1 (Standard) – Wählen Sie diese Option zur Ausführung der ersten Version des Dokuments, welches der Standard ist.
5. Wählen Sie Weiter aus.
6. Wählen Sie im Abschnitt Execution Mode (Ausführungsmodus) die Option Simple execution (Einfache Ausführung) aus.
7. Geben Sie im Abschnitt Eingabeparameter die erforderlichen Eingaben an. Optional können Sie eine IAM-Dienstrolle aus der AutomationAssumeRoleListe auswählen.
8. (Optional) Wählen Sie einen CloudWatch Alarm aus, der zur Überwachung auf Ihre Automatisierung angewendet werden soll. Um Ihrer Automatisierung einen CloudWatch Alarm hinzuzufügen, muss der IAM-Prinzipal, der die Automatisierung startet, über die entsprechende Genehmigung verfügen`iam:createServiceLinkedRole`. Weitere Informationen zu CloudWatch Alarmen finden Sie unter CloudWatch Amazon-Alarme verwenden. Beachten Sie, dass die Automatisierung gestoppt wird, wenn Ihr Alarm aktiviert wird. Wenn Sie dies verwenden AWS CloudTrail, wird der API-Aufruf in Ihrem Trail angezeigt.
9. Wählen Sie Ausführen.

Die Konsole zeigt den Status der Automatisierung an. Wenn Automatisierung nicht ausgeführt werden kann, finden Sie weitere Informationen unter Fehlerbehebung für Systems Manager Automation..

Nach Abschluss einer Automatisierungsausführung können Sie die Ausführung mit denselben oder geänderten Parametern erneut ausführen. Weitere Informationen finden Sie unter Wiederholtes Ausführen von Automatisierungsausführungen.

Ausführen einer einfachen Automatisierung (Befehlszeile)

Das folgende Verfahren beschreibt, wie Sie das AWS CLI (unter Linux oder Windows) verwenden oder AWS -Tools für PowerShell eine einfache Automatisierung ausführen.

Ausführen einer einfachen Automatisierung

1. Installieren und konfigurieren Sie das AWS CLI oder das AWS -Tools für PowerShell, falls Sie das noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.

2. Führen Sie den folgenden Befehl aus, um eine einfache Automatisierung zu starten. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   Linux & macOS

   ```
   aws ssm start-automation-execution \
       --document-name runbook name \
       --parameters runbook parameters
   ```

   Windows

   ```
   aws ssm start-automation-execution ^
       --document-name runbook name ^
       --parameters runbook parameters
   ```

   PowerShell

   ```
   Start-SSMAutomationExecution `
     -DocumentName runbook name `
     -Parameter runbook parameters
   ```

   Hier ist ein Beispiel, bei dem das Runbook verwendet wird`AWS-RestartEC2Instance`, um die angegebene EC2 Instanz neu zu starten.

Linux & macOS

```
aws ssm start-automation-execution \
    --document-name "AWS-RestartEC2Instance" \
    --parameters "InstanceId=i-02573cafcfEXAMPLE"
```

Windows

```
aws ssm start-automation-execution ^
    --document-name "AWS-RestartEC2Instance" ^
    --parameters "InstanceId=i-02573cafcfEXAMPLE"
```

PowerShell

```
Start-SSMAutomationExecution `
  -DocumentName AWS-RestartEC2Instance `
  -Parameter @{"InstanceId"="i-02573cafcfEXAMPLE"}
```

Das System gibt unter anderem folgende Informationen zurück

Linux & macOS

```
{
    "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0123456789ab"
}
```

Windows

```
{
    "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0123456789ab"
}
```

PowerShell

```
4105a4fc-f944-11e6-9d32-0123456789ab
```

3. Führen Sie den folgenden Befehl aus, um den Status der Automatisierung abzurufen.

Linux & macOS

```
aws ssm describe-automation-executions \
    --filter "Key=ExecutionId,Values=4105a4fc-f944-11e6-9d32-0123456789ab"
```

Windows

```
aws ssm describe-automation-executions ^
    --filter "Key=ExecutionId,Values=4105a4fc-f944-11e6-9d32-0123456789ab"
```

PowerShell

```
Get-SSMAutomationExecutionList | `
  Where {$_.AutomationExecutionId -eq "4105a4fc-f944-11e6-9d32-0123456789ab"}
```

Das System gibt unter anderem folgende Informationen zurück

Linux & macOS

```
{
    "AutomationExecutionMetadataList": [
        {
            "AutomationExecutionStatus": "InProgress",
            "CurrentStepName": "stopInstances",
            "Outputs": {},
            "DocumentName": "AWS-RestartEC2Instance",
            "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0123456789ab",
            "DocumentVersion": "1",
            "ResolvedTargets": {
                "ParameterValues": [],
                "Truncated": false
            },
            "AutomationType": "Local",
            "Mode": "Auto",
            "ExecutionStartTime": 1564600648.159,
            "CurrentAction": "aws:changeInstanceState",
            "ExecutedBy": "arn:aws:sts::123456789012:assumed-role/Administrator/
Admin",
            "LogFile": "",
            "Targets": []
```

```
        }
    ]
}
```

Windows

```
{
    "AutomationExecutionMetadataList": [
        {
            "AutomationExecutionStatus": "InProgress",
            "CurrentStepName": "stopInstances",
            "Outputs": {},
            "DocumentName": "AWS-RestartEC2Instance",
            "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0123456789ab",
            "DocumentVersion": "1",
            "ResolvedTargets": {
                "ParameterValues": [],
                "Truncated": false
            },
            "AutomationType": "Local",
            "Mode": "Auto",
            "ExecutionStartTime": 1564600648.159,
            "CurrentAction": "aws:changeInstanceState",
            "ExecutedBy": "arn:aws:sts::123456789012:assumed-role/Administrator/
Admin",
            "LogFile": "",
            "Targets": []
        }
    ]
}
```

PowerShell

```
AutomationExecutionId       : 4105a4fc-f944-11e6-9d32-0123456789ab
AutomationExecutionStatus   : InProgress
AutomationType              : Local
CurrentAction               : aws:changeInstanceState
CurrentStepName             : startInstances
DocumentName                : AWS-RestartEC2Instance
DocumentVersion             : 1
ExecutedBy                  : arn:aws:sts::123456789012:assumed-role/
Administrator/Admin
ExecutionEndTime            : 1/1/0001 12:00:00 AM
```

```
ExecutionStartTime          : 7/31/2019 7:17:28 PM
FailureMessage              :
LogFile                     :
MaxConcurrency              :
MaxErrors                   :
Mode                        : Auto
Outputs                     : {}
ParentAutomationExecutionId :
ResolvedTargets             :
 Amazon.SimpleSystemsManagement.Model.ResolvedTargets
Target                      :
TargetMaps                  : {}
TargetParameterName         :
Targets                     : {}
```

Wiederholtes Ausführen von Automatisierungsausführungen

Sie können AWS Systems Manager Automatisierungsausführungen erneut ausführen, um Aufgaben mit identischen oder geänderten Parametern zu wiederholen. Die Wiederholungsfunktion ermöglicht Ihnen, Automatisierungsausführungen effizient zu replizieren, ohne die Automatisierungskonfigurationen manuell neu erstellen zu müssen. Das reduziert den betrieblichen Aufwand und potenzielle Konfigurationsfehler.

Wenn Sie eine Automatisierungsausführung erneut ausführen, behält Systems Manager die ursprünglichen Runbook-Parameter, CloudWatch Amazon-Alarme und -Tags aus der vorherigen Ausführung bei. Das System erstellt eine neue Ausführung mit einer neuen Ausführungs-ID und aktualisierten Zeitstempeln. Sie können jede Art von Automatisierungsausführung erneut ausführen, einschließlich einfacher Ausführungen, Ratensteuerungsausführungen, konto- und regionsübergreifender Ausführungen sowie manueller Ausführungen.

Erneutes Ausführen einer Automatisierungsausführung (Konsole)

In den folgenden Verfahren wird beschrieben, wie Sie mithilfe der Systems-Manager-Konsole eine Automatisierungsausführung erneut ausführen.

Führen Sie eine Automatisierungsausführung auf der Automation-Startseite wie folgt erneut aus:

Öffnen Sie die Konsole unter AWS Systems Manager . https://console.aws.amazon.com/systems-manager/

1. Klicken Sie im Navigationsbereich auf Automation.

2. Wählen Sie in der Ausführungsliste die Ausführung aus, die Sie erneut ausführen möchten.
3. Wählen Sie Ausführung erneut ausführen aus.
4. Überprüfen Sie auf der Seite Automatisierungsdokument ausführen die vorab ausgefüllten Parameter, den Ausführungsmodus und die Zielkonfiguration der ursprünglichen Ausführung.
5. (Optional) Ändern Sie nach Bedarf Parameter, Ziele oder andere Einstellungen für die erneute Ausführung.
6. Wählen Sie Ausführen aus, um die erneute Ausführung mit einer neuen Ausführungs-ID zu starten.

Führen Sie eine Automatisierungsausführung auf der Seite mit den Ausführungsdetails wie folgt erneut aus:

Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

1. Klicken Sie im Navigationsbereich auf Automation.
2. Wählen Sie die Ausführungs-ID der Automatisierung aus, die Sie erneut ausführen möchten.
3. Wählen Sie auf der Seite mit den Ausführungsdetails die Option Ausführung erneut ausführen aus.
4. Überprüfen Sie auf der Seite Automatisierungsdokument ausführen die vorab ausgefüllten Parameter, den Ausführungsmodus und die Zielkonfiguration der ursprünglichen Ausführung.
5. (Optional) Ändern Sie nach Bedarf Parameter, Ziele oder andere Einstellungen für die erneute Ausführung.
6. Wählen Sie Ausführen aus, um die erneute Ausführung mit einer neuen Ausführungs-ID zu starten.

Kopieren Sie eine Automatisierungsausführung in eine neue Ausführung wie folgt:

Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

1. Klicken Sie im Navigationsbereich auf Automation.
2. Wählen Sie die Ausführungs-ID der Automatisierung aus, die Sie kopieren möchten.
3. Wählen Sie auf der Seite mit den Ausführungsdetails Aktionen und dann In neue Ausführung kopieren aus.

4. Überprüfen Sie auf der Seite Automatisierungsdokument ausführen die vorab ausgefüllten Parameter, den Ausführungsmodus und die Zielkonfiguration der ursprünglichen Ausführung.
5. (Optional) Ändern Sie nach Bedarf Parameter, Ziele oder andere Einstellungen für die neue Ausführung.
6. Wählen Sie Ausführen aus, um die neue Ausführung zu starten.

Eine Automatisierung ausführen, für die Genehmigungen erforderlich sind

Die folgenden Verfahren beschreiben, wie Sie die AWS Systems Manager Konsole verwenden und AWS Command Line Interface (AWS CLI) eine Automatisierung mit Genehmigungen mithilfe einer einfachen Ausführung ausführen. Die Automatisierung verwendet die Automatisierungsaktion `aws:approve`, die die Automatisierung vorübergehend unterbricht, bis die Aktion von den designierten Prinzipalen entweder genehmigt oder abgelehnt wird. Die Automatisierung wird im Kontext des aktuellen Benutzers ausgeführt. Das bedeutet, dass Sie keine zusätzlichen IAM-Berechtigungen konfigurieren müssen, solange Sie über die Berechtigung zum Ausführen des Runbooks verfügen und alle Aktionen von dem Runbook aufgerufen werden. Wenn Sie über Administrator-Berechtigungen in IAM verfügen, haben Sie bereits die Berechtigung zur Verwendung dieses Runbooks.

Bevor Sie beginnen

Zusätzlich zu den Standardeingaben, die für das Runbook erforderlich sind, erfordert die Aktion `aws:approve` die beiden folgenden Parameter:

- Eine Liste der Genehmiger. Die Liste der Genehmiger muss mindestens einen Genehmiger in Form eines Benutzernamens oder eines Benutzer-ARN enthalten. Wenn mehrere Genehmiger angegeben sind, muss im Runbook eine entsprechende minimale Genehmigungsanzahl festgelegt werden.
- Ein Amazon Simple Notification Service (Amazon SNS)-Thema ARN Der Name des Amazon SNS-Themas muss mit `Automation` beginnen.

Bei diesem Verfahren wird davon ausgegangen, dass Sie bereits ein Amazon SNS-Thema erstellt haben. Dies ist erforderlich, um den Genehmigungsprozess bereitzustellen. Weitere Informationen finden Sie unter Erstellen eines Themas im Amazon Simple Notification Service-Entwicklerhandbuch.

Ausführen einer Automatisierung mit Genehmigern (Konsole)

So führen Sie eine Automatisierung mit Genehmigern aus

Im folgenden Verfahren wird beschrieben, wie Sie mithilfe der Systems Manager-Konsole eine Automatisierung mit Genehmigern ausführen.

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Automatisierung und Automatisierung ausführen aus.
3. Wählen Sie in der Liste Automation-Dokument ein Runbook. Wählen Sie eine oder mehrere Optionen im Bereich Dokumentkategorien, um SSM-Dokumente nach ihrem Zweck zu filtern. Um ein Runbook anzuzeigen, das Sie besitzen, wählen Sie die Im Besitz von mir-Registerkarte. Um ein Runbook anzuzeigen, das für Ihr Konto freigegeben ist, wählen Sie die Registerkarte Mit mir geteilt. Um alle Runbooks anzuzeigen, wählen Sie die Registerkarte Alle Dokumente.

   > **Note**
   >
   > Sie können Informationen zu einem Runbook einsehen, indem Sie den Runbook-Namen auswählen.

4. Überprüfen Sie im Abschnitt Dokument-Details, ob Dokumentversion auf die Version gesetzt ist, die Sie ausführen möchten. Das System bietet die folgenden Versionsoptionen:

   - Standardversion zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird und eine neue Standardversion zugewiesen ist.
   - Letzte Version zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird, und Sie die Version auszuführen möchten, die zuletzt aktualisiert wurde.
   - 1 (Standard) – Wählen Sie diese Option zur Ausführung der ersten Version des Dokuments, welches der Standard ist.

5. Wählen Sie Weiter aus.
6. Klicken Sie auf der Seite Execute automation document (Automation-Dokument ausführen) auf Simple execution (Einfache Ausführung).
7. Geben Sie im Abschnitt Eingabeparameter die erforderlichen Eingabeparameter an.

Wenn Sie beispielsweise das **`AWS-StartEC2InstanceWithApproval`** Runbook ausgewählt haben, müssen Sie IDs für den InstanceIdParameter eine Instanz angeben oder auswählen.

8. Geben Sie im Abschnitt Genehmiger die Benutzernamen oder Benutzer der ARNs Genehmiger für die Automatisierungsaktion an.
9. Geben Sie im Abschnitt SNSTopicARN das SNS-Thema ARN an, das für das Senden der Genehmigungsbenachrichtigung verwendet werden soll. Der SNS-Themenname muss mit Automation beginnen.
10. Optional können Sie eine IAM-Dienstrolle aus der AutomationAssumeRoleListe auswählen. Wenn Sie auf mehr als 100 Konten und Regionen abzielen, müssen Sie die `AWS-SystemsManager-AutomationAdministrationRole` angeben.
11. Wählen Sie Automatisierung ausführen.

Der angegebene Genehmiger erhält eine Amazon SNS-Benachrichtigung mit Details zum Genehmigen oder Ablehnen der Automatisierung. Diese Genehmigungsaktion ist 7 Tage ab dem Ausstellungsdatum gültig und kann über die Systems Manager Manager-Konsole oder die AWS Command Line Interface (AWS CLI) ausgeführt werden.

Wenn Sie die Automatisierung genehmigen, führt die Automatisierung die im angegebenen Runbook enthaltenen Schritte aus. Die Konsole zeigt den Status der Automatisierung an. Wenn Automatisierung nicht ausgeführt werden kann, finden Sie weitere Informationen unter Fehlerbehebung für Systems Manager Automation..

So genehmigen Sie eine Automatisierung oder lehnen sie ab

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation und wählen Sie dann die Automatisierung aus, die im vorherigen Verfahren ausgeführt wurde.
3. Wählen Sie Actions (Aktionen) und dann Approve/Deny (Genehmigen/ablehnen) aus.
4. Wählen Sie entweder Approve (Genehmigen) oder Deny (Ablehnen) aus und geben Sie bei Bedarf einen Kommentar ein.
5. Wählen Sie Absenden aus.

Ausführen einer Automatisierung mit Genehmigern (Befehlszeile)

Das folgende Verfahren beschreibt, wie Sie die AWS CLI (unter Linux oder Windows) verwenden oder AWS -Tools für PowerShell eine Automatisierung mit Genehmigern ausführen.

So führen Sie eine Automatisierung mit Genehmigern aus

1. Installieren und konfigurieren Sie das AWS CLI oder das AWS -Tools für PowerShell, falls Sie das noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.

2. Verwenden Sie den folgenden Befehl, um eine Automatisierung mit Genehmigern auszuführen. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen. Geben Sie im Abschnitt Dokumentname ein Runbook an, das die Automatisierungsaktion `aws:approve` enthält.

   Geben Sie für `Approvers` die Aktion die Benutzernamen oder den Benutzer ARNs der Genehmiger an. Geben Sie für `SNSTopic` den SNS-Themen-ARN an, der zum Senden von Genehmigungsbenachrichtigungen verwendet werden soll. Der Name des Amazon SNS-Themas muss mit `Automation` beginnen.

   > **Note**
   >
   > Die spezifischen Namen der Parameterwerte für Genehmiger und das SNS-Thema hängen von den im ausgewählten Runbook angegebenen Werten ab.

   Linux & macOS

   ```
   aws ssm start-automation-execution \
       --document-name "AWS-StartEC2InstanceWithApproval" \
       --parameters
    "InstanceId=i-02573cafcfEXAMPLE,Approvers=arn:aws:iam::123456789012:role/
   Administrator,SNSTopicArn=arn:aws:sns:region:123456789012:AutomationApproval"
   ```

   Windows

   ```
   aws ssm start-automation-execution ^
   ```

```
    --document-name "AWS-StartEC2InstanceWithApproval" ^
    --parameters
 "InstanceId=i-02573cafcfEXAMPLE,Approvers=arn:aws:iam::123456789012:role/
Administrator,SNSTopicArn=arn:aws:sns:region:123456789012:AutomationApproval"
```

PowerShell

```
Start-SSMAutomationExecution `
    -DocumentName AWS-StartEC2InstanceWithApproval `
    -Parameters @{
        "InstanceId"="i-02573cafcfEXAMPLE"
        "Approvers"="arn:aws:iam::123456789012:role/Administrator"
        "SNSTopicArn"="arn:aws:sns:region:123456789012:AutomationApproval"
    }
```

Das System gibt unter anderem folgende Informationen zurück

Linux & macOS

```
{
    "AutomationExecutionId": "df325c6d-b1b1-4aa0-8003-6cb7338213c6"
}
```

Windows

```
{
    "AutomationExecutionId": "df325c6d-b1b1-4aa0-8003-6cb7338213c6"
}
```

PowerShell

```
df325c6d-b1b1-4aa0-8003-6cb7338213c6
```

So genehmigen Sie eine Automatisierung

- Führen Sie den folgenden Befehl aus, um eine Automatisierung zu genehmigen. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

Linux & macOS

```
aws ssm send-automation-signal \
    --automation-execution-id "df325c6d-b1b1-4aa0-8003-6cb7338213c6" \
    --signal-type "Approve" \
    --payload "Comment=your comments"
```

Windows

```
aws ssm send-automation-signal ^
    --automation-execution-id "df325c6d-b1b1-4aa0-8003-6cb7338213c6" ^
    --signal-type "Approve" ^
    --payload "Comment=your comments"
```

PowerShell

```
Send-SSMAutomationSignal `
    -AutomationExecutionId df325c6d-b1b1-4aa0-8003-6cb7338213c6 `
    -SignalType Approve `
    -Payload @{"Comment"="your comments"}
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

So lehnen Sie eine Automatisierung ab

- Führen Sie den folgenden Befehl aus, um eine Automatisierung abzulehnen. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

  Linux & macOS

  ```
  aws ssm send-automation-signal \
      --automation-execution-id "df325c6d-b1b1-4aa0-8003-6cb7338213c6" \
      --signal-type "Deny" \
      --payload "Comment=your comments"
  ```

  Windows

  ```
  aws ssm send-automation-signal ^
  ```

```
    --automation-execution-id "df325c6d-b1b1-4aa0-8003-6cb7338213c6" ^
    --signal-type "Deny" ^
    --payload "Comment=your comments"
```

PowerShell

```
Send-SSMAutomationSignal `
    -AutomationExecutionId df325c6d-b1b1-4aa0-8003-6cb7338213c6 `
    -SignalType Deny `
    -Payload @{"Comment"="your comments"}
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

Automatisierte Abläufe in großem Umfang ausführen

Mit AWS Systems Manager Automation können Sie mithilfe von Zielen Automatisierungen auf einer Flotte von AWS Ressourcen ausführen. Außerdem können Sie die Bereitstellung der Automatisierung innerhalb Ihrer Flotte steuern, indem Sie einen Gleichzeitigkeitswert und einen Fehlergrenzwert angeben. Die Gleichzeitigkeits- und die Fehlergrenzwertfeature werden gemeinsam als Ratensteuerungen bezeichnet. Der Gleichzeitigkeitswert legt fest, wie viele Ressourcen die Automatisierung gleichzeitig ausführen kann. Automation bietet außerdem einen adaptiven Nebenläufigkeitsmodus, den Sie aktivieren können. Die adaptive Nebenläufigkeit skaliert Ihr Automatisierungskontingent automatisch von 100 gleichzeitig ausgeführten Automatisierungen auf bis zu 500. Ein Fehlergrenzwert legt fest, wie viele Automatisierungsausführungen fehlschlagen dürfen, bevor Systems Manager damit aufhört, die Automatisierung an andere Ressourcen zu senden.

Weitere Informationen über Gleichzeitigkeits- und Fehlergrenzwerte finden Sie unter Steuern von Automatisierungen im großen Maßstab. Weitere Informationen über Ziele finden Sie unter Zuordnen von Zielen für eine Automatisierung.

Die folgenden Verfahren veranschaulichen, wie Sie die adaptive Nebenläufigkeit aktivieren und eine Automatisierung mit Zielen und Ratensteuerelementen über die Systems-Manager-Konsole und AWS Command Line Interface (AWS CLI) ausführen.

Ausführen einer Automatisierung mit Zielen und Ratensteuerungen (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie mit der Systems Manager-Konsole eine Automatisierung mit Ziel- und Ratensteuerungen ausführen.

So führen Sie eine Automatisierung mit Zielen und Ratensteuerungen aus

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Automatisierung und Automatisierung ausführen aus.
3. Wählen Sie in der Liste Automation-Dokument ein Runbook. Wählen Sie eine oder mehrere Optionen im Bereich Dokumentkategorien, um SSM-Dokumente nach ihrem Zweck zu filtern. Um ein Runbook anzuzeigen, das Sie besitzen, wählen Sie die Im Besitz von mir-Registerkarte. Um ein Runbook anzuzeigen, das für Ihr Konto freigegeben ist, wählen Sie die Registerkarte Mit mir geteilt. Um alle Runbooks anzuzeigen, wählen Sie die Registerkarte Alle Dokumente.

   > **Note**
   >
   > Sie können Informationen zu einem Runbook einsehen, indem Sie den Runbook-Namen auswählen.

4. Überprüfen Sie im Abschnitt Dokument-Details, ob Dokumentversion auf die Version gesetzt ist, die Sie ausführen möchten. Das System bietet die folgenden Versionsoptionen:

   - Standardversion zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird und eine neue Standardversion zugewiesen ist.
   - Letzte Version zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird, und Sie die Version auszuführen möchten, die zuletzt aktualisiert wurde.
   - 1 (Standard) – Wählen Sie diese Option zur Ausführung der ersten Version des Dokuments, welches der Standard ist.

5. Wählen Sie Weiter aus.
6. Wählen Sie im Abschnitt Execution Mode (Ausführungsmodus) die Option Rate Control (Ratensteuerung) aus. Sie müssen diesen Modus oder die Option Multi-account and Region (Mehrere Konten und Regionen) verwenden, wenn Sie Ziele und Ratensteuerungen nutzen möchten.
7. Wählen Sie im Abschnitt Targets (Ziele) die Ausrichtung auf die AWS -Ressourcen für die Ausführung der Automation. Diese Optionen sind erforderlich.

   a. Wählen Sie in der Liste Parameter einen Parameter aus. Die Elemente in der Liste Parameter richten sich nach den Parametern in dem Automation-Runbook, das Sie zu

Beginn dieses Verfahrens ausgewählt haben. Durch Auswahl eines Parameters legen Sie den Typ der Ressource fest, für die der Automation-Workflow ausgeführt wird.

b. Wählen Sie in der Liste Ziele aus, wie Sie Ressourcen als Ziele verwenden möchten.

   i. Wenn Sie die Zielressourcen mithilfe von Parameterwerten ausgewählt haben, geben Sie den Parameterwert für den gewählten Parameter im Feld Eingabeparameter eingeben ein.

   ii. Wenn Sie Ressourcen mithilfe von als Ziel verwenden möchten AWS - Ressourcengruppen, wählen Sie den Namen der Gruppe aus der Liste der Ressourcengruppen aus.

   iii. Wenn Sie die Zielressourcen mithilfe von Tags ausgewählt haben, geben Sie den Tag-Schlüssel und (optional) den Tag-Wert in die entsprechenden Felder ein. Wählen Sie Hinzufügen aus.

   iv. Wenn Sie ein Automatisierungs-Runbook auf allen Instanzen im aktuellen AWS-Konto und ausführen möchten AWS-Region, wählen Sie Alle Instanzen aus.

8. Geben Sie im Abschnitt Eingabeparameter die erforderlichen Eingaben an. Optional können Sie eine IAM-Servicerolle aus der AutomationAssumeRoleListe auswählen.

> **Note**
>
> Möglicherweise müssen Sie einige der Optionen im Abschnitt Input parameters (Eingabeparameter) nicht auswählen. Dies liegt daran, dass Sie Ressourcen mithilfe von Tags oder einer Ressourcengruppe als Ziele ausgewählt haben. Wenn Sie beispielsweise das `AWS-RestartEC2Instance` Runbook ausgewählt haben, müssen Sie IDs im Abschnitt Eingabeparameter keine Instanz angeben oder auswählen. Die Automation-Ausführung sucht die Instances für den Neustart mit den von Ihnen angegebenen Tags oder Ressourcengruppen.

9. Verwenden Sie die Optionen im Abschnitt Tarifsteuerung, um die Anzahl der AWS Ressourcen zu beschränken, die die Automatisierung innerhalb der einzelnen Konto-Region-Paare ausführen können.

   Wählen Sie im Abschnitt Gleichzeitigkeit eine Option aus:

   - Wählen Sie Ziele aus, um eine absolute Anzahl von Zielen einzugeben, die den Automation-Workflow gleichzeitig ausführen können.

- Wählen Sie Prozentsatz aus, um einen Prozentsatz der Ziele anzugeben, die den Automation-Workflow gleichzeitig ausführen können.

10. Wählen Sie im Abschnitt Fehlerschwellenwert eine Option aus:

    - Wählen Sie Fehler, um eine absolute Anzahl von zulässigen Fehlern anzugeben, bevor Automation damit aufhört, den Workflow an andere Ressourcen zu senden.
    - Wählen Sie Prozentsatz aus, um einen Prozentsatz von zulässigen Fehlern anzugeben, bevor Automation damit aufhört, den Workflow an andere Ressourcen zu senden.

11. (Optional) Wählen Sie einen CloudWatch Alarm aus, der zur Überwachung auf Ihre Automatisierung angewendet werden soll. Um Ihrer Automatisierung einen CloudWatch Alarm zuzuweisen, muss der IAM-Principal, der die Automatisierung startet, über die entsprechende Genehmigung verfügen`iam:createServiceLinkedRole`. Weitere Informationen zu CloudWatch Alarmen finden Sie unter CloudWatch Amazon-Alarme verwenden. Beachten Sie, dass die Automatisierung gestoppt wird, wenn Ihr Alarm aktiviert wird. Wenn Sie dies verwenden AWS CloudTrail, wird der API-Aufruf in Ihrem Trail angezeigt.
12. Wählen Sie Ausführen.

Um Automatisierungen anzuzeigen, die von der Automatisierung der Ratensteuerung gestartet wurden, wählen Sie im Navigationsbereich Automation (Automatisierung) und wählen Sie dann Anzeigen von untergeordneten Automatisierungen.

Nach Abschluss einer Automatisierungsausführung können Sie die Ausführung mit denselben oder geänderten Parametern erneut ausführen. Weitere Informationen finden Sie unter Wiederholtes Ausführen von Automatisierungsausführungen.

Ausführen einer Automatisierung mit Zielen und Ratensteuerungen (Befehlszeile)

Das folgende Verfahren beschreibt, wie Sie die AWS CLI (unter Linux oder Windows) verwenden oder AWS -Tools für PowerShell eine Automatisierung mit Zielen und Ratensteuerungen ausführen.

So führen Sie eine Automatisierung mit Zielen und Ratensteuerungen aus

1. Installieren und konfigurieren Sie das AWS CLI oder das AWS -Tools für PowerShell, falls Sie das noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.
2. Nutzen Sie den folgenden Befehl, um eine Liste der Dokumente anzuzeigen.

Linux & macOS

```
aws ssm list-documents
```

Windows

```
aws ssm list-documents
```

PowerShell

```
Get-SSMDocumentList
```

Beachten Sie den Namen des Runbooks, das Sie verwenden möchten.

3. Führen Sie den folgenden Befehl aus, um Details des Runbooks einsehen zu können: Ersetzen Sie das *runbook name* durch den Namen des Runbooks, dessen Details Sie anzeigen möchten. Notieren Sie auch einen Parameternamen (z. B. `InstanceId`), den Sie für die Option `--target-parameter-name` verwenden möchten. Dieser Parameter bestimmt den Typ der Ressource, für die die Automatisierung ausgeführt wird.

   Linux & macOS

   ```
   aws ssm describe-document \
       --name runbook name
   ```

   Windows

   ```
   aws ssm describe-document ^
       --name runbook name
   ```

   PowerShell

   ```
   Get-SSMDocumentDescription `
       -Name runbook name
   ```

4. Erstellen Sie einen Befehl, der die Ziel- und Ratensteuerungsoptionen verwendet, die Sie ausführen möchten. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

Ausrichtung mithilfe von Tags

Linux & macOS

```
aws ssm start-automation-execution \
    --document-name runbook name \
    --targets Key=tag:key name,Values=value \
    --target-parameter-name parameter name \
    --parameters "input parameter name=input parameter value,input parameter 2
 name=input parameter 2 value" \
    --max-concurrency 10 \
    --max-errors 25%
```

Windows

```
aws ssm start-automation-execution ^
    --document-name runbook name ^
    --targets Key=tag:key name,Values=value ^
    --target-parameter-name parameter name ^
    --parameters "input parameter name=input parameter value,input parameter 2
 name=input parameter 2 value" ^
    --max-concurrency 10 ^
    --max-errors 25%
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
$Targets.Key = "tag:key name"
$Targets.Values = "value"

Start-SSMAutomationExecution `
    DocumentName "runbook name" `
    -Targets $Targets `
    -TargetParameterName "parameter name" `
    -Parameter @{"input parameter name"="input parameter value";"input parameter
 2 name"="input parameter 2 value"} `
    -MaxConcurrency "10" `
    -MaxError "25%"
```

Ausrichtung mithilfe von Parameterwerten

Linux & macOS

```
aws ssm start-automation-execution \
    --document-name runbook name \
    --targets Key=ParameterValues,Values=value,value 2,value 3 \
    --target-parameter-name parameter name \
    --parameters "input parameter name=input parameter value" \
    --max-concurrency 10 \
    --max-errors 25%
```

Windows

```
aws ssm start-automation-execution ^
    --document-name runbook name ^
    --targets Key=ParameterValues,Values=value,value 2,value 3 ^
    --target-parameter-name parameter name ^
    --parameters "input parameter name=input parameter value" ^
    --max-concurrency 10 ^
    --max-errors 25%
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
$Targets.Key = "ParameterValues"
$Targets.Values = "value","value 2","value 3"

Start-SSMAutomationExecution `
    -DocumentName "runbook name" `
    -Targets $Targets `
    -TargetParameterName "parameter name" `
    -Parameter @{"input parameter name"="input parameter value"} `
    -MaxConcurrency "10" `
    -MaxError "25%"
```

Targeting mit AWS -Ressourcengruppen

Linux & macOS

```
aws ssm start-automation-execution \
    --document-name runbook name \
```

```
    --targets Key=ResourceGroup,Values=Resource group nname \
    --target-parameter-name parameter name \
    --parameters "input parameter name=input parameter value" \
    --max-concurrency 10 \
    --max-errors 25%
```

Windows

```
aws ssm start-automation-execution ^
    --document-name runbook name ^
    --targets Key=ResourceGroup,Values=Resource group name ^
    --target-parameter-name parameter name ^
    --parameters "input parameter name=input parameter value" ^
    --max-concurrency 10 ^
    --max-errors 25%
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
$Targets.Key = "ResourceGroup"
$Targets.Values = "Resource group name"

Start-SSMAutomationExecution `
    -DocumentName "runbook name" `
    -Targets $Targets `
    -TargetParameterName "parameter name" `
    -Parameter @{"input parameter name"="input parameter value"} `
    -MaxConcurrency "10" `
    -MaxError "25%"
```

Ausrichtung auf alle EC2 Amazon-Instances in der aktuellen AWS-Konto und AWS-Region

Linux & macOS

```
aws ssm start-automation-execution \
    --document-name runbook name \
    --targets "Key=AWS::EC2::Instance,Values=*"  \
    --target-parameter-name instanceId \
    --parameters "input parameter name=input parameter value" \
    --max-concurrency 10 \
```

```
    --max-errors 25%
```

Windows

```
aws ssm start-automation-execution ^
    --document-name runbook name ^
    --targets Key=AWS::EC2::Instance,Values=* ^
    --target-parameter-name instanceId ^
    --parameters "input parameter name=input parameter value" ^
    --max-concurrency 10 ^
    --max-errors 25%
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
$Targets.Key = "AWS::EC2::Instance"
$Targets.Values = "*"

Start-SSMAutomationExecution `
    -DocumentName "runbook name" `
    -Targets $Targets `
    -TargetParameterName "instanceId" `
    -Parameter @{"input parameter name"="input parameter value"} `
    -MaxConcurrency "10" `
    -MaxError "25%"
```

Der Befehl gibt eine Ausführungs-ID zurück. Kopieren Sie diese ID in die Zwischenablage. Sie können diese ID zum Anzeigen des Status der Automatisierung verwenden.

Linux & macOS

```
{
    "AutomationExecutionId": "a4a3c0e9-7efd-462a-8594-01234EXAMPLE"
}
```

Windows

```
{
    "AutomationExecutionId": "a4a3c0e9-7efd-462a-8594-01234EXAMPLE"
}
```

PowerShell

```
a4a3c0e9-7efd-462a-8594-01234EXAMPLE
```

5. Führen Sie den folgenden Befehl aus, um die Automatisierung anzuzeigen. Ersetzen Sie jeden *automation execution ID* durch Ihre Informationen.

   Linux & macOS

   ```
   aws ssm describe-automation-executions \
       --filter Key=ExecutionId,Values=automation execution ID
   ```

   Windows

   ```
   aws ssm describe-automation-executions ^
       --filter Key=ExecutionId,Values=automation execution ID
   ```

   PowerShell

   ```
   Get-SSMAutomationExecutionList | `
       Where {$_.AutomationExecutionId -eq "automation execution ID"}
   ```

6. Führen Sie den folgenden Befehl aus, um Details über den Automatisierungsprozess anzuzeigen. Ersetzen Sie jeden *automation execution ID* durch Ihre Informationen.

   Linux & macOS

   ```
   aws ssm get-automation-execution \
       --automation-execution-id automation execution ID
   ```

   Windows

   ```
   aws ssm get-automation-execution ^
       --automation-execution-id automation execution ID
   ```

   PowerShell

   ```
   Get-SSMAutomationExecution `
       -AutomationExecutionId automation execution ID
   ```

Das System gibt unter anderem folgende Informationen zurück

Linux & macOS


```
{
    "AutomationExecution": {
        "StepExecutionsTruncated": false,
        "AutomationExecutionStatus": "Success",
        "MaxConcurrency": "1",
        "Parameters": {},
        "MaxErrors": "1",
        "Outputs": {},
        "DocumentName": "AWS-StopEC2Instance",
        "AutomationExecutionId": "a4a3c0e9-7efd-462a-8594-01234EXAMPLE",
        "ResolvedTargets": {
            "ParameterValues": [
                "i-02573cafcfEXAMPLE"
            ],
            "Truncated": false
        },
        "ExecutionEndTime": 1564681619.915,
        "Targets": [
            {
                "Values": [
                    "DEV"
                ],
                "Key": "tag:ENV"
            }
        ],
        "DocumentVersion": "1",
        "ExecutionStartTime": 1564681576.09,
        "ExecutedBy": "arn:aws:sts::123456789012:assumed-role/Administrator/
Admin",
        "StepExecutions": [
            {
                "Inputs": {
                    "InstanceId": "i-02573cafcfEXAMPLE"
                },
                "Outputs": {},
                "StepName": "i-02573cafcfEXAMPLE",
                "ExecutionEndTime": 1564681619.093,
                "StepExecutionId": "86c7b811-3896-4b78-b897-01234EXAMPLE",
```

```
                "ExecutionStartTime": 1564681576.836,
                "Action": "aws:executeAutomation",
                "StepStatus": "Success"
            }
        ],
        "TargetParameterName": "InstanceId",
        "Mode": "Auto"
    }
}
```

Windows

```
{
    "AutomationExecution": {
        "StepExecutionsTruncated": false,
        "AutomationExecutionStatus": "Success",
        "MaxConcurrency": "1",
        "Parameters": {},
        "MaxErrors": "1",
        "Outputs": {},
        "DocumentName": "AWS-StopEC2Instance",
        "AutomationExecutionId": "a4a3c0e9-7efd-462a-8594-01234EXAMPLE",
        "ResolvedTargets": {
            "ParameterValues": [
                "i-02573cafcfEXAMPLE"
            ],
            "Truncated": false
        },
        "ExecutionEndTime": 1564681619.915,
        "Targets": [
            {
                "Values": [
                    "DEV"
                ],
                "Key": "tag:ENV"
            }
        ],
        "DocumentVersion": "1",
        "ExecutionStartTime": 1564681576.09,
        "ExecutedBy": "arn:aws:sts::123456789012:assumed-role/Administrator/
Admin",
        "StepExecutions": [
            {
```

```
                "Inputs": {
                    "InstanceId": "i-02573cafcfEXAMPLE"
                },
                "Outputs": {},
                "StepName": "i-02573cafcfEXAMPLE",
                "ExecutionEndTime": 1564681619.093,
                "StepExecutionId": "86c7b811-3896-4b78-b897-01234EXAMPLE",
                "ExecutionStartTime": 1564681576.836,
                "Action": "aws:executeAutomation",
                "StepStatus": "Success"
            }
        ],
        "TargetParameterName": "InstanceId",
        "Mode": "Auto"
    }
}
```

PowerShell

```
AutomationExecutionId       : a4a3c0e9-7efd-462a-8594-01234EXAMPLE
AutomationExecutionStatus   : Success
CurrentAction               :
CurrentStepName             :
DocumentName                : AWS-StopEC2Instance
DocumentVersion             : 1
ExecutedBy                  : arn:aws:sts::123456789012:assumed-role/
Administrator/Admin
ExecutionEndTime            : 8/1/2019 5:46:59 PM
ExecutionStartTime          : 8/1/2019 5:46:16 PM
FailureMessage              :
MaxConcurrency              : 1
MaxErrors                   : 1
Mode                        : Auto
Outputs                     : {}
Parameters                  : {}
ParentAutomationExecutionId :
ProgressCounters            :
ResolvedTargets             :
 Amazon.SimpleSystemsManagement.Model.ResolvedTargets
StepExecutions              : {i-02573cafcfEXAMPLE}
StepExecutionsTruncated     : False
Target                      :
TargetLocations             : {}
```

```
TargetMaps                  : {}
TargetParameterName         : InstanceId
Targets                     : {tag:Name}
```

> **Note**
>
> Sie können auch den Status der Automatisierung in der Konsole überwachen. Wählen Sie in der Liste Automatisierungs-Ausführung die Automatisierung, die Sie gerade ausgeführt haben, und wählen Sie dann die Registerkarte Ausführungsschritte. Diese Registerkarte zeigt Ihnen den Status der Automatisierungs-Aktionen.

Zuordnen von Zielen für eine Automatisierung

Verwenden Sie den `Targets`-Parameter, um schnell zu definieren, auf welche Ressourcen eine Automatisierung abzielt. Wenn Sie beispielsweise eine Automatisierung ausführen möchten, die Ihre verwalteten Instances neu startet, können Sie, anstatt manuell Dutzende von Instance-IDs in der Konsole oder in einem Befehl einzugeben, Ziel-Instances festlegen, indem Sie Amazon Elastic Compute Cloud (Amazon EC2)-Tags mit dem `Targets`-Parameter verwenden.

Wenn Sie eine Automatisierung ausführen, die ein Ziel verwendet, erstellt AWS Systems Manager eine untergeordnete Automatisierung für jedes Ziel. Wenn Sie z. B. mithilfe von Tags Amazon Elastic Block Store (Amazon EBS)-Volume angeben und diese Tags auf 100 Amazon EBS-Volumes aufgelöst werden, dann erstellt Systems Manager 100 untergeordnete Automatisierungen. Die übergeordnete Automatisierung ist abgeschlossen, wenn alle untergeordneten Automatisierungen einen endgültigen Status erreicht haben.

> **Note**
>
> Alle `input parameters`, die Sie zur Laufzeit angeben (entweder im Abschnitt Input parameters (Eingabeparameter) der Konsole oder mithilfe der Option `parameters` auf der Befehlszeile) werden automatisch von allen untergeordneten Automatisierungen verarbeitet.

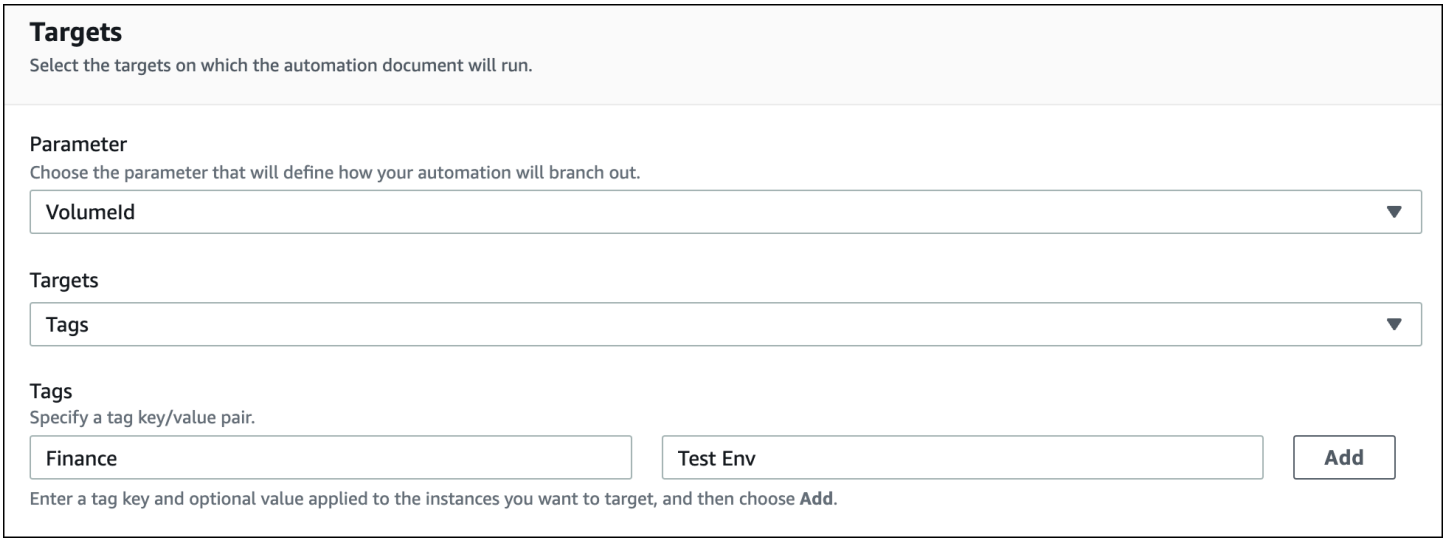
Sie können Ressourcen für eine Automatisierung gezielt einsetzen, indem Sie Tags, Resource Groups und Parameterwerte verwenden. Darüber hinaus können Sie mit der Option `TargetMaps` mehrere Parameterwerte über die Befehlszeile oder eine Datei als Ziel einrichten. Der folgende Abschnitt beschreibt die einzelnen Targeting-Optionen eingehender.

Anzielen eines Tags

Sie können einen einzelnen Tag als Ziel einer Automatisierung bestimmen. Viele AWS-Ressourcen unterstützen Tags, einschließlich Amazon Elastic Compute Cloud (Amazon EC2) und Amazon Relational Database Service (Amazon RDS) -Instances, Amazon Elastic Block Store (Amazon EBS) -Volumes und -Snapshots, Resource Groups und Amazon Simple Storage Service (Amazon S3) -Buckets. Sie können Automatisierungen schnell auf Ihren AWS-Ressourcen ausführen, indem Sie einen Tag anzielen. Ein Tag ist ein Schlüssel-Wert-Paar, z. B. Operating_System:Linux oder Department:Finance. Wenn Sie einer Ressource einen bestimmten Namen zuweisen, können Sie auch das Wort „Name“ als Schlüssel und den Namen der Ressource als Wert verwenden.

Wenn Sie einen Tag als Ziel für eine Automatisierung angeben, geben Sie auch einen Ziel-Parameter an. Der Ziel-Parameter verwendet die Option `TargetParameterName`. Durch Auswahl eines Zielparameters legen Sie den Typ der Ressource fest, für die die Automatisierung ausgeführt wird. Der Zielparameter, den Sie mit dem Tag angeben, muss ein im Runbook definierter gültiger Parameter sein. Wenn Sie beispielsweise Tags für Dutzende von EC2-Instances verwenden möchten, wählen Sie den Zielparameter `InstanceId`. Durch die Auswahl dieses Parameters legen Sie Instances als Ressourcentyp für die Automatisierung fest. Beim Erstellen eines benutzerdefinierten Runbooks müssen Sie den Zieltyp als `/AWS::EC2::Instance` angeben, um sicherzustellen, dass nur Instances verwendet werden. Andernfalls werden alle Ressourcen mit demselben Tag als Ziel ausgewählt. Wenn Sie auf Instances mit einem Tag abzielen, werden möglicherweise beendete Instances eingeschlossen.

Im folgenden Screenshot werden die `AWS-DetachEBSVolume`-Runbook verwendet. Der logische Ziel-Parameter ist `VolumeId`.



Das `AWS-DetachEBSVolume`-Runbook enthält auch eine spezielle Eigenschaft namens Zieltyp, welche auf `/AWS::EC2::Volume` gesetzt wird. Dies bedeutet: Wenn das Tag-Schlüssel-Paar `Finance:TestEnv` unterschiedliche Ressourcentypen zurückgibt (zum Beispiel EC2-Instances, Amazon EBS-Volumes, Amazon EBS-Snapshots), werden nur Amazon EBS-Volumes verwendet.

> **Important**
>
> Bei Zielparameternamen muss die Groß- und Kleinschreibung beachtet werden. Wenn Sie Automatisierungen mit AWS Command Line Interface (AWS CLI) oder AWS Tools for Windows PowerShell ausführen, müssen Sie den Namen des Zielparameters genau wie im Runbook definiert angeben. Andernfalls gibt das System einen `InvalidAutomationExecutionParametersException`-Fehler aus. Sie können die API-Operation DescribeDocument verwenden, um Informationen zu den in einem bestimmten Runbook verfügbaren Zielparametern anzuzeigen. Im Folgenden sehen Sie einen Beispiel AWS CLI-Befehl, der Informationen über das `AWS-DeleteSnapshot`-Dokument gibt.
>
> ```
> aws ssm describe-document \
>     --name AWS-DeleteSnapshot
> ```

Hier sehen Sie einige AWS CLI-Beispielbefehle, die Ziel-Ressourcen mithilfe von Tags angeben.

Beispiel 1: Zielgerichtete Tags mit einem Schlüssel-Wert-Paar zum Neustarten von Amazon-EC2-Instances

Dieses Beispiel startet alle Amazon-EC2-Instances neu, die mit dem Schlüssel Department und dem Wert HumanResources markiert sind. Der Zielparameter verwendet den Parameter InstanceId aus dem Runbook. Im Beispiel wird ein zusätzlicher Parameter für die Ausführung der Automation mithilfe einer Automation-Servicerolle (auch als Übernahmerolle bezeichnet) verwendet.

```
aws ssm start-automation-execution \
    --document-name AWS-RestartEC2Instance \
    --targets Key=tag:Department,Values=HumanResources \
    --target-parameter-name InstanceId \
    --parameters "AutomationAssumeRole=arn:aws:iam::111122223333:role/
AutomationServiceRole"
```

Beispiel 2: Zielgerichtete Tags mit einem Schlüssel-Wert-Paar zum Löschen von Amazon-EBS-Snapshots

Das folgende Beispiel verwendet das `AWS-DeleteSnapshot`-Runbook zum Löschen aller Snapshots mit dem Schlüssel Name und dem Wert January2018Backups. Der Ziel-Parameter verwendet den Parameter VolumeId.

```
aws ssm start-automation-execution \
    --document-name AWS-DeleteSnapshot \
    --targets Key=tag:Name,Values=January2018Backups \
    --target-parameter-name VolumeId
```

Targeting von AWS -Ressourcengruppen

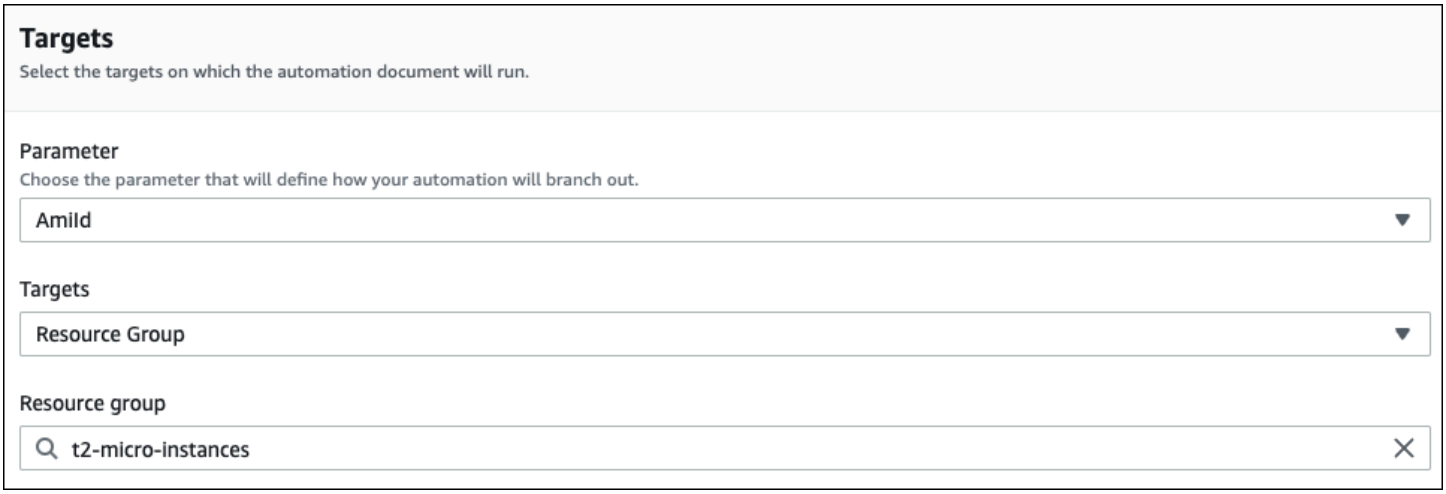
Sie können eine einzelne AWS-Ressourcengruppe als Ziel einer Automatisierung bestimmen. Systems Manager erstellt eine untergeordnete Automatisierung für jedes Objekt in der Ziel-Ressourcengruppe.

Beispiel: Angenommen, eine Ihrer Ressourcengruppen ist PatchedAMIs. Diese Ressourcengruppe enthält eine Liste von 25 Windows Amazon Machine Images (AMIs), die routinemäßig gepatcht werden. Wenn Sie eine Automatisierung ausführen, die das `AWS-CreateManagedWindowsInstance`-Runbook verwendet, und Sie diese auf diese Resource Group ausrichten, erstellt Systems Manager eine untergeordnete Automatisierung für jede der 25 AMIs. Dies bedeutet, dass die Automatisierung aufgrund der Ausrichtung auf die Resource Group PatchedAMIs 25 Instances aus einer Liste von gepatchten AMIserstellt. Die übergeordnete Automatisierung ist abgeschlossen, wenn alle untergeordneten Automatisierungen abgeschlossen sind oder einen endgültigen Status erreicht haben.

Der folgende AWS CLI-Befehl gilt für das Beispiel mit der Ressourcengruppe PatchAMIs. Der Befehl verwendet den Parameter AmiId für die Option `--target-parameter-name`. Der Befehl enthält keinen zusätzlichen Parameter, der festlegt, welche Art von Instance aus jeder AMI erstellt werden soll. Das `AWS-CreateManagedWindowsInstance`-Runbook verwendet standardmäßig den Instance-Typ t2.medium, so dass dieser Befehl 25 t2.medium Amazon-EC2-Instances für Windows Server erstellt.

```
aws ssm start-automation-execution \
    --document-name AWS-CreateManagedWindowsInstance \
    --targets Key=ResourceGroup,Values=PatchedAMIs  \
    --target-parameter-name AmiId
```

Das folgende Konsolenbeispiel verwendet eine Ressourcengruppe mit dem Namen t2-micro-instances.



Ausrichtung auf Parameterwerte

Sie können auch einen Parameterwert zur Ausrichtung verwenden. Geben Sie `ParameterValues` als Schlüssel und dann den spezifischen Ressourcenwert für die Ausführung der Automatisierung ein. Wenn Sie mehrere Werte angeben, führt Systems Manager eine untergeordnete Automatisierung für jeden angegebenen Wert aus.

Nehmen Sie beispielsweise an, dass das Runbook einen InstanceID-Parameter enthält. Wenn Sie die Werte des InstanceID-Parameters beim Ausführen von Automation verwenden, führt Systems Manager eine untergeordnete Automatisierung für jeden angegebenen Instance-ID-Wert aus. Die übergeordnete Automatisierung ist abgeschlossen, wenn Automatisierung die Ausführung jeder angegebenen Instance abgeschlossen hat oder wenn die Automatisierung fehlschlägt. Sie können maximal 50 Parameterwerte für die Ausrichtung verwenden.

Im folgenden Beispiel wird das `AWS-CreateImage`-Runbook verwendet. Der angegebene Ziel-Parametername ist InstanceId. Der Schlüssel verwendetParameterValues. Die Werte sind zwei Amazon-EC2-Instance-IDs. Dieser Befehl erstellt einen Automatisierung für jede Instance, wodurch eine AMI von jeder Instance erzeugt wird.

```
aws ssm start-automation-execution
    --document-name AWS-CreateImage \
    --target-parameter-name InstanceId \
    --targets Key=ParameterValues,Values=i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE
```

> **Note**
>
> `AutomationAssumeRole` ist kein gültiger Parameter. Wählen Sie dieses Element nicht aus, wenn Sie die Automatisierung ausführen, die auf einen Parameterwert abzielt.

Ausrichtung auf Parameterwert-Maps

Die Option `TargetMaps` erweitert die Möglichkeiten zur Ausrichtung auf `ParameterValues`. Sie können ein Array von Parameterwerten mithilfe von `TargetMaps` auf der Befehlszeile eingeben. Sie können maximal 50 Parameterwerte in der Befehlszeile angeben. Wenn Sie Befehle ausführen möchten, die mehr als 50 Parameterwerte angeben, können Sie die Werte in einer JSON-Datei eingeben. Sie können dann die Datei von der Befehlszeile aus aufrufen.

> **Note**
>
> Die `TargetMaps`-Option wird in der Konsole nicht unterstützt.

Verwenden Sie das folgende Format, um mehrere Parameterwerte angeben, indem Sie die Option `TargetMaps` in einem Befehl verwenden. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

```
aws ssm start-automation-execution \
    --document-name runbook name \
    --target-maps “parameter=value, parameter 2=value, parameter 3=value”  “parameter
 4=value, parameter 5=value, parameter 6=value”
```

Wenn Sie mehr als 50 Parameterwerte für die Option `TargetMaps` angeben möchten, geben Sie die Werte mit dem folgenden JSON-Format an. Die Verwendung einer JSON-Datei verbessert auch die Lesbarkeit bei mehreren Parameterwerten.

```
[

    {“parameter”: "value", “parameter 2”: "value", “parameter 3”: "value"},

    {“parameter 4”: "value", “parameter 5”: "value", "parameter 6": "value"}
```

```
]
```

Speichen Sie die Datei mit der Dateierweiterung .json. Sie können die Datei mit dem folgenden Befehl ausführen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
aws ssm start-automation-execution \
    --document-name runbook name \
    --parameters input parameters \
    --target-maps path to file/file name.json
```

Sie können die auch aus einem Amazon Simple Storage Service (Amazon S3)-Bucket herunterladen, sofern Sie über die Berechtigung zum lesen von Daten aus dem Bucket verfügen. Verwenden Sie das folgende Befehlsformat. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

```
aws ssm start-automation-execution \
    --document-name runbook name \
    --target-maps http://amzn-s3-demo-bucket.s3.amazonaws.com/file_name.json
```

Hier sehen Sie ein Beispiel für ein Szenario, das Ihnen dabei hilft, die Option `TargetMaps` zu verstehen. In diesem Szenario möchte ein Benutzer Amazon EC2-Instances verschiedener Typen aus verschiedenen AMIs erstellen. Für diese Aufgabe erstellt der Benutzer erstellt ein Runbook mit dem Namen AMI_Testing. Dieses Runbook definiert zwei Eingabeparameter: `instanceType` und `imageId`.

```
{
  "description": "AMI Testing",
  "schemaVersion": "0.3",
  "assumeRole": "{{assumeRole}}",
  "parameters": {
    "assumeRole": {
      "type": "String",
      "description": "Role under which to run the automation",
      "default": ""
    },
    "instanceType": {
      "type": "String",
      "description": "Type of EC2 Instance to launch for this test"
    },
```

```
    "imageId": {
      "type": "String",
      "description": "Source AMI id from which to run instance"
    }
  },
  "mainSteps": [
    {
      "name": "runInstances",
      "action": "aws:runInstances",
      "maxAttempts": 1,
      "onFailure": "Abort",
      "inputs": {
        "ImageId": "{{imageId}}",
        "InstanceType": "{{instanceType}}",
        "MinInstanceCount": 1,
        "MaxInstanceCount": 1
      }
    }
  ],
  "outputs": [
    "runInstances.InstanceIds"
  ]
}
```

Dann gibt der Benutzer die folgenden Ziel-Parameterwerte in einer Datei mit dem Namen `AMI_instance_types.json` an.

```
[
  {
    "instanceType" : ["t2.micro"],
    "imageId" : ["ami-b70554c8"]
  },
  {
    "instanceType" : ["t2.small"],
    "imageId" : ["ami-b70554c8"]
  },
  {
    "instanceType" : ["t2.medium"],
    "imageId" : ["ami-cfe4b2b0"]
  },
  {
    "instanceType" : ["t2.medium"],
    "imageId" : ["ami-cfe4b2b0"]
```

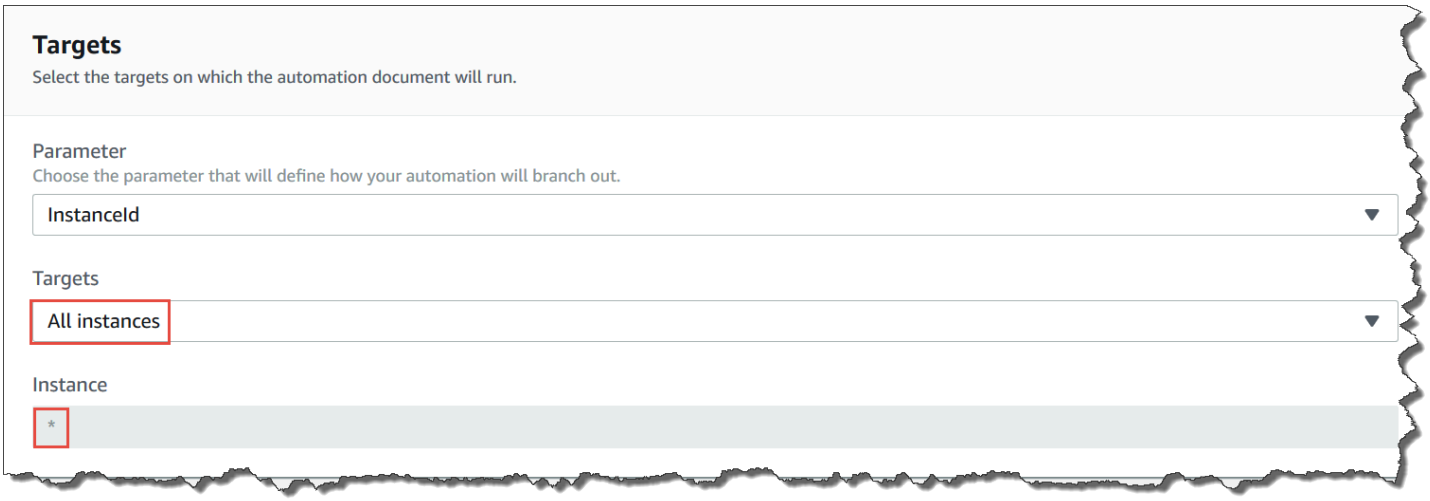
```
  },
  {
    "instanceType" : ["t2.medium"],
    "imageId" : ["ami-cfe4b2b0"]
  }
]
```

Mit dem folgenden Befehl kann der Benutzer die Automatisierung ausführen und die fünf EC2-Instances erstellen, die in `AMI_instance_types.json` definiert sind.

```
aws ssm start-automation-execution \
    --document-name AMI_Testing \
    --target-parameter-name imageId \
    --target-maps file:///home/TestUser/workspace/runinstances/AMI_instance_types.json
```

Ausrichtung auf alle Amazon-EC2-Instances

Sie können eine Automatisierung für alle Amazon-EC2-Instances im aktuellen AWS-Konto und AWS-Region ausführen, indem Sie Alle Instances in der Liste Ziele auswählen. Wenn Sie zum Beispiel alle Amazon-EC2-Instances Ihres AWS-Konto und der aktuellen AWS-Region neu starten möchten, können Sie das **`AWS-RestartEC2Instance`**-Runbook und dann Alle Instances in der Liste Ziele auswählen.



Nachdem Sie Alle Instances gewählt haben, versieht Systems Manager das Instance-Feld einem Sternchen (*) und macht das Feld für Änderungen nicht verfügbar (das Feld ist ausgegraut). Systems Manager macht auch das InstanceId-Feld im Eingabeparameter-Feld für Änderungen nicht verfügbar. Diese Felder für Änderungen nicht verfügbar zu machen, ist ein erwartetes Verhalten, wenn Sie sich dafür entscheiden, alle Instances abzudecken.

Steuern von Automatisierungen im großen Maßstab

Sie können die Bereitstellung einer Automatisierung innerhalb einer Flotte von AWS-Ressourcen steuern, indem Sie einen Gleichzeitigkeitswert und einen Fehlergrenzwert angeben. Die Gleichzeitigkeits- und die Fehlergrenzwertfunktion werden gemeinsam als Ratensteuerungen bezeichnet.

Nebenläufigkeit

Mit dem Gleichzeitigkeitswert können Sie angeben, wie viele Ressourcen eine Automatisierung gleichzeitig ausführen können. Die Gleichzeitigkeitsfunktion hilft dabei, die Auswirkungen auf Ihre Ressourcen oder Ausfälle werden der Ausführung einer Automatisierung zu begrenzen. Sie können entweder eine absolute Anzahl an Ressourcen, z. B. 20, oder einen Prozentsatz des festgelegten Ziels, beispielsweise 10 %, festlegen.

Das Warteschlangensystem übermittelt die Automatisierung an eine einzelne Ressource und wartet, bis der erste Aufruf abgeschlossen ist, bevor die Automatisierung an zwei weitere Ressourcen geschickt wird. Das System sendet die Automatisierung exponentiell an mehrere Ressourcen, bis der Gleichzeitigkeitswert erreicht ist.

Fehlerschwellenwerte

Verwenden Sie einen Fehlerschwellenwert, um festzulegen, wie viele Automatisierungen fehlschlagen dürfen, bevor AWS Systems Manager damit aufhört, die Automatisierung an andere Ressourcen zu senden. Sie können entweder eine absolute Anzahl an Fehlern, z. B. 10, oder einen Prozentsatz des festgelegten Ziels, beispielsweise 10 % festlegen.

Wenn Sie z. B. die absolute Zahl von 3 Fehlern angeben, führt das System keine Automatisierung mehr aus, wenn der vierte Fehler empfangen wird. Wenn Sie 0 angeben, führt das System keine weitere Automatisierung auf zusätzlichen Zielen aus, nachdem das erste Fehlerergebnis zurückgegeben wird.

Wenn Sie eine Automatisierung etwa an 50 Instances senden und den Fehlerschwellenwert auf 10 % festlegen, sendet das System keinen Befehl mehr an weitere Instances, wenn der fünfte Fehler empfangen wird. Aufrufe, die bereits eine Automatisierung ausführen, wenn ein Fehlerschwellenwert erreicht wird, können abgeschlossen werden, einige dieser Automatisierungen können jedoch dennoch fehlschlagen. Wenn Sie sicherstellen müssen, dass nicht mehr Fehlern als der angegebene Wert für den Fehlergrenzwert auftreten, setzen Sie den Wert für die Concurrency (Gleichzeitigkeit) auf 1, sodass die Automatisierungen jeweils einzeln ausgeführt werden.

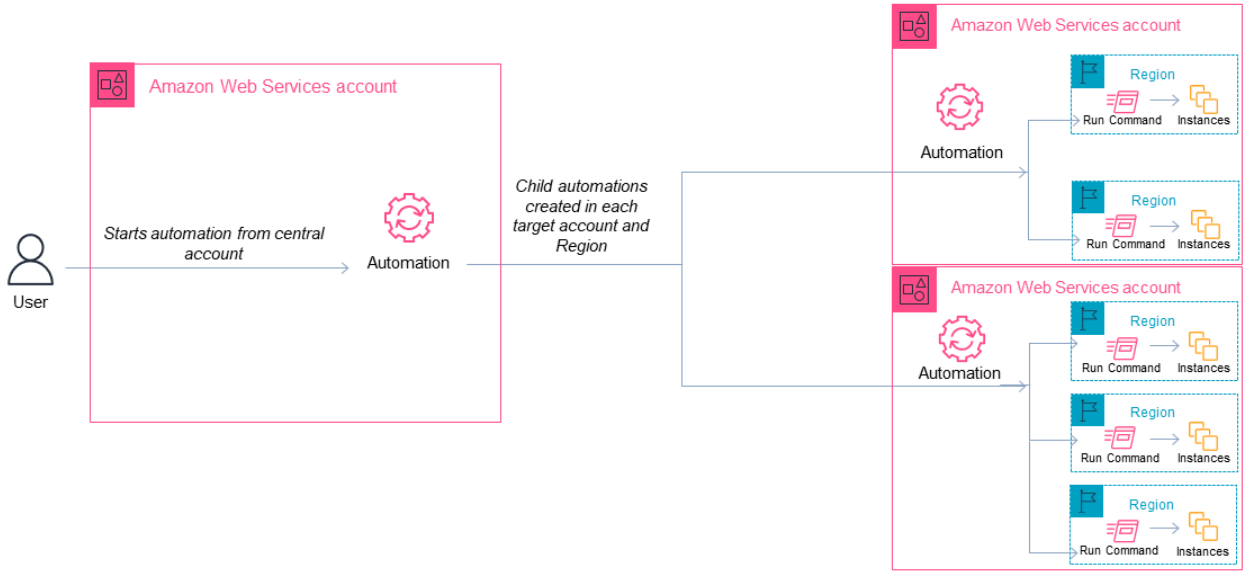
Automatisierungen in mehreren Konten AWS-Regionen ausführen

Sie können AWS Systems Manager Automatisierungen für mehrere AWS-Konten und/oder AWS-Regionen AWS Organizations organisatorische Einheiten (OUs) von einem zentralen Konto aus ausführen. Automation ist ein Tool in AWS Systems Manager. Durch die Ausführung von Automatisierungen in mehreren Regionen und Konten OUs wird der Zeitaufwand für die Verwaltung Ihrer AWS Ressourcen reduziert und gleichzeitig die Sicherheit Ihrer Computerumgebung verbessert.

Sie können z. B. Folgendes tun, indem Sie Automatisierungs-Runbooks verwenden:

- Implementieren Sie Patches und Sicherheitsupdates zentral.
- Korrigieren Sie Compliance-Abweichungen bei VPC-Konfigurationen oder Amazon-S3-Bucket-Richtlinien.
- Verwalten Sie Ressourcen wie Amazon Elastic Compute Cloud (Amazon EC2) EC2 -Instances in großem Umfang.

Das folgende Diagramm zeigt ein Beispiel für einen Benutzer, der das `AWS-RestartEC2Instances`-Runbook in mehreren Regionen und Konten von einem zentralen Konto aus ausführt. Die Automatisierung sucht die Instances unter Verwendung der angegebenen Tags in den Zielregionen und -konten.



Auswählen eines zentralen Kontos für Automation

Wenn Sie Automatisierungen überall ausführen möchten OUs, muss das zentrale Konto über die Berechtigungen verfügen, um alle Konten in der OUs aufzulisten. Dies ist nur über ein delegiertes Administratorkonto oder das Verwaltungskonto der Organisation möglich. Wir empfehlen Ihnen, AWS Organizations bewährte Methoden zu befolgen und ein delegiertes Administratorkonto zu verwenden. Weitere Informationen zu AWS Organizations bewährten Methoden finden Sie unter Bewährte Methoden für das Verwaltungskonto im AWS Organizations Benutzerhandbuch. Um ein delegiertes Administratorkonto für Systems Manager zu erstellen, können Sie den `register-delegated-administrator` Befehl mit dem verwenden, AWS CLI wie im folgenden Beispiel gezeigt.

```
aws organizations register-delegated-administrator \
    --account-id delegated admin account ID \
    --service-principal ssm.amazonaws.com
```

Wenn Sie Automatisierungen für mehrere Konten ausführen möchten, die nicht von AWS Organizations verwaltet werden, empfehlen wir, ein dediziertes Konto für die Automatisierungsverwaltung zu erstellen. Die Ausführung aller kontoübergreifenden Automatisierungen über ein dediziertes Konto vereinfacht die Verwaltung von IAM-Berechtigungen, die Fehlerbehebung und schafft eine Trennungsebene zwischen Betrieb und Verwaltung. Dieser

Ansatz wird auch empfohlen, wenn Sie einzelne Konten verwenden AWS Organizations, aber nicht OUs darauf abzielen möchten.

So funktioniert das Ausführen von Automatisierungen

Das Ausführen von Automatisierungen über mehrere Regionen und Konten hinweg OUs funktioniert wie folgt:

1. Melden Sie sich bei dem Konto an, das Sie als zentrales Automation-Konto konfigurieren möchten.
2. Verwenden Sie das Einrichten von Managementkonto-Berechtigungen für regionen- und kontenübergreifende Automatisierungen.-Verfahren in diesem Thema, um die folgenden IAM-Rollen zu erstellen:
   - **`AWS-SystemsManager-AutomationAdministrationRole`**- Diese Rolle gibt dem Benutzer die Erlaubnis, Automatisierungen in mehreren Konten auszuführen und. OUs
   - **`AWS-SystemsManager-AutomationExecutionRole`** – Diese Rolle erteilt dem Benutzer die Berechtigung, Automatisierungen in den Zielkonten auszuführen.
3. Wählen Sie das Runbook, die Regionen und die Konten oder den Ort aus, OUs an dem Sie die Automatisierung ausführen möchten.

   > **Note**
   >
   > Stellen Sie sicher, dass die Zielorganisationseinheit die gewünschten Konten enthält. Wenn Sie ein benutzerdefiniertes Runbook auswählen, muss das Runbook für alle Zielkonten freigegeben werden. Weitere Informationen zum Teilen von Runbooks finden Sie unter Freigeben von SSM-Dokumenten. Weitere Informationen zur Verwendung von freigegebenen Runbooks finden Sie unter Verwenden von freigegebenen SSM-Dokumenten.

4. Führen Sie die Automatisierung aus.
5. Verwenden Sie die DescribeAutomationExecutionsAPI-Operationen GetAutomationExecutionDescribeAutomationStepExecutions, und von der AWS Systems Manager Konsole aus oder verwenden Sie, AWS CLI um den Automatisierungsfortschritt zu überwachen. Die Ausgabe der Schritte für die Automatisierung in Ihrem primären Konto wird die `AutomationExecutionId` der untergeordneten Automatisierungen sein. Um die Ausgabe der untergeordneten Automatisierungen anzuzeigen, die in Ihren Zielkonten erstellt wurden, müssen Sie das entsprechende Konto, die Region und die `AutomationExecutionId` In Ihrer Anfrage angeben.

Einrichten von Managementkonto-Berechtigungen für regionen- und kontenübergreifende Automatisierungen.

Verwenden Sie das folgende Verfahren, um die erforderlichen IAM-Rollen für die Systems Manager Automation regionen- und kontenübergreifende Ausführung von Automation mit AWS CloudFormation zu erstellen. In diesem Verfahren wird beschrieben, wie Sie die `AWS-SystemsManager-AutomationAdministrationRole`-Rolle erstellen. Sie müssen nur diese Rolle im zentralen Automation-Konto erstellen. In diesem Verfahren wird auch beschrieben, wie Sie die `AWS-SystemsManager-AutomationExecutionRole`-Rolle erstellen. Sie müssen diese Rolle in jedem Konto erstellen, das für die Ausführung von regionen- und kontenübergreifenden Automatisierungen verwendet werden soll. Wir empfehlen CloudFormation StackSets , sie zu verwenden, um die `AWS-SystemsManager-AutomationExecutionRole` Rolle in den Konten zu erstellen, auf die Sie für die Ausführung von Automatisierungen mit mehreren Regionen und Konten abzielen möchten.

Um die erforderliche IAM-Administrationsrolle für Automatisierungen mit mehreren Regionen und mehreren Konten zu erstellen, verwenden Sie CloudFormation

1. Laden Sie das `AWS-SystemsManager-AutomationAdministrationRole.zip` herunter und entpacken Sie es.

   –oder–

   Wenn Ihre Konten verwaltet werden von. AWS Organizations `AWS-SystemsManager-AutomationAdministrationRole (org).zip`

   Diese Dateien enthalten die jeweiligen `AWS-SystemsManager-AutomationAdministrationRole (org).yaml` CloudFormation Vorlagendateien `AWS-SystemsManager-AutomationAdministrationRole.yaml` und die Vorlagendateien.
2. Öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
3. Wählen Sie Stack erstellen aus.
4. Wählen Sie im Abschnitt Vorlage angeben die Option Vorlage hochladen.
5. Wählen Sie Datei auswählen und dann je nach Ihrer Auswahl in Schritt 1 die Datei `AWS-SystemsManager-AutomationAdministrationRole.yaml` oder die `AWS-SystemsManager-AutomationAdministrationRole (org).yaml` CloudFormation Vorlagendatei aus.
6. Wählen Sie Weiter aus.

7. Geben Sie auf der Seite Stack-Details angeben im Feld Stack-Name einen Namen ein.
8. Wählen Sie Weiter aus.
9. Geben Sie auf der Seite Stack-Optionen konfigurieren Werte für die Optionen ein, die Sie verwenden möchten. Wählen Sie Weiter aus.
10. Scrollen Sie auf der Seite „Überprüfen“ nach unten und wählen Sie die Option Ich bestätige, dass CloudFormation möglicherweise IAM-Ressourcen mit benutzerdefinierten Namen erstellt werden.
11. Wählen Sie Stack erstellen aus.

CloudFormation zeigt den Status CREATE_IN_PROGRESS für ungefähr drei Minuten an. Der Status wechselt zu CREATE_COMPLETE.

Sie müssen die folgende Vorgehensweise in jedem Konto, für das Sie regionen- und kontenübergreifende Automatisierungen ausführen möchten, wiederholen.

Um die erforderliche IAM-Automatisierungsrolle für Automatisierungen mit mehreren Regionen und Konten zu erstellen, verwenden Sie CloudFormation

1. Laden Sie das `AWS-SystemsManager-AutomationExecutionRole.zip` herunter.

   –oder

   Wenn Ihre Konten verwaltet werden von. AWS Organizations `AWS-SystemsManager-AutomationExecutionRole (org).zip`

   Diese Dateien enthalten die `AWS-SystemsManager-AutomationExecutionRole.yaml`- bzw. die `AWS-SystemsManager-AutomationExecutionRole (org).yaml`CloudFormation -Vorlagendateien.
2. Öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
3. Wählen Sie Stack erstellen aus.
4. Wählen Sie im Abschnitt Vorlage angeben die Option Vorlage hochladen.
5. Wählen Sie Datei auswählen und dann je nach Ihrer Auswahl in Schritt 1 die Datei `AWS-SystemsManager-AutomationExecutionRole.yaml` oder die `AWS-SystemsManager-AutomationExecutionRole (org).yaml` CloudFormation Vorlagendatei aus.
6. Wählen Sie Weiter aus.
7. Geben Sie auf der Seite Stack-Details angeben im Feld Stack-Name einen Namen ein.

8. Geben Sie im Abschnitt Parameter in das AdminAccountIdFeld die ID für das zentrale Automation-Konto ein.
9. Wenn Sie diese Rolle für eine AWS Organizations Umgebung einrichten, gibt es im Abschnitt ein weiteres Feld namens OrganizationID. Geben Sie die ID Ihrer AWS Organisation ein.
10. Wählen Sie Weiter aus.
11. Geben Sie auf der Seite Stack-Optionen konfigurieren Werte für die Optionen ein, die Sie verwenden möchten. Wählen Sie Weiter aus.
12. Scrollen Sie auf der Seite „Überprüfen“ nach unten und wählen Sie die Option Ich bestätige, dass CloudFormation möglicherweise IAM-Ressourcen mit benutzerdefinierten Namen erstellt werden.
13. Wählen Sie Stack erstellen aus.

CloudFormation zeigt den Status CREATE_IN_PROGRESS für ungefähr drei Minuten an. Der Status wechselt zu CREATE_COMPLETE.

Ausführen von Automatisierungen in mehreren Regionen und Konten (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie mithilfe der Systems Manager-Konsole eine Automatisierung in mehreren Regionen und Konten über das Automation-Managementkonto ausführen.

Bevor Sie beginnen

Bevor Sie das folgende Verfahren ausführen, beachten Sie die folgenden Informationen:

- Der Benutzer oder die Rolle, mit der Sie eine Automation für mehrere Regionen oder Konten ausführen, muss über die Berechtigung `iam:PassRole` für die Rolle `AWS-SystemsManager-AutomationAdministrationRole` verfügen.
- AWS-Konto IDs oder OUs wo Sie die Automatisierung ausführen möchten.
- Von Systems Manager unterstützte Regionen, in denen Sie die Automatisierung ausführen möchten.
- Den Tag-Schlüssel und den Tag-Wert oder den Namen der Ressourcengruppe für die Ausführung der Automatisierung.

So führen Sie eine Automatisierung in mehreren Regionen und Konten aus

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Automatisierung und Automatisierung ausführen aus.
3. Wählen Sie in der Liste Automation-Dokument ein Runbook. Wählen Sie eine oder mehrere Optionen im Bereich Dokumentkategorien, um SSM-Dokumente nach ihrem Zweck zu filtern. Um ein Runbook anzuzeigen, das Sie besitzen, wählen Sie die Im Besitz von mir-Registerkarte. Um ein Runbook anzuzeigen, das für Ihr Konto freigegeben ist, wählen Sie die Registerkarte Mit mir geteilt. Um alle Runbooks anzuzeigen, wählen Sie die Registerkarte Alle Dokumente.

   > **Note**
   >
   > Sie können Informationen zu einem Runbook einsehen, indem Sie den Runbook-Namen auswählen.

4. Überprüfen Sie im Abschnitt Dokument-Details, ob Dokumentversion auf die Version gesetzt ist, die Sie ausführen möchten. Das System bietet die folgenden Versionsoptionen:

   - Standardversion zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird und eine neue Standardversion zugewiesen ist.
   - Letzte Version zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird, und Sie die Version auszuführen möchten, die zuletzt aktualisiert wurde.
   - 1 (Standard) – Wählen Sie diese Option zur Ausführung der ersten Version des Dokuments, welches der Standard ist.

5. Wählen Sie Weiter aus.
6. Wählen Sie auf der Seite Execute automation document (Automation-Dokument ausführen) die Option Multi-account and Region (Mehrere Konten und Regionen).
7. Verwenden Sie im Abschnitt Zielkonten und Regionen das Feld Konten, Organisationseinheiten (OUs) und Stammverzeichnisse, um die verschiedenen Organisationseinheiten AWS-Konten oder AWS Organisationseinheiten (OUs) anzugeben, in denen Sie die Automatisierung ausführen möchten. Trennen Sie mehrere Konten oder OUs durch ein Komma voneinander.

   a. (Optional) Aktivieren Sie das OUs Kontrollkästchen „Kind einbeziehen“, um alle untergeordneten Organisationseinheiten innerhalb der angegebenen OUs Organisationseinheiten einzubeziehen.

   b. (Optional) Geben Sie im Feld Konten und Organisationseinheiten ausschließen (OUs) eine durch Kommas getrennte Liste der Konten IDs und Organisationseinheiten ein IDs , die Sie aus den oben eingegebenen erweiterten Entitäten ausschließen möchten.

8. Verwenden Sie die Liste Regionen zur Auswahl einer oder mehrerer Regionen für die Ausführung der Automatisierung.
9. Verwenden Sie die Optionen für Multi-Region and account rate control (Regionen- und kontenübergreifende Ratensteuerung), um die Automatisierungen auf eine begrenzte Anzahl von Konten in einer begrenzten Anzahl von Regionen zu beschränken. Diese Optionen schränken nicht die Anzahl der AWS -Ressourcen ein, die die Automatisierungen ausführen können.

    a. Wählen Sie im Abschnitt Location (account-Region pair) concurrency (Standort- (Konto-Region-Paar) Gleichzeitigkeit) eine Option, um die Anzahl der Automatisierungen zu begrenzen, die gleichzeitig in mehreren Konten und Regionen ausgeführt werden können. Wenn Sie sich beispielsweise dafür entscheiden, eine Automatisierung in fünf (5) auszuführen AWS-Konten, die sich in vier (4) befinden AWS-Regionen, führt Systems Manager Automatisierungen in insgesamt 20 Konto-Region-Paaren aus. Sie können diese Option verwenden, um eine absolute Zahl anzugeben, z. B. **2**, sodass die Automatisierung nur in 2 Konto-Region-Paaren gleichzeitig ausgeführt wird. Sie können aber auch einen Prozentsatz der Konto-Region-Paare angeben, der gleichzeitig ausgeführt werden kann. Beispielsweise geben Sie bei 20 Konto-Region-Paaren 20 % an: Dann wird die Automatisierung in maximal fünf (5) Konto-Region-Paaren gleichzeitig ausgeführt.

        - Wählen Sie targets (Ziele) aus, um eine absolute Anzahl von Konto-Region-Paaren einzugeben, die die Automatisierung gleichzeitig ausführen können.
        - Wählen Sie percent (Prozent) aus, um einen Prozentsatz von Konto-Region-Paaren einzugeben, die die Automatisierung gleichzeitig ausführen können.

    b. Wählen Sie im Abschnitt Fehlerschwellenwert eine Option aus:

        - Wählen Sie Fehler, um eine absolute Anzahl von zulässigen Fehlern anzugeben, bevor die Automation damit aufhört, die Automatisierung an andere Ressourcen zu senden.
        - Wählen Sie percentage aus, um einen Prozentsatz von zulässigen Fehlern anzugeben, bevor Automation damit aufhört, die Automatisierung an andere Ressourcen zu senden.
10. Wählen Sie im Abschnitt Ziele die Ausrichtung auf die AWS -Ressourcen für die Ausführung der Automation. Diese Optionen sind erforderlich.

    a. Wählen Sie in der Liste Parameter einen Parameter aus. Die Elemente in der Liste Parameter richten sich nach den Parametern in dem Automation-Runbook, das Sie zu Beginn dieses Verfahrens ausgewählt haben. Durch Auswahl eines Parameters legen Sie den Typ der Ressource fest, für die der Automation-Workflow ausgeführt wird.

    b. Wählen Sie in der Liste Ziele aus, wie Sie Ressourcen als Ziele verwenden möchten.

i. Wenn Sie die Zielressourcen mithilfe von Parameterwerten ausgewählt haben, geben Sie den Parameterwert für den gewählten Parameter im Feld Eingabeparameter eingeben ein.

ii. Wenn Sie Ressourcen mithilfe von als Ziel verwenden möchten AWS -Ressourcengruppen, wählen Sie den Namen der Gruppe aus der Liste der Ressourcengruppen aus.

iii. Wenn Sie die Zielressourcen mithilfe von Tags ausgewählt haben, geben Sie den Tag-Schlüssel und (optional) den Tag-Wert in die entsprechenden Felder ein. Wählen Sie Hinzufügen aus.

iv. Wenn Sie ein Automatisierungs-Runbook auf allen Instanzen im aktuellen AWS-Konto und ausführen möchten AWS-Region, wählen Sie Alle Instanzen aus.

11. Geben Sie im Abschnitt Eingabeparameter die erforderlichen Eingaben an. Wählen Sie die `AWS-SystemsManager-AutomationAdministrationRole` IAM-Servicerolle aus der AutomationAssumeRoleListe aus.

> **Note**
>
> Möglicherweise müssen Sie einige der Optionen im Abschnitt Eingabeparameter nicht auswählen. Dies liegt daran, dass Sie Ressourcen in mehreren Regionen und Konten mithilfe von Tags oder einer Ressourcengruppe als Ziele ausgewählt haben. Wenn Sie beispielsweise das `AWS-RestartEC2Instance` Runbook ausgewählt haben, müssen Sie IDs im Abschnitt Eingabeparameter keine Instanz angeben oder auswählen. Die Automatisierung sucht die Instances für den Neustart mit den von Ihnen angegebenen Tags.

12. (Optional) Wählen Sie einen CloudWatch Alarm aus, der zur Überwachung auf Ihre Automatisierung angewendet werden soll. Um Ihrer Automatisierung einen CloudWatch Alarm zuzuweisen, muss der IAM-Principal, der die Automatisierung startet, über die Genehmigung für die `iam:createServiceLinkedRole` Aktion verfügen. Weitere Informationen zu CloudWatch Alarmen finden Sie unter CloudWatch Amazon-Alarme verwenden. Beachten Sie, dass, wenn Ihr Alarm ausgelöst wird, die Automatisierung abgebrochen wird und alle von Ihnen definierten `0nCancel`-Schritte ausgeführt werden. Wenn Sie dies verwenden AWS CloudTrail, wird der API-Aufruf in Ihrem Trail angezeigt.

13. Verwenden Sie die Optionen im Abschnitt Preissteuerung, um die Anzahl der AWS Ressourcen zu beschränken, mit denen die Automatisierung innerhalb jedes Konto-Region-Paares ausgeführt werden kann.

    Wählen Sie im Abschnitt Gleichzeitigkeit eine Option aus:

    - Wählen Sie Ziele aus, um eine absolute Anzahl von Zielen einzugeben, die den Automation-Workflow gleichzeitig ausführen können.
    - Wählen Sie Prozentsatz aus, um einen Prozentsatz der Ziele anzugeben, die den Automation-Workflow gleichzeitig ausführen können.
14. Wählen Sie im Abschnitt Fehlerschwellenwert eine Option aus:

    - Wählen Sie Fehler, um eine absolute Anzahl von zulässigen Fehlern anzugeben, bevor Automation damit aufhört, den Workflow an andere Ressourcen zu senden.
    - Wählen Sie Prozentsatz aus, um einen Prozentsatz von zulässigen Fehlern anzugeben, bevor Automation damit aufhört, den Workflow an andere Ressourcen zu senden.
15. Wählen Sie Ausführen.

Nach Abschluss einer Automatisierungsausführung können Sie die Ausführung mit denselben oder geänderten Parametern erneut ausführen. Weitere Informationen finden Sie unter Wiederholtes Ausführen von Automatisierungsausführungen.

Ausführen von Automatisierungen in mehreren Regionen und Konten (Befehlszeile)

Das folgende Verfahren beschreibt, wie Sie das AWS CLI (unter Linux oder Windows) verwenden oder AWS -Tools für PowerShell eine Automatisierung in mehreren Regionen und Konten vom Automatisierungsverwaltungskonto aus ausführen.

Bevor Sie beginnen

Bevor Sie das folgende Verfahren ausführen, beachten Sie die folgenden Informationen:

- AWS-Konto IDs oder OUs wo Sie die Automatisierung ausführen möchten.
- Von Systems Manager unterstützte Regionen, in denen Sie die Automatisierung ausführen möchten.
- Den Tag-Schlüssel und den Tag-Wert oder den Namen der Ressourcengruppe für die Ausführung der Automatisierung.

So führen Sie eine Automatisierung in mehreren Regionen und Konten aus

1. Installieren und konfigurieren Sie das AWS CLI oder das AWS -Tools für PowerShell, falls Sie es noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.

2. Verwenden Sie das folgende Format, um eine Automatisierung in mehreren Regionen und Konten auszuführen. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   Linux & macOS

   ```
   aws ssm start-automation-execution \
           --document-name runbook name \
           --parameters AutomationAssumeRole=arn:aws:iam::management account
    ID:role/AWS-SystemsManager-AutomationAdministrationRole \
           --target-parameter-name parameter name \
           --targets Key=tag key,Values=value \
           --target-locations Accounts=account ID,account ID
    2,Regions=Region,Region 2,ExecutionRoleName=AWS-SystemsManager-
   AutomationExecutionRole
   ```

   Windows

   ```
   aws ssm start-automation-execution ^
           --document-name runbook name ^
           --parameters AutomationAssumeRole=arn:aws:iam::management account
    ID:role/AWS-SystemsManager-AutomationAdministrationRole ^
           --target-parameter-name parameter name ^
           --targets Key=tag key,Values=value ^
           --target-locations Accounts=account ID,account ID
    2,Regions=Region,Region 2,ExecutionRoleName=AWS-SystemsManager-
   AutomationExecutionRole
   ```

   PowerShell

   ```
   $Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
       $Targets.Key = "tag key"
       $Targets.Values = "value"
   ```

```
    Start-SSMAutomationExecution `
        -DocumentName "runbook name" `
        -Parameter @{
        "AutomationAssumeRole"="arn:aws:iam::management account ID:role/AWS-
SystemsManager-AutomationAdministrationRole" } `
        -TargetParameterName "parameter name" `
        -Target $Targets `
        -TargetLocation @{
        "Accounts"="account ID","account ID 2";
        "Regions"="Region","Region 2";
        "ExecutionRoleName"="AWS-SystemsManager-AutomationExecutionRole" }
```

Beispiele: Ausführen einer Automatisierung in mehreren Regionen und Konten

Im Folgenden finden Sie Beispiele, wie Sie das AWS CLI und PowerShell zum Ausführen von Automatisierungen in mehreren Konten und Regionen mit einem einzigen Befehl verwenden können.

Beispiel 1: In diesem Beispiel werden EC2 Instanzen in drei Regionen einer gesamten AWS Organizations Organisation neu gestartet. Dies wird erreicht, indem die Root-ID der Organisation und das untergeordnete OUs Objekt als Ziel ausgewählt werden.

Linux & macOS

```
aws ssm start-automation-execution \
        --document-name "AWS-RestartEC2Instance" \
        --target-parameter-name InstanceId \
        --targets '[{"Key":"AWS::EC2::Instance","Values":["*"]}]' \
        --target-locations '[{
            "Accounts": ["r-example"],
            "IncludeChildOrganizationUnits": true,
            "Regions": ["us-east-1", "us-east-2", "us-west-2"]
        }]'
```

Windows

```
aws ssm start-automation-execution \
        --document-name "AWS-RestartEC2Instance" ^
        --target-parameter-name InstanceId ^
        --targets '[{"Key":"AWS::EC2::Instance","Values":["*"]}]' ^
        --target-locations '[{
```

```
            "Accounts": ["r-example"],
            "IncludeChildOrganizationUnits": true,
            "Regions": ["us-east-1", "us-east-2", "us-west-2"]
        }]'
```

PowerShell

```
Start-SSMAutomationExecution `
        -DocumentName "AWS-RestartEC2Instance" `
        -TargetParameterName "InstanceId" `
        -Targets '[{"Key":"AWS::EC2::Instance","Values":["*"]}]'
        -TargetLocation @{
            "Accounts"="r-example";
            "Regions"="us-east-1", "us-east-2", "us-west-2";
            "IncludeChildOrganizationUnits"=true}
```

Beispiel 2: In diesem Beispiel werden bestimmte EC2 Instanzen in verschiedenen Konten und Regionen neu gestartet.

> **Note**
>
> Die `TargetLocationMaxConcurrency` Option ist mit dem AWS CLI und AWS SDKs verfügbar.

Linux & macOS

```
aws ssm start-automation-execution \
        --document-name "AWS-RestartEC2Instance" \
        --target-parameter-name InstanceId \
        --target-locations '[{
            "Accounts": ["123456789012"],
            "Targets": [{
                "Key":"ParameterValues",
                "Values":["i-02573cafcfEXAMPLE", "i-0471e04240EXAMPLE"]
            }],
            "TargetLocationMaxConcurrency": "100%",
            "Regions": ["us-east-1"]
        }, {
            "Accounts": ["987654321098"],
```

```
            "Targets": [{
                "Key":"ParameterValues",
                "Values":["i-07782c72faEXAMPLE"]
            }],
            "TargetLocationMaxConcurrency": "100%",
            "Regions": ["us-east-2"]
        }]'
```

Windows

```
aws ssm start-automation-execution ^
        --document-name "AWS-RestartEC2Instance" ^
        --target-parameter-name InstanceId ^
        --target-locations '[{
            "Accounts": ["123456789012"],
            "Targets": [{
                "Key":"ParameterValues",
                "Values":["i-02573cafcfEXAMPLE", "i-0471e04240EXAMPLE"]
            }],
            "TargetLocationMaxConcurrency": "100%",
            "Regions": ["us-east-1"]
        }, {
            "Accounts": ["987654321098"],
            "Targets": [{
                "Key":"ParameterValues",
                "Values":["i-07782c72faEXAMPLE"]
            }],
            "TargetLocationMaxConcurrency": "100%",
            "Regions": ["us-east-2"]
        }]'
```

PowerShell

```
Start-SSMAutomationExecution `
        -DocumentName "AWS-RestartEC2Instance" `
        -TargetParameterName "InstanceId" `
        -Targets '[{"Key":"AWS::EC2::Instance","Values":["*"]}]'
        -TargetLocation @({
            "Accounts"="123456789012",
            "Targets"= @{
                "Key":"ParameterValues",
                "Values":["i-02573cafcfEXAMPLE", "i-0471e04240EXAMPLE"]
            },
```

```
            "TargetLocationMaxConcurrency"="100%",
            "Regions"=["us-east-1"]
        }, {
            "Accounts"="987654321098",
            "Targets": @{
                "Key":"ParameterValues",
                "Values":["i-07782c72faEXAMPLE"]
            },
            "TargetLocationMaxConcurrency": "100%",
            "Regions"=["us-east-2"]
        })
```

Beispiel 3: Dieses Beispiel zeigt die Angabe mehrerer Bereiche AWS-Konten und Regionen, in denen die Automatisierung mithilfe der `--target-locations-url` Option ausgeführt werden soll. Der Wert für diese Option muss eine JSON-Datei in einer öffentlich zugänglichen, vorsignierten Amazon S3-URL sein.

> **Note**
>
> `--target-locations-url`ist verfügbar, wenn Sie AWS CLI und verwenden AWS SDKs.

Linux & macOS

```
aws ssm start-automation-execution \
    --document-name "MyCustomAutomationRunbook" \
    --target-locations-url "https://amzn-s3-demo-bucket.s3.amazonaws.com/target-
locations.json"
```

Windows

```
aws ssm start-automation-execution ^
    --document-name "MyCustomAutomationRunbook" ^
    --target-locations-url "https://amzn-s3-demo-bucket.s3.amazonaws.com/target-
locations.json"
```

PowerShell

```
Start-SSMAutomationExecution `
    -DocumentName "MyCustomAutomationRunbook" `
    -TargetLocationsUrl "https://amzn-s3-demo-bucket.s3.amazonaws.com/target-
locations.json"
```

Beispielinhalt für die JSON-Datei:

```
[
{
        "Accounts": [ "123456789012", "987654321098", "456789123012" ],
        "ExcludeAccounts": [ "111222333444", "999888444666" ],
        "ExecutionRoleName": "MyAutomationExecutionRole",
        "IncludeChildOrganizationUnits": true,
        "Regions": [ "us-east-1", "us-west-2", "ap-south-1", "ap-northeast-1" ],
        "Targets": ["Key": "AWS::EC2::Instance", "Values": ["i-2"]],
        "TargetLocationMaxConcurrency": "50%",
        "TargetLocationMaxErrors": "10",
        "TargetsMaxConcurrency": "20",
        "TargetsMaxErrors": "12"
 }
]
```

Beispiel 4: In diesem Beispiel werden EC2 Instanzen in den 987654321098 Konten 123456789012 und neu gestartet, die sich in den us-west-1 Regionen us-east-2 und befinden. Die Instances müssen mit dem Tag-Schlüsselpaarwert Env-PROD markiert sein.

Linux & macOS

```
aws ssm start-automation-execution \
        --document-name AWS-RestartEC2Instance \
        --parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole \
        --target-parameter-name InstanceId \
        --targets Key=tag:Env,Values=PROD \
        --target-locations Accounts=123456789012,987654321098,Regions=us-
east-2,us-west-1,ExecutionRoleName=AWS-SystemsManager-AutomationExecutionRole
```

Windows

```
aws ssm start-automation-execution ^
        --document-name AWS-RestartEC2Instance ^
        --parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole ^
        --target-parameter-name InstanceId ^
        --targets Key=tag:Env,Values=PROD ^
        --target-locations Accounts=123456789012,987654321098,Regions=us-
east-2,us-west-1,ExecutionRoleName=AWS-SystemsManager-AutomationExecutionRole
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
    $Targets.Key = "tag:Env"
    $Targets.Values = "PROD"

    Start-SSMAutomationExecution `
        -DocumentName "AWS-RestartEC2Instance" `
        -Parameter @{
        "AutomationAssumeRole"="arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole" } `
        -TargetParameterName "InstanceId" `
        -Target $Targets `
        -TargetLocation @{
        "Accounts"="123456789012","987654321098";
        "Regions"="us-east-2","us-west-1";
        "ExecutionRoleName"="AWS-SystemsManager-AutomationExecutionRole" }
```

Beispiel 5: In diesem Beispiel werden EC2 Instanzen in den 987654321098 Konten 123456789012 und, die sich in der `eu-central-1` Region befinden, neu gestartet. Die Instanzen müssen Mitglieder der `prod-instances` AWS Ressourcengruppe sein.

Linux & macOS

```
aws ssm start-automation-execution \
        --document-name AWS-RestartEC2Instance \
        --parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole \
        --target-parameter-name InstanceId \
```

```
        --targets Key=ResourceGroup,Values=prod-instances \
        --target-locations Accounts=123456789012,987654321098,Regions=eu-
central-1,ExecutionRoleName=AWS-SystemsManager-AutomationExecutionRole
```

Windows

```
aws ssm start-automation-execution ^
        --document-name AWS-RestartEC2Instance ^
        --parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole ^
        --target-parameter-name InstanceId ^
        --targets Key=ResourceGroup,Values=prod-instances ^
        --target-locations Accounts=123456789012,987654321098,Regions=eu-
central-1,ExecutionRoleName=AWS-SystemsManager-AutomationExecutionRole
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
    $Targets.Key = "ResourceGroup"
    $Targets.Values = "prod-instances"

    Start-SSMAutomationExecution `
        -DocumentName "AWS-RestartEC2Instance" `
        -Parameter @{
        "AutomationAssumeRole"="arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole" } `
        -TargetParameterName "InstanceId" `
        -Target $Targets `
        -TargetLocation @{
        "Accounts"="123456789012","987654321098";
        "Regions"="eu-central-1";
        "ExecutionRoleName"="AWS-SystemsManager-AutomationExecutionRole" }
```

Beispiel 6: In diesem Beispiel werden EC2 Instanzen in der `ou-1a2b3c-4d5e6c` AWS Organisationseinheit (OU) neu gestartet. Die Instances befinden sich in den Regionen `us-west-1` und `us-west-2`. Die Instanzen müssen Mitglieder der `WebServices` AWS Ressourcengruppe sein.

Linux & macOS

```
aws ssm start-automation-execution \
        --document-name AWS-RestartEC2Instance \
        --parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole \
        --target-parameter-name InstanceId \
        --targets Key=ResourceGroup,Values=WebServices \
        --target-locations Accounts=ou-1a2b3c-4d5e6c,Regions=us-west-1,us-
west-2,ExecutionRoleName=AWS-SystemsManager-AutomationExecutionRole
```

Windows

```
aws ssm start-automation-execution ^
        --document-name AWS-RestartEC2Instance ^
        --parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole ^
        --target-parameter-name InstanceId ^
        --targets Key=ResourceGroup,Values=WebServices ^
        --target-locations Accounts=ou-1a2b3c-4d5e6c,Regions=us-west-1,us-
west-2,ExecutionRoleName=AWS-SystemsManager-AutomationExecutionRole
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
    $Targets.Key = "ResourceGroup"
    $Targets.Values = "WebServices"

    Start-SSMAutomationExecution `
        -DocumentName "AWS-RestartEC2Instance" `
        -Parameter @{
        "AutomationAssumeRole"="arn:aws:iam::123456789012:role/AWS-
SystemsManager-AutomationAdministrationRole" } `
        -TargetParameterName "InstanceId" `
        -Target $Targets `
        -TargetLocation @{
        "Accounts"="ou-1a2b3c-4d5e6c";
        "Regions"="us-west-1";
        "ExecutionRoleName"="AWS-SystemsManager-AutomationExecutionRole" }
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

Linux & macOS

```
{
        "AutomationExecutionId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
    }
```

Windows

```
{
        "AutomationExecutionId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
    }
```

PowerShell

```
4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE
```

3. Führen Sie den folgenden Befehl aus, um Details zu der Automatisierung anzuzeigen. *automation execution ID*Ersetzen Sie sie durch Ihre eigenen Informationen.

   Linux & macOS

   ```
   aws ssm describe-automation-executions \
           --filters Key=ExecutionId,Values=automation execution ID
   ```

   Windows

   ```
   aws ssm describe-automation-executions ^
           --filters Key=ExecutionId,Values=automation execution ID
   ```

   PowerShell

   ```
   Get-SSMAutomationExecutionList | `
           Where {$_.AutomationExecutionId -eq "automation execution ID"}
   ```

4. Führen Sie den folgenden Befehl aus, um Details über den Automatisierungsprozess anzuzeigen.

Linux & macOS

```
aws ssm get-automation-execution \
        --automation-execution-id 4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE
```

Windows

```
aws ssm get-automation-execution ^
        --automation-execution-id 4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE
```

PowerShell

```
Get-SSMAutomationExecution `
        -AutomationExecutionId a4a3c0e9-7efd-462a-8594-01234EXAMPLE
```

> **Note**
>
> Sie können auch den Status der Automatisierung in der Konsole überwachen. Wählen Sie in der Liste Automatisierungs-Ausführung die Automatisierung, die Sie gerade ausgeführt haben, und wählen Sie dann die Registerkarte Ausführungsschritte. Diese Registerkarte zeigt Ihnen den Status der Automatisierungs-Aktionen.

Weitere Informationen

Zentralisiertes regions- und kontenübergreifendes Patching mit AWS Systems Manager Automation

Führen Sie Automatisierungen auf EventBridge der Grundlage von Ereignissen aus

Sie können eine Automatisierung starten, indem Sie ein Runbook als Ziel eines EventBridge Amazon-Events angeben. Sie können Automatisierungen nach einem Zeitplan oder beim Eintreten eines bestimmten AWS -Ereignisses starten. Nehmen wir zum Beispiel an, Sie erstellen ein Runbook mit dem Namen BootStrapInstances, dass beim Start einer Instance Software auf einer Instance installiert. Um das BootStrapInstancesRunbook (und die entsprechende Automatisierung) als Ziel eines EventBridge Ereignisses anzugeben, erstellen Sie zunächst eine neue EventBridge Regel.

(Hier ist eine Beispielregel: Dienstname: EC2, Ereignistyp: Benachrichtigung über Änderung des EC2 Instanzstatus, Spezifischer Status: läuft, Beliebige Instanz.) Anschließend verwenden Sie die folgenden Verfahren, um mithilfe der EventBridge Konsole und AWS Command Line Interface ()AWS CLI das BootStrapInstancesRunbook als Ziel des Ereignisses anzugeben. Beim Starten einer neuen Instance führt das System die Automatisierung aus und installiert Software.

Weitere Informationen zum Erstellen eines Runbooks finden Sie unter Erstellen Ihrer eigenen Runbooks.

Erstellen eines EventBridge Ereignisses, das ein Runbook (Konsole) verwendet

Gehen Sie wie folgt vor, um ein Runbook als Ziel eines EventBridge Ereignisses zu konfigurieren.

So konfigurieren Sie ein Runbook als Ziel einer Ereignisregel EventBridge

1. Öffnen Sie die EventBridge Amazon-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Regeln aus.
3. Wählen Sie Regel erstellen aus.
4. Geben Sie einen Namen und eine Beschreibung für die Regel ein.

   Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben Region und auf demselben Event Bus haben.
5. Wählen Sie für Event Bus den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel auf entsprechende Ereignisse reagiert, die von Ihnen selbst stammen AWS-Konto, wählen Sie Standard. Wenn ein AWS-Service in Ihrem Konto ein Ereignis ausgibt, wird es immer an den Standard-Event-Bus Ihres Kontos weitergeleitet.
6. Wählen Sie aus, wie die Regel ausgelöst wird.

So erstellen Sie eine Regel auf der Basis von …	Vorgehensweise	
Veranstaltung	a. Bei Regeltyp wählen Sie Regel mit einem Ereignismuster aus. b. Wählen Sie Weiter aus. c. Wählen Sie als Eventquelle AWS Events oder	

So erstellen Sie eine Regel auf der Basis von …	Vorgehensweise	
	EventBridge Partnerev ents aus. d. Führen Sie im Abschnitt Event pattern (Ereignis muster) einen der folgenden Schritte aus: • Um eine Vorlage zum Erstellen Ihres Ereignismusters zu verwenden, wählen Sie Event pattern form (Ereignismusterfor mular) und wählen Sie Event source (Ereignis quelle), AWS servi ce (-Service) und Event type (Ereignis typ). Wenn Sie „Alle Ereignisse“ als Ereignist yp wählen, entsprech en alle von der AWS-Service ausgegebenen Ereignisse der Regel. Um die Vorlage anzupassen, wählen Sie Custom pattern (JSON editor) (Benutzer definiertes Muster (JSON-Editor)) und nehmen Sie die erforderlichen Änderungen vor.	

So erstellen Sie eine Regel auf der Basis von …	Vorgehensweise	
	• Wenn Sie ein benutzerdefiniertes Ereignismuster verwenden möchten, wählen Sie Custom pattern (JSON editor) (Benutzerdefiniertes Muster (JSON-Editor)) und erstellen Sie Ihr Ereignismuster.	

So erstellen Sie eine Regel auf der Basis von …	Vorgehensweise	
Plan	a. Wählen Sie unter Rule type (Regeltyp) die Option Schedule (Zeitplan) aus. b. Wählen Sie Weiter aus. c. Gehen Sie bei Schedule pattern (Zeitplanmuster) wie folgt vor: • Um den Zeitplan mithilfe eines Cron-Ausdrucks zu definieren, wählen Sie A fine-grained schedule that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month (Detaillierter Zeitplan, der zu einem bestimmten Zeitpunkt (z. B. 8:00 Uhr) PST am ersten Montag jedes Monats PST ausgeführt wird) und geben Sie den Cron-Ausdruck ein. • Um den Zeitplan mithilfe eines Rate-Ausdrucks zu definieren, wählen Sie A schedule that runs at a regular rate, such as every 10 minutes (Zeitplan, der mit einer regulären Rate läuft, z. B. alle 10 Minuten)	

So erstellen Sie eine Regel auf der Basis von …	Vorgehensweise	
	und geben Sie den Rate-Ausdruck ein.	

7. Wählen Sie Weiter aus.
8. Bei Zieltypen wählen Sie AWS -Service aus.
9. Für Select target (Ziel auswählen), wählen Sie Systems Manager Automation.
10. Wählen Sie für Dokument ein Runbook aus, das Sie verwenden möchten, wenn das Ziel aufgerufen wird.
11. Behalten Sie im Abschnitt Configure automation parameter(s) (Automatisierungsparameter konfigurieren) entweder die Standardparameterwerte bei (sofern verfügbar) oder geben Sie Ihre eigenen Werte ein.

> **Note**
>
> Um ein Ziel zu erstellen, müssen Sie bei jedem erforderlichen Parameter einen Wert angeben. Wenn Sie dies nicht tun, erstellt das System die Regel, aber die Regel wird nicht ausgeführt.

12. Für viele Zieltypen sind EventBridge Berechtigungen erforderlich, um Ereignisse an das Ziel zu senden. In diesen Fällen EventBridge kann die IAM-Rolle erstellt werden, die für die Ausführung Ihrer Regel erforderlich ist. Führen Sie eine der folgenden Aktionen aus:

    - Um automatisch eine IAM-Rolle zu erstellen, wählen Sie Eine neue Rolle für diese spezifische Ressource erstellen aus.
    - Wenn Sie eine zuvor erstellte IAM-Rolle verwenden möchten, wählen Sie Use existing role (Vorhandene Rolle verwenden) und wählen Sie die vorhandene Rolle aus der Dropdown-Liste aus. Beachten Sie, dass Sie möglicherweise die Vertrauensrichtlinie für Ihre IAM-Rolle aktualisieren müssen, um sie einzubeziehen. EventBridge Im Folgenden wird ein Beispiel gezeigt:

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "events.amazonaws.com",
                    "ssm.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

13. Wählen Sie Weiter aus.
14. (Optional) Geben Sie ein oder mehrere Tags für die Regel ein. Weitere Informationen finden Sie unter Tagging Your Amazon EventBridge Resources im EventBridge Amazon-Benutzerhandbuch.
15. Wählen Sie Weiter aus.
16. Überprüfen Sie die Details der Regel und wählen Sie dann Regel erstellen aus.

Erstellen Sie ein EventBridge Ereignis, das ein Runbook (Befehlszeile) verwendet

Das folgende Verfahren beschreibt, wie Sie die AWS CLI (unter Linux oder Windows) verwenden oder AWS -Tools für PowerShell eine EventBridge Ereignisregel erstellen und ein Runbook als Ziel konfigurieren.

So konfigurieren Sie ein Runbook als Ziel einer Ereignisregel EventBridge

1. Installieren und konfigurieren Sie das AWS CLI oder das AWS -Tools für PowerShell, falls Sie das noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.
2. Erstellen Sie einen Befehl, um eine neue EventBridge Ereignisregel anzugeben. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   Auslöser nach Zeitplan

Linux & macOS

```
aws events put-rule \
--name "rule name" \
--schedule-expression "cron or rate expression"
```

Windows

```
aws events put-rule ^
--name "rule name" ^
--schedule-expression "cron or rate expression"
```

PowerShell

```
Write-CWERule `
-Name "rule name" `
-ScheduleExpression "cron or rate expression"
```

Im folgenden Beispiel wird eine EventBridge Ereignisregel erstellt, die jeden Tag um 9:00 Uhr (UTC) beginnt.

Linux & macOS

```
aws events put-rule \
--name "DailyAutomationRule" \
--schedule-expression "cron(0 9 * * ? *)"
```

Windows

```
aws events put-rule ^
--name "DailyAutomationRule" ^
--schedule-expression "cron(0 9 * * ? *)"
```

PowerShell

```
Write-CWERule `
-Name "DailyAutomationRule" `
-ScheduleExpression "cron(0 9 * * ? *)"
```

Auslöser basierend auf einem Ereignis

Linux & macOS

```
aws events put-rule \
--name "rule name" \
--event-pattern "{\"source\":[\"aws.service\"],\"detail-type\":[\"service event
 detail type\"]}"
```

Windows

```
aws events put-rule ^
--name "rule name" ^
--event-pattern "{\"source\":[\"aws.service\"],\"detail-type\":[\"service event
 detail type\"]}"
```

PowerShell

```
Write-CWERule `
-Name "rule name" `
-EventPattern '{"source":["aws.service"],"detail-type":["service event detail
 type"]}'
```

Im folgenden Beispiel wird eine EventBridge Ereignisregel erstellt, die startet, wenn sich der Status einer EC2 Instanz in der Region ändert.

Linux & macOS

```
aws events put-rule \
--name "EC2InstanceStateChanges" \
--event-pattern "{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance
 State-change Notification\"]}"
```

Windows

```
aws events put-rule ^
--name "EC2InstanceStateChanges" ^
```

```
--event-pattern "{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance
 State-change Notification\"]}"
```

PowerShell

```
Write-CWERule `
-Name "EC2InstanceStateChanges" `
-EventPattern '{"source":["aws.ec2"],"detail-type":["EC2 Instance State-change
 Notification"]}'
```

Der Befehl gibt Details für die neue EventBridge Regel zurück, die der folgenden ähneln.

Linux & macOS

```
{
"RuleArn": "arn:aws:events:us-east-1:123456789012:rule/automationrule"
}
```

Windows

```
{
"RuleArn": "arn:aws:events:us-east-1:123456789012:rule/automationrule"
}
```

PowerShell

```
arn:aws:events:us-east-1:123456789012:rule/EC2InstanceStateChanges
```

3. Erstellen Sie einen Befehl, um ein Runbook als Ziel der in Schritt 2 erstellten EventBridge Ereignisregel anzugeben. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   Linux & macOS

   ```
   aws events put-targets \
   --rule rule name \
   --targets '{"Arn": " arn:aws:ssm:region:account ID:automation-definition/runbook
    name","Input":"{\"Message\":[\"{\\\"Key\\\":\\\"key name\\\",\\\"Values\\\":[\\
   ```

```
\"value\\\"]}\"]}","Id": "target ID","RoleArn": "arn:aws:iam::123456789012:role/
service-role/EventBridge service role"}'
```

Windows

```
aws events put-targets ^
--rule rule name ^
--targets '{"Arn": "arn:aws:ssm:region:account ID:automation-definition/runbook
 name","Input":"{\"Message\":[\"{\\\"Key\\\":\\\"key name\\\",\\\"Values\\\":[\\
\"value\\\"]}\"]}","Id": "target ID","RoleArn": "arn:aws:iam::123456789012:role/
service-role/EventBridge service role"}'
```

PowerShell

```
$Target = New-Object Amazon.CloudWatchEvents.Model.Target
$Target.Id = "target ID"
$Target.Arn = "arn:aws:ssm:region:account ID:automation-definition/runbook name"
$Target.RoleArn = "arn:aws:iam::123456789012:role/service-role/EventBridge
 service role"
$Target.Input = '{"input parameter":["value"],"AutomationAssumeRole":
["arn:aws:iam::123456789012:role/AutomationServiceRole"]}'

Write-CWETarget `
-Rule "rule name" `
-Target $Target
```

Im folgenden Beispiel wird ein EventBridge Ereignisziel erstellt, das die angegebene Instanz-ID mithilfe des `AWS-StartEC2Instance` Runbooks startet.

Linux & macOS

```
aws events put-targets \
--rule DailyAutomationRule \
--targets '{"Arn": "arn:aws:ssm:region:*:automation-definition/AWS-
StartEC2Instance","Input":"{\"InstanceId\":[\"i-02573cafcfEXAMPLE\"],
\"AutomationAssumeRole\":[\"arn:aws:iam::123456789012:role/AutomationServiceRole
\"]}","Id": "Target1","RoleArn": "arn:aws:iam::123456789012:role/service-role/
AWS_Events_Invoke_Start_Automation_Execution_1213609520"}'
```

Windows

```
aws events put-targets ^
--rule DailyAutomationRule ^
--targets '{"Arn": "arn:aws:ssm:region:*:automation-definition/AWS-
StartEC2Instance","Input":"{\"InstanceId\":[\"i-02573cafcfEXAMPLE\"],
\"AutomationAssumeRole\":[\"arn:aws:iam::123456789012:role/AutomationServiceRole
\"]}","Id": "Target1","RoleArn": "arn:aws:iam::123456789012:role/service-role/
AWS_Events_Invoke_Start_Automation_Execution_1213609520"}'
```

PowerShell

```
$Target = New-Object Amazon.CloudWatchEvents.Model.Target
$Target.Id = "Target1"
$Target.Arn = "arn:aws:ssm:region:*:automation-definition/AWS-StartEC2Instance"
$Target.RoleArn = "arn:aws:iam::123456789012:role/service-role/
AWS_Events_Invoke_Start_Automation_Execution_1213609520"
$Target.Input = '{"InstanceId":["i-02573cafcfEXAMPLE"],"AutomationAssumeRole":
["arn:aws:iam::123456789012:role/AutomationServiceRole"]}'

Write-CWETarget `
-Rule "DailyAutomationRule" `
-Target $Target
```

Das System gibt unter anderem folgende Informationen zurück

Linux & macOS

```
{
"FailedEntries": [],
"FailedEntryCount": 0
}
```

Windows

```
{
"FailedEntries": [],
"FailedEntryCount": 0
}
```

PowerShell

Es erfolgt keine Ausgabe, wenn der Befehl für erfolgreich ist. PowerShell

Ausführen einer Automatisierung Schritt für Schritt

Die folgenden Verfahren beschreiben, wie Sie die AWS Systems Manager Konsole verwenden und AWS Command Line Interface (AWS CLI) eine Automatisierung im manuellen Ausführungsmodus ausführen. Im manuellen Ausführungsmodus startet die Automatisierung in einem Wartestatus und verharrt zwischen den einzelnen Schritten im Wartestatus. So können Sie steuern, wann der Automatisierung fortgesetzt wird. Dies ist hilfreich, wenn Sie das Ergebnis eines Schritts überprüfen müssen, bevor Sie fortfahren.

Die Automatisierung wird im Kontext des aktuellen Benutzers ausgeführt. Das bedeutet, dass Sie keine zusätzlichen IAM-Berechtigungen konfigurieren müssen, solange Sie über die Berechtigung zum Ausführen des Runbooks verfügen und alle Aktionen von dem Runbook aufgerufen werden. Wenn Sie über Administrator-Berechtigungen in IAM verfügen, haben Sie bereits die Berechtigung zum Ausführen dieser Automatisierung.

Ausführen einer Automatisierung Schritt für Schritt (Konsole)

Das folgende Verfahren zeigt, wie Sie mithilfe der Systems Manager-Konsole eine Automatisierung Schritt für Schritt manuell ausführen.

So führen Sie einen Automatisierung Schritt für Schritt aus

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Automatisierung und Automatisierung ausführen aus.
3. Wählen Sie in der Liste Automation-Dokument ein Runbook. Wählen Sie eine oder mehrere Optionen im Bereich Dokumentkategorien, um SSM-Dokumente nach ihrem Zweck zu filtern. Um ein Runbook anzuzeigen, das Sie besitzen, wählen Sie die Im Besitz von mir-Registerkarte. Um ein Runbook anzuzeigen, das für Ihr Konto freigegeben ist, wählen Sie die Registerkarte Mit mir geteilt. Um alle Runbooks anzuzeigen, wählen Sie die Registerkarte Alle Dokumente.

> **Note**
>
> Sie können Informationen zu einem Runbook einsehen, indem Sie den Runbook-Namen auswählen.

4. Überprüfen Sie im Abschnitt Dokument-Details, ob Dokumentversion auf die Version gesetzt ist, die Sie ausführen möchten. Das System bietet die folgenden Versionsoptionen:

   - Standardversion zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird und eine neue Standardversion zugewiesen ist.
   - Letzte Version zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird, und Sie die Version auszuführen möchten, die zuletzt aktualisiert wurde.
   - 1 (Standard) – Wählen Sie diese Option zur Ausführung der ersten Version des Dokuments, welches der Standard ist.

5. Wählen Sie Weiter aus.
6. Wählen Sie im Abschnitt Execution mode (Ausführungsmodus) die Option Manual execution (Manuelle Ausführung) aus.
7. Geben Sie im Abschnitt Eingabeparameter die erforderlichen Eingaben an. Optional können Sie eine IAM-Dienstrolle aus der AutomationAssumeRoleListe auswählen.
8. Wählen Sie Ausführen.
9. Wählen Sie Execute this step (Diesen Schritt ausführen) aus, wenn Sie zum ersten Schritt der Automatisierung bereit sind. Die Automatisierung fährt mit Schritt 1 fort und hält an, bevor die weiteren Schritte des Runbooks, das Sie in Schritt 3 dieses Verfahrens ausgewählt haben, ausgeführt werden. Wenn das Runbook mehrere Schritte umfasst, müssen Sie für jeden Schritt Execute this step (Diesen Schritt ausführen) auswählen, damit die Automatisierung fortgesetzt wird. Jedes Mal, wenn Sie diesen Schritt ausführen, wird die Aktion ausgeführt.

> **Note**
>
> Die Konsole zeigt den Status der Automatisierung an. Wenn die Automatisierung einen Schritt nicht ausführen kann, finden Sie weitere Informationen unter Fehlerbehebung für Systems Manager Automation..

10. Nachdem Sie alle Schritte im Runbook abgeschlossen haben, wählen Sie Complete and view results (Abschließen und Ergebnisse anzeigen) aus, um die Automatisierung zu beenden und die Ergebnisse anzuzeigen.

Nach Abschluss einer Automatisierungsausführung können Sie die Ausführung mit denselben oder geänderten Parametern erneut ausführen. Weitere Informationen finden Sie unter Wiederholtes Ausführen von Automatisierungsausführungen.

Ausführen einer Automatisierung Schritt für Schritt (Befehlszeile)

Das folgende Verfahren beschreibt Schritt für Schritt, wie Sie die AWS CLI (unter LinuxmacOS, oder AWS -Tools für PowerShell Windows) verwenden oder manuell ausführen.

So führen Sie einen Automatisierung Schritt für Schritt aus

1. Installieren und konfigurieren Sie das AWS CLI oder das AWS -Tools für PowerShell, falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.
2. Führen Sie den folgenden Befehl aus, um eine manuelle Automatisierung zu starten. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   Linux & macOS

   ```
   aws ssm start-automation-execution \
       --document-name runbook name \
       --mode Interactive \
       --parameters runbook parameters
   ```

   Windows

   ```
   aws ssm start-automation-execution ^
       --document-name runbook name ^
       --mode Interactive ^
       --parameters runbook parameters
   ```

   PowerShell

   ```
   Start-SSMAutomationExecution `
   ```

```
    -DocumentName runbook name `
    -Mode Interactive `
    -Parameter runbook parameters
```

Hier ist ein Beispiel, bei dem das Runbook verwendet wird`AWS-RestartEC2Instance`, um die angegebene EC2 Instanz neu zu starten.

Linux & macOS

```
aws ssm start-automation-execution \
    --document-name "AWS-RestartEC2Instance" \
    --mode Interactive \
    --parameters "InstanceId=i-02573cafcfEXAMPLE"
```

Windows

```
aws ssm start-automation-execution ^
    --document-name "AWS-RestartEC2Instance" ^
    --mode Interactive ^
    --parameters "InstanceId=i-02573cafcfEXAMPLE"
```

PowerShell

```
Start-SSMAutomationExecution `
    -DocumentName AWS-RestartEC2Instance `
    -Mode Interactive
    -Parameter @{"InstanceId"="i-02573cafcfEXAMPLE"}
```

Das System gibt unter anderem folgende Informationen zurück

Linux & macOS

```
{
    "AutomationExecutionId": "ba9cd881-1b36-4d31-a698-0123456789ab"
}
```

Windows

```
{
```

```
    "AutomationExecutionId": "ba9cd881-1b36-4d31-a698-0123456789ab"
}
```

PowerShell

```
ba9cd881-1b36-4d31-a698-0123456789ab
```

3. Führen Sie den folgenden Befehl aus, wenn Sie bereit sind, den ersten Schritt der Automatisierung zu starten. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen. Die Automatisierung fährt mit Schritt 1 fort und hält an, bevor die weiteren Schritte des Runbooks, das Sie in Schritt 1 dieses Verfahrens ausgewählt haben, ausgeführt werden. Wenn das Runbook mehrere Schritte umfasst, müssen Sie den folgenden Befehl für jeden Schritt ausführen, damit die Automatisierung fortfahren kann.

   Linux & macOS

   ```
   aws ssm send-automation-signal \
       --automation-execution-id ba9cd881-1b36-4d31-a698-0123456789ab \
       --signal-type StartStep \
       --payload StepName="stopInstances"
   ```

   Windows

   ```
   aws ssm send-automation-signal ^
       --automation-execution-id ba9cd881-1b36-4d31-a698-0123456789ab ^
       --signal-type StartStep ^
       --payload StepName="stopInstances"
   ```

   PowerShell

   ```
   Send-SSMAutomationSignal `
       -AutomationExecutionId ba9cd881-1b36-4d31-a698-0123456789ab `
       -SignalType StartStep
       -Payload @{"StepName"="stopInstances"}
   ```

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

4. Führen Sie den folgenden Befehl aus, um den Status jeder Schrittausführung in der Automatisierung abzurufen.

Linux & macOS

```
aws ssm describe-automation-step-executions \
    --automation-execution-id ba9cd881-1b36-4d31-a698-0123456789ab
```

Windows

```
aws ssm describe-automation-step-executions ^
    --automation-execution-id ba9cd881-1b36-4d31-a698-0123456789ab
```

PowerShell

```
Get-SSMAutomationStepExecution `
    -AutomationExecutionId ba9cd881-1b36-4d31-a698-0123456789ab
```

Das System gibt unter anderem folgende Informationen zurück

Linux & macOS

```
{
    "StepExecutions": [
        {
            "StepName": "stopInstances",
            "Action": "aws:changeInstanceState",
            "ExecutionStartTime": 1557167178.42,
            "ExecutionEndTime": 1557167220.617,
            "StepStatus": "Success",
            "Inputs": {
                "DesiredState": "\"stopped\"",
                "InstanceIds": "[\"i-02573cafcfEXAMPLE\"]"
            },
            "Outputs": {
                "InstanceStates": [
                    "stopped"
                ]
            },
            "StepExecutionId": "654243ba-71e3-4771-b04f-0123456789ab",
            "OverriddenParameters": {},
            "ValidNextSteps": [
                "startInstances"
```

```
            ]
        },
        {
            "StepName": "startInstances",
            "Action": "aws:changeInstanceState",
            "ExecutionStartTime": 1557167273.754,
            "ExecutionEndTime": 1557167480.73,
            "StepStatus": "Success",
            "Inputs": {
                "DesiredState": "\"running\"",
                "InstanceIds": "[\"i-02573cafcfEXAMPLE\"]"
            },
            "Outputs": {
                "InstanceStates": [
                    "running"
                ]
            },
            "StepExecutionId": "8a4a1e0d-dc3e-4039-a599-0123456789ab",
            "OverriddenParameters": {}
        }
    ]
}
```

Windows

```
{
    "StepExecutions": [
        {
            "StepName": "stopInstances",
            "Action": "aws:changeInstanceState",
            "ExecutionStartTime": 1557167178.42,
            "ExecutionEndTime": 1557167220.617,
            "StepStatus": "Success",
            "Inputs": {
                "DesiredState": "\"stopped\"",
                "InstanceIds": "[\"i-02573cafcfEXAMPLE\"]"
            },
            "Outputs": {
                "InstanceStates": [
                    "stopped"
                ]
            },
            "StepExecutionId": "654243ba-71e3-4771-b04f-0123456789ab",
```

```
            "OverriddenParameters": {},
            "ValidNextSteps": [
                "startInstances"
            ]
        },
        {
            "StepName": "startInstances",
            "Action": "aws:changeInstanceState",
            "ExecutionStartTime": 1557167273.754,
            "ExecutionEndTime": 1557167480.73,
            "StepStatus": "Success",
            "Inputs": {
                "DesiredState": "\"running\"",
                "InstanceIds": "[\"i-02573cafcfEXAMPLE\"]"
            },
            "Outputs": {
                "InstanceStates": [
                    "running"
                ]
            },
            "StepExecutionId": "8a4a1e0d-dc3e-4039-a599-0123456789ab",
            "OverriddenParameters": {}
        }
    ]
}
```

PowerShell

```
Action: aws:changeInstanceState
ExecutionEndTime     : 5/6/2019 19:45:46
ExecutionStartTime   : 5/6/2019 19:45:03
FailureDetails       :
FailureMessage       :
Inputs               : {[DesiredState, "stopped"], [InstanceIds,
 ["i-02573cafcfEXAMPLE"]]}
IsCritical           : False
IsEnd                : False
MaxAttempts          : 0
NextStep             :
OnFailure            :
Outputs              : {[InstanceStates,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
OverriddenParameters : {}
```

```
Response             :
ResponseCode         :
StepExecutionId      : 8fcc9641-24b7-40b3-a9be-0123456789ab
StepName             : stopInstances
StepStatus           : Success
TimeoutSeconds       : 0
ValidNextSteps       : {startInstances}
```

5. Führen Sie den folgenden Befehl aus, um die Automatisierung abzuschließen, nachdem alle im ausgewählten Runbook angegebenen Schritte abgeschlossen sind. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   Linux & macOS

   ```
   aws ssm stop-automation-execution \
       --automation-execution-id ba9cd881-1b36-4d31-a698-0123456789ab \
       --type Complete
   ```

   Windows

   ```
   aws ssm stop-automation-execution ^
       --automation-execution-id ba9cd881-1b36-4d31-a698-0123456789ab ^
       --type Complete
   ```

   PowerShell

   ```
   Stop-SSMAutomationExecution `
       -AutomationExecutionId ba9cd881-1b36-4d31-a698-0123456789ab `
       -Type Complete
   ```

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

Planen von Automatisierungen mit State Manager-Zuordnungen

Sie können eine Automatisierung starten, indem Sie eine State Manager-Verknüpfung mit einem Runbook erstellen. State Manager ist ein Tool in AWS Systems Manager. Indem Sie eine State Manager Zuordnung zu einem Runbook erstellen, können Sie verschiedene Ressourcentypen als Ziel verwenden. AWS Sie können beispielsweise Zuordnungen erstellen, die einen gewünschten Status für eine AWS Ressource erzwingen, einschließlich der folgenden:

- Ordnen Sie Amazon Elastic Compute Cloud (Amazon EC2) -Instances eine Systems Manager Manager-Rolle zu, um sie zu verwalteten Instances zu machen.
- Erzwingen Sie die gewünschten Eingangs- und Ausgangsregeln für eine Sicherheitsgruppe.
- Erstellen oder löschen Sie Amazon DynamoDB-Backups.
- Erstellen oder löschen Sie Amazon Elastic Block Store (Amazon EBS)-Snapshots.
- Deaktivieren Sie Lese- und Schreibberechtigungen für Amazon Simple Storage Service (Amazon S3)-Buckets.
- Starten, Stoppen oder starten Sie verwaltete Instances und Amazon Relational Database Service (Amazon RDS)-Instances neu.
- Anwenden von Patches auf Linux, macOS und Windows AMIs.

Gehen Sie wie folgt vor, um eine State Manager Assoziation zu erstellen, die mithilfe der AWS Systems Manager Konsole und AWS Command Line Interface (AWS CLI) eine Automatisierung ausführt. Allgemeine Informationen zu Verknüpfungen und Informationen zum Erstellen einer Zuordndung, die ein `Command`-SSM-Dokument oder `Policy` verwendet, finden Sie unter Erstellen von Zuordnungen.

Bevor Sie beginnen

Beachten Sie die folgenden wichtigen Details, bevor Sie eine Automatisierung mithilfe von State Manager ausführen:

- Bevor Sie eine Zuordnung erstellen können, die ein Runbook verwendet, stellen Sie sicher, dass Sie Berechtigungen für Automation konfiguriert haben, ein Tool in AWS Systems Manager. Weitere Informationen finden Sie unter Einrichten der Automatisierung.
- State Manager-Zuordnungen, die Runbooks verwenden, zählen zu der maximalen Anzahl der gleichzeitig ausgeführten Automatisierungen in Ihrem AWS-Konto hinzu. Sie können maximal 100 Automatisierungen gleichzeitig ausführen. Informationen finden Sie unter Systems Manager Service Quotas im Allgemeine Amazon Web Services-Referenz.
- Beim Ausführen einer Automatisierung protokolliert State Manager nicht die API-Operationen, die von der Automatisierung in AWS CloudTrail initiiert wurden.
- Systems Manager erstellt automatisch eine serviceverknüpfte Rolle, damit State Manager die Berechtigung hat, API-Vorgänge von Systems Manager Automation aufzurufen. Wenn Sie möchten, können Sie die serviceverknüpfte Rolle selbst erstellen, indem Sie den folgenden Befehl von AWS CLI oder AWS -Tools für PowerShell aus ausführen.

Linux & macOS

```
aws iam create-service-linked-role \
--aws-service-name ssm.amazonaws.com
```

Windows

```
aws iam create-service-linked-role ^
--aws-service-name ssm.amazonaws.com
```

PowerShell

```
New-IAMServiceLinkedRole `
-AWSServiceName ssm.amazonaws.com
```

Weitere Informationen zu serviceverknüpften Rollen finden Sie unter Verwenden von serviceverknüpften Rollen für Systems Manager.

Erstellen einer Zuordnung, die eine Automatisierung ausführt (Konsole)

Im folgenden Verfahren wird beschrieben, wie mithilfe der Systems Manager-Konsole eine State Manager-Zuordnung erstellt wird, die eine Automatisierung ausführt.

So erstellen Sie eine State Manager-Zuordnung zum Ausführen einer Automatisierung

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich State Manager und anschließend Zuordnung erstellen aus.
3. Geben Sie im Feld Name einen Namen an. Dies ist zwar optional, wird aber empfohlen.
4. Wählen Sie in der Liste Document ein Runbook aus. Verwenden Sie die Suchleiste, um nach allen Runbooks mit Document type : Equal : Automation zu filtern. Zur Anzeige von weiteren Runbooks verwenden Sie die Zahlen rechts neben der Suchleiste.

> **Note**
>
> Sie können Informationen zu einem Runbook einsehen, indem Sie den Runbook-Namen auswählen.

5. Wählen Sie Simple execution (Einfache Ausführung) aus, um die Automatisierung auf einem oder mehreren Zielen auszuführen, indem Sie die Ressourcen-ID für diese Ziele angeben. Wählen Sie Ratensteuerung, um die Automatisierung für eine ganze Flotte von AWS Ressourcen auszuführen, indem Sie eine Targeting-Option wie Tags oder angeben AWS - Ressourcengruppen. Sie können auch die Operation der Automatisierung auf Ihren Ressourcen steuern, indem Sie Gleichzeitigkeits- und Fehlergrenzwerte angeben.

   Wenn Sie Rate control (Ratensteuerung) auswählen, wird der Abschnitt Targets (Ziele) angezeigt.

6. Wählen Sie im Abschnitt Targets (Ziele) eine Methode zur Ausrichtung der Ressourcen aus.

   a. (Erforderlich) Wählen Sie in der Liste Parameter einen Parameter aus. Die Elemente in der Liste Parameter richten sich nach den Parametern in dem Runbook, das Sie zu Beginn dieses Verfahrens ausgewählt haben. Durch Auswahl eines Parameters legen Sie den Typ der Ressource fest, für die die Automatisierung ausgeführt wird.

   b. (Erforderlich) Wählen Sie in der Liste Targets (Ziele) ein Verfahren für die Ausrichtung auf Ressourcen aus.

      - Resource Group (Ressourcengruppe): Wählen Sie den Namen der Gruppe aus der Liste Resource Group (Ressourcengruppe) aus. Weitere Informationen zum Targeting AWS -Ressourcengruppen in Runbooks finden Sie unterTargeting von AWS - Ressourcengruppen.
      - Tags: Geben Sie den Tag-Schlüssel und (optional) den Tag-Wert in die dafür vorgesehenen Felder ein. Wählen Sie Hinzufügen aus. Weitere Informationen zum Targeting von Tags in Runbooks finden Sie unter Anzielen eines Tags.
      - Parameter Values (Parameterwerte): Geben Sie die Werte im Abschnitt Input parameters (Eingabeparameter) ein. Wenn Sie mehrere Werte angeben, führt Systems Manager eine untergeordnete Automatisierung für jeden angegebenen Wert aus.

        Nehmen Sie beispielsweise an, dass das Runbook einen InstanceID-Parameter enthält. Wenn Sie die Werte des InstanceID-Parameters beim Ausführen der Automatisierung verwenden, führt Systems Manager eine untergeordnete Automatisierung für jeden angegebenen Instance-ID-Wert aus. Die übergeordnete Automatisierung ist abgeschlossen, wenn Automatisierung die Ausführung jeder angegebenen Instance abgeschlossen hat oder wenn die Automatisierung fehlschlägt. Sie können maximal 50 Parameterwerte für die Ausrichtung verwenden. Weitere Informationen zum Targeting von Parameterwerten in Runbooks finden Sie unter Ausrichtung auf Parameterwerte.

7. Geben Sie im Abschnitt Eingabeparameter die erforderlichen Eingabeparameter an.

   Wenn Sie die Zielressourcen mithilfe von Tags oder einer Ressourcengruppe ausgewählt haben, müssen Sie möglicherweise keine der Optionen im Abschnitt Input parameters (Eingabeparameter) auswählen. Wenn Sie beispielsweise das `AWS-RestartEC2Instance` Runbook ausgewählt haben und sich dafür entschieden haben, Instances mithilfe von Tags anzusprechen, müssen Sie IDs im Abschnitt Eingabeparameter keine Instanz angeben oder auswählen. Die Automatisierung sucht die Instances für den Neustart mit den von Ihnen angegebenen Tags.

   > **Important**
   >
   > Sie müssen in dem AutomationAssumeRoleFeld einen Rollen-ARN angeben. State Managerverwendet die Rolle „Übernehmen“, um die im Runbook AWS-Services angegebene Rolle aufzurufen und Automatisierungszuordnungen in Ihrem Namen auszuführen.

8. Wählen Sie im Abschnitt Specify schedule (Zeitplan angeben) die Option On Schedule (Nach Zeitplan) aus, wenn Sie die Zuordnungen in regelmäßigen Abständen ausführen möchten. Wenn Sie diese Option auswählen, verwenden Sie die bereitgestellten Optionen zum Erstellen des Zeitplans mithilfe von Cron- oder Rate-Ausdrücken. Weitere Informationen zu Cron- und Rate-Ausdrücken für State Manager finden Sie unter Cron- und Rate-Ausdrücke für Zuordnungen.

   > **Note**
   >
   > Rate-Ausdrücke werden bevorzugt zur Planung für State Manager-Zuordnungen verwendet, die Automatisierungen verwenden ausführen. Rate-Ausdrücke ermöglichen mehr Flexibilität für die Ausführung von Zuordnungen für den Fall, dass Sie die maximale Anzahl von gleichzeitig ausgeführten Automatisierungen erreichen. Mit einem Ratenzeitplan kann Systems Manager die Automatisierung kurz nach dem Empfangen der Benachrichtigungen, dass gleichzeitige Automatisierungen das Maximum erreicht haben und gedrosselt wurden, wiederholen.

   Wählen Sie No schedule (Kein Zeitplan)aus, wenn Sie die Zuordnung einmalig ausführen möchten.

9. (Optional) Wählen Sie im Bereich Rate Control die Optionen Parallelität und Fehlerschwellenwert aus, um die Bereitstellung der Automatisierung auf Ihren AWS Ressourcen zu steuern.

   a. Wählen Sie im Abschnitt Gleichzeitigkeit eine Option aus:

      - Wählen Sie targets (Ziele) aus, um eine absolute Anzahl von Zielen einzugeben, die die Automatisierung gleichzeitig ausführen können.
      - Wählen Sie percentage (Prozentsatz) aus, um einen Prozentsatz der Ziele anzugeben, die den Automatisierung gleichzeitig ausführen können.

   b. Wählen Sie im Abschnitt Fehlerschwellenwert eine Option aus:

      - Wählen Sie errors (Fehler), um eine absolute Anzahl von zulässigen Fehlern anzugeben, bevor die Automation damit aufhört, die Automatisierung an andere Ressourcen zu senden.
      - Wählen Sie percentage (Prozentsatz) aus, um einen Prozentsatz von zulässigen Fehlern anzugeben, bevor Automation damit aufhört, die Automatisierung an andere Ressourcen zu senden.

   Weitere Informationen zur Verwendung von Zielen und Ratensteuerungen mit Automation finden Sie unter Automatisierte Abläufe in großem Umfang ausführen.

10. Wählen Sie Zuordnung erstellen.

> **Important**
>
> Wenn Sie eine Zuordnung erstellen, wird die Zuordnung sofort für die ausgewählten Ziele ausgeführt. Die Zuordnung wird anschließend auf Grundlage des ausgewählten Cron- oder Rate-Ausdrucks ausgeführt. Wenn Sie No schedule (Kein Zeitplan) ausgewählt haben, wird die Zuordnung nicht mehr ausgeführt.

Erstellen einer Zuordnung, die eine Automatisierung ausführt (Befehlszeile)

Das folgende Verfahren beschreibt, wie Sie die AWS CLI (unter Linux oderWindows Server) verwenden oder AWS -Tools für PowerShell eine State Manager Zuordnung erstellen, die eine Automatisierung ausführt.

Bevor Sie beginnen

Bevor Sie das folgende Verfahren ausführen, stellen Sie sicher, dass Sie eine IAM-Servicerolle erstellt haben, die die zum Ausführen des Runbooks erforderlichen Berechtigungen enthält, und eine Vertrauensstellung für Automation konfiguriert haben, ein Tool in AWS Systems Manager. Weitere Informationen finden Sie unter Aufgabe 1: Erstellen einer Servicerolle für Automation.

So erstellen Sie eine Zuordnung zum Ausführen einer Automatisierung

1. Installieren und konfigurieren Sie das AWS CLI oder das AWS -Tools für PowerShell, falls Sie das noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.
2. Nutzen Sie den folgenden Befehl, um eine Liste der Dokumente anzuzeigen.

   Linux & macOS

   ```
   aws ssm list-documents
   ```

   Windows

   ```
   aws ssm list-documents
   ```

   PowerShell

   ```
   Get-SSMDocumentList
   ```

   Notieren Sie den Namen des Runbooks, das Sie für die Zuordnung verwenden möchten.
3. Führen Sie den folgenden Befehl aus, um Details des Runbooks einsehen zu können: Ersetzen Sie den Befehl im folgenden Befehl *runbook name* durch Ihre eigenen Informationen.

   Linux & macOS

   ```
   aws ssm describe-document \
   --name runbook name
   ```

   Notieren Sie einen Parameternamen (z. B. `InstanceId`), den Sie für die Option `--automation-target-parameter-name` verwenden möchten. Dieser Parameter bestimmt den Typ der Ressource, für die die Automatisierung ausgeführt wird.

Windows

```
aws ssm describe-document ^
--name runbook name
```

Notieren Sie einen Parameternamen (z. B. `InstanceId`), den Sie für die Option `--automation-target-parameter-name` verwenden möchten. Dieser Parameter bestimmt den Typ der Ressource, für die die Automatisierung ausgeführt wird.

PowerShell

```
Get-SSMDocumentDescription `
-Name runbook name
```

Notieren Sie einen Parameternamen (z. B. `InstanceId`), den Sie für die Option `AutomationTargetParameterName` verwenden möchten. Dieser Parameter bestimmt den Typ der Ressource, für die die Automatisierung ausgeführt wird.

4. Erstellen Sie einen Befehl, der eine Automatisierung mithilfe einer State Manager-Zuordnung ausführt. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   Ausrichtung mithilfe von Tags

   Linux & macOS

   ```
   aws ssm create-association \
   --association-name association name \
   --targets Key=tag:key name,Values=value \
   --name runbook name \
   --parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/
   RunbookAssumeRole \
   --automation-target-parameter-name target parameter \
   --schedule "cron or rate expression"
   ```

   > **Note**
   >
   > Wenn Sie eine Assoziation mithilfe von erstellen AWS CLI, verwenden Sie den `--targets` Parameter, um Instances als Ziel für die Zuordnung festzulegen.

Verwenden Sie nicht den Parameter `--instance-id`. Der Parameter `--instance-id` ist veraltet.

Windows

```
aws ssm create-association ^
--association-name association name ^
--targets Key=tag:key name,Values=value ^
--name runbook name ^
--parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/
RunbookAssumeRole ^
--automation-target-parameter-name target parameter ^
--schedule "cron or rate expression"
```

Note

Wenn Sie eine Assoziation mithilfe von erstellen AWS CLI, verwenden Sie den `--targets` Parameter, um Instances als Ziel für die Zuordnung festzulegen. Verwenden Sie nicht den Parameter `--instance-id`. Der Parameter `--instance-id` ist veraltet.

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
$Targets.Key = "tag:key name"
$Targets.Values = "value"

New-SSMAssociation `
-AssociationName "association name" `
-Target $Targets `
-Name "runbook name" `
-Parameters @{
"AutomationAssumeRole"="arn:aws:iam::123456789012:role/RunbookAssumeRole" } `
-AutomationTargetParameterName "target parameter" `
-ScheduleExpression "cron or rate expression"
```

> **Note**
>
> Wenn Sie eine Assoziation mithilfe von erstellen AWS -Tools für PowerShell, verwenden Sie den `Target` Parameter, um Instances als Ziel für die Zuordnung festzulegen. Verwenden Sie nicht den Parameter `InstanceId`. Der Parameter `InstanceId` ist veraltet.

Ausrichtung mithilfe von Parameterwerten

Linux & macOS

```
aws ssm create-association \
--association-name association name \
--targets Key=ParameterValues,Values=value,value 2,value 3 \
--name runbook name \
--parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/
RunbookAssumeRole \
--automation-target-parameter-name target parameter \
--schedule "cron or rate expression"
```

Windows

```
aws ssm create-association ^
--association-name association name ^
--targets Key=ParameterValues,Values=value,value 2,value 3 ^
--name runbook name ^
--parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/
RunbookAssumeRole ^
--automation-target-parameter-name target parameter ^
--schedule "cron or rate expression"
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
$Targets.Key = "ParameterValues"
$Targets.Values = "value","value 2","value 3"

New-SSMAssociation `
-AssociationName "association name" `
```

```
-Target $Targets `
-Name "runbook name" `
-Parameters @{
"AutomationAssumeRole"="arn:aws:iam::123456789012:role/RunbookAssumeRole"} `
-AutomationTargetParameterName "target parameter" `
-ScheduleExpression "cron or rate expression"
```

Targeting mit AWS -Ressourcengruppen

Linux & macOS

```
aws ssm create-association \
--association-name association name \
--targets Key=ResourceGroup,Values=resource group name \
--name runbook name \
--parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/
RunbookAssumeRole \
--automation-target-parameter-name target parameter \
--schedule "cron or rate expression"
```

Windows

```
aws ssm create-association ^
--association-name association name ^
--targets Key=ResourceGroup,Values=resource group name ^
--name runbook name ^
--parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/
RunbookAssumeRole ^
--automation-target-parameter-name target parameter ^
--schedule "cron or rate expression"
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
$Targets.Key = "ResourceGroup"
$Targets.Values = "resource group name"

New-SSMAssociation `
-AssociationName "association name" `
-Target $Targets `
-Name "runbook name" `
```

```
-Parameters @{
"AutomationAssumeRole"="arn:aws:iam::123456789012:role/RunbookAssumeRole"} `
-AutomationTargetParameterName "target parameter" `
-ScheduleExpression "cron or rate expression"
```

Targeting mehrerer Konten und Regionen

Linux & macOS

```
aws ssm create-association \
--association-name association name \
--targets Key=ResourceGroup,Values=resource group name \
--name runbook name \
--parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/
RunbookAssumeRole \
--automation-target-parameter-name target parameter \
--schedule "cron or rate expression" \
--target-locations
 Accounts=111122223333,444455556666,444455556666,Regions=region,region
```

Windows

```
aws ssm create-association ^
--association-name association name ^
--targets Key=ResourceGroup,Values=resource group name ^
--name runbook name ^
--parameters AutomationAssumeRole=arn:aws:iam::123456789012:role/
RunbookAssumeRole ^
--automation-target-parameter-name target parameter ^
--schedule "cron or rate expression" ^
--target-locations
 Accounts=111122223333,444455556666,444455556666,Regions=region,region
```

PowerShell

```
$Targets = New-Object Amazon.SimpleSystemsManagement.Model.Target
$Targets.Key = "ResourceGroup"
$Targets.Values = "resource group name"

New-SSMAssociation `
-AssociationName "association name" `
```

```
-Target $Targets `
-Name "runbook name" `
-Parameters @{
"AutomationAssumeRole"="arn:aws:iam::123456789012:role/RunbookAssumeRole"} `
-AutomationTargetParameterName "target parameter" `
-ScheduleExpression "cron or rate expression" `
-TargetLocations @{
    "Accounts"=["111122223333,444455556666,444455556666"],
    "Regions"=["region,region"]
```

Der Befehl gibt Details für die neue Zuordnung zurück, die den folgenden ähneln.

Linux & macOS

```
{
"AssociationDescription": {
    "ScheduleExpression": "cron(0 7 ? * MON *)",
    "Name": "AWS-StartEC2Instance",
    "Parameters": {
        "AutomationAssumeRole": [
            "arn:aws:iam::123456789012:role/RunbookAssumeRole"
        ]
    },
    "Overview": {
        "Status": "Pending",
        "DetailedStatus": "Creating"
    },
    "AssociationId": "1450b4b7-bea2-4e4b-b340-01234EXAMPLE",
    "DocumentVersion": "$DEFAULT",
    "AutomationTargetParameterName": "InstanceId",
    "LastUpdateAssociationDate": 1564686638.498,
    "Date": 1564686638.498,
    "AssociationVersion": "1",
    "AssociationName": "CLI",
    "Targets": [
        {
            "Values": [
                "DEV"
            ],
            "Key": "tag:ENV"
        }
    ]
```

```
}
}
```

Windows

```
{
"AssociationDescription": {
    "ScheduleExpression": "cron(0 7 ? * MON *)",
    "Name": "AWS-StartEC2Instance",
    "Parameters": {
        "AutomationAssumeRole": [
            "arn:aws:iam::123456789012:role/RunbookAssumeRole"
        ]
    },
    "Overview": {
        "Status": "Pending",
        "DetailedStatus": "Creating"
    },
    "AssociationId": "1450b4b7-bea2-4e4b-b340-01234EXAMPLE",
    "DocumentVersion": "$DEFAULT",
    "AutomationTargetParameterName": "InstanceId",
    "LastUpdateAssociationDate": 1564686638.498,
    "Date": 1564686638.498,
    "AssociationVersion": "1",
    "AssociationName": "CLI",
    "Targets": [
        {
            "Values": [
                "DEV"
            ],
            "Key": "tag:ENV"
        }
    ]
}
}
```

PowerShell

```
Name                 : AWS-StartEC2Instance
InstanceId           :
Date                 : 8/1/2019 7:31:38 PM
Status.Name          :
Status.Date          :
```

```
Status.Message        :
Status.AdditionalInfo :
```

> **Note**
>
> Wenn Sie auf einer oder mehreren Instances eine Zuordnung anhand von Tags erstellen und von einer dieser Instances die Tags entfernen, wird die Zuordnung auf dieser Instance nicht mehr ausgeführt. Die Zuordnung zwischen der Instance und dem State Manager-Dokument ist aufgehoben.

Fehlerbehebung bei Automatisierungen, die von State Manager-Zuordnungen ausgeführt werden

Systems Manager setzt ein Limit von 100 gleichzeitigen Automatisierungen und 1 000 Automatisierungen in der Warteschlange pro Konto und Region. Wenn eine State Manager Assoziation, die ein Runbook verwendet, den Status Fehlgeschlagen und den detaillierten Status anzeigt AutomationExecutionLimitExceeded, hat Ihre Automatisierung möglicherweise das Limit erreicht. Daher drosselt Systems Manager die Automatisierungen. Führen Sie folgende Schritte aus, um dieses Problem zu lösen:

- Verwenden Sie einen anderen Rate- oder Cron-Ausdruck für Ihre Zuordnung. Beispiel: Wenn die Zuordnung alle 30 Minuten ausgeführt werden soll, ändern Sie den Ausdruck so, dass er jede Stunde oder alle zwei Stunden ausgeführt wird.
- Löschen Sie vorhandene Automatisierungen mit dem Status Pending (Ausstehend). Durch Löschen dieser Automatisierungen bereinigen Sie die aktuelle Warteschlange.

Planen von Automatisierungen mit Wartungsfenstern

Sie starten eine Automatisierung, indem Sie ein Runbook als registrierte Aufgabe für ein Wartungsfenster konfigurieren. Durch die Registrierung des Runbooks als registrierte Aufgabe führt ein Wartungsfenster die Automatisierung während des geplanten Wartungszeitraums aus.

Angenommen, Sie erstellen beispielsweise ein Runbook mit dem Namen `CreateAMI`, welches eine Amazon Machine Image (AMI) von Instances erstellt, welche als Ziele für das Wartungsfenster registriert sind. Um ein `CreateAMI`-Runbook (und die entsprechende Automatisierung) als eine registrierte Aufgabe eines Wartungsfensters angeben zu können, müssen Sie zunächst ein Wartungsfenster erstellen und Ziele registrieren. Im Anschluss daran geben Sie mit den folgenden

Schritten das Dokument `CreateAMI` als registrierte Aufgabe innerhalb des Wartungsfensters an. Wenn das Wartungsfenster während des geplanten Zeitraums gestartet wird, führt das System die Automatisierung aus und erstellt ein der AMI registrierten Ziele.

Weitere Informationen zum Erstellen eines Automation-Runbooks finden Sie unter Erstellen Ihrer eigenen Runbooks. Automation ist ein Tool in AWS Systems Manager.

Gehen Sie wie folgt vor, um eine Automatisierung mithilfe der AWS Systems Manager Konsole, AWS Command Line Interface (AWS CLI) oder als registrierte Aufgabe für ein Wartungsfenster zu konfigurieren AWS Tools for Windows PowerShell.

Registrieren einer Automatisierungsaufgabe für ein Wartungsfenster (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie mithilfe der Systems Manager-Konsole eine Automatisierung als registrierte Aufgabe für ein Wartungsfenster konfigurieren.

Bevor Sie beginnen

Bevor Sie die folgenden Schritte ausführen, müssen Sie ein Wartungsfenster erstellen und mindestens ein Ziel registrieren. Weitere Informationen finden Sie in den folgenden Verfahren:

- Erstellen eines Wartungsfensters mit der Konsole.
- Ziele zu einem Wartungsfenster mit der Konsole zuweisen

So konfigurieren Sie einen Automatisierung als registrierte Aufgabe für ein Wartungsfenster

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im linken Navigationsbereich Maintenance Windows und dann das Wartungsfenster aus, für das Sie eine Automation-Aufgabe registrieren möchten.
3. Wählen Sie Aktionen. Wählen Sie dann Register Automation task (Automation-Aufgabe registrieren) aus, um die gewünschte Automatisierung mithilfe eines Runbooks auf den Zielen auszuführen.
4. Geben Sie unter Name einen Namen für die Aufgabe ein.
5. Geben Sie im Feld Description (Beschreibung) eine Beschreibung ein.
6. Wählen Sie für Document (Dokument) das Runbook aus, das die auszuführende Aufgabe definiert.
7. Wählen Sie für Document version (Dokumentversion) die zu verwendende Runbook-Version aus.

8. Wählen Sie für Task priority (Aufgabenpriorität) eine Priorität für diese Aufgabe aus. 1 ist die höchste Priorität. Aufgaben in einem Wartungsfenster werden in Reihenfolge der Priorität geplant. Dabei werden Aufgaben mit derselben Priorität parallel ausgeführt.
9. Geben Sie im Abschnitt Targets (Ziele) die Ziele an, auf denen Sie diesen Automation-Workflow ausführen möchten, wenn das von Ihnen gewählte Runbook eines ist, das Aufgaben auf Ressourcen aufführt. Hierzu können Sie entweder Tags angeben oder die Instances manuell auswählen.

   > **Note**
   >
   > Wenn Sie die Ressourcen über Eingabeparameter anstelle von Zielen übergeben möchten, müssen Sie kein Wartungsfensterziel angeben.
   > In vielen Fällen müssen Sie kein Ziel für eine Automation-Aufgabe explizit angeben. Angenommen, Sie erstellen beispielsweise eine Automation-Aufgabe, um eine Amazon Machine Image (AMI) für Linux mit dem `AWS-UpdateLinuxAmi`-Runbook zu aktualisieren. Wenn die Aufgabe ausgeführt wird, wird AMI mit den neuesten verfügbaren Linux-Verteilungspaketen und Amazon-Software aktualisiert. Neue Instances, die aus der AMI erstellt wurden, haben diese Updates bereits installiert. Da die ID des AMI in den Eingabeparametern für das Runbook angegeben ist, muss in der Wartungsfenster-Aufgabe kein Ziel erneut angegeben werden.

   Informationen zu Wartungsfenster-Tasks, für die keine Ziele erforderlich sind, finden Sie unter the section called “Wartungsfenster-Tasks ohne Ziele registrieren”.
10. (Optional) Für Rate control (Ratenregelung):

    > **Note**
    >
    > Wenn die ausgeführte Aufgabe keine Ziele angibt, müssen Sie keine Ratensteuerungen angeben.

    - Geben Sie für Concurrency (Gleichzeitigkeit) entweder eine Anzahl oder einen Prozentsatz der Ziele ein, auf denen die Automatisierung gleichzeitig ausgeführt wird.

      Wenn Sie Ziele anhand von Tag-Schlüssel-Wert-Paaren ausgewählt haben und nicht sicher sind, von wie vielen Zielen die ausgewählten Tags verwendet werden, sollten Sie die Anzahl

der Automatisierungen, die gleichzeitig ausgeführt werden können, durch einen Prozentsatz begrenzen.

Wenn das Wartungsfenster ausgeführt wird, wird pro Ziel eine neue Automatisierung eingeleitet. Es gilt ein Limit von 100 gleichzeitigen Automationen pro AWS-Konto. Wenn Sie einen Gleichzeitigkeitswert über 100 angeben, werden alle gleichzeitigen Automatisierungen über die 100. hinaus automatisch zur Automatisierungswarteschlange hinzugefügt. Informationen finden Sie unter Systems Manager Service Quotas im Allgemeine Amazon Web Services-Referenz.

- Geben Sie unter Error threshold (Fehlerschwellenwert) an, wann die Ausführung der Automatisierung auf anderen Zielen beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Zielen ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, führt Systems Manager keine Automatisierungen mehr aus, wenn der vierte Fehler empfangen wird. Von Zielen, auf denen die Automatisierung noch ausgeführt wird, werden unter Umständen ebenfalls Fehler gesendet.

11. Geben Sie im Abschnitt Input Parameters die Parameter für das Runbook an. Bei Runbooks werden die Werte vom System automatisch gefüllt. Sie können diese Werte beibehalten oder ersetzen.

> **Important**
>
> Für Runbooks können Sie optional eine Automatisierungsübernahmerolle angeben. Wenn Sie keine Rolle für diesen Parameter angeben, übernimmt die Automatisierung die Wartungsfenster-Servicerolle, die Sie in Schritt 11 gewählt haben. Daher müssen Sie sicherstellen, dass die von Ihnen gewählte Wartungsfenster-Servicerolle über die entsprechenden AWS Identity and Access Management (IAM-) Berechtigungen verfügt, um die im Runbook definierten Aktionen auszuführen.
> Beispiel: Die serviceverknüpfte Rolle für Systems Manager verfügt nicht über die IAM-Berechtigung `ec2:CreateSnapshot`, die zur Verwendung des Runbooks `AWS-CopySnapshot` benötigt wird. Hier müssen Sie entweder eine benutzerdefinierte Wartungsfenster-Servicerolle verwenden oder eine Automation-Übernahmerolle angeben, die über `ec2:CreateSnapshot`-Berechtigungen verfügt. Weitere Informationen finden Sie unter Einrichten der Automatisierung.

12. Wählen Sie im Bereich IAM service role (IAM-Servicerolle) eine Rolle aus, um Systems Manager Berechtigungen zum Starten der Automatisierung zu erteilen.

Informationen zum Erstellen einer Servicerolle für Wartungsfenster-Aufgaben finden Sie unter Einrichten von Maintenance Windows.

13. Wählen Sie Register Automation task (Automation-Aufgabe registrieren) aus.

Registrieren einer Automation-Aufgabe für ein Wartungsfenster (Befehlszeile)

Das folgende Verfahren beschreibt, wie Sie die AWS CLI (unter Linux oderWindows Server) verwenden oder AWS -Tools für PowerShell eine Automatisierung als registrierte Aufgabe für ein Wartungsfenster konfigurieren.

Bevor Sie beginnen

Bevor Sie die folgenden Schritte ausführen, müssen Sie ein Wartungsfenster erstellen und mindestens ein Ziel registrieren. Weitere Informationen finden Sie in den folgenden Verfahren:

- Schritt 1: Erstellen Sie das Wartungsfenster mit dem AWS CLI.
- Schritt 2: Ein Zielknoten mit dem Wartungsfenster mithilfe von AWS CLI registrieren

So konfigurieren Sie einen Automatisierung als registrierte Aufgabe für ein Wartungsfenster

1. Installieren und konfigurieren Sie das AWS CLI oder das AWS -Tools für PowerShell, falls Sie das noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.

2. Erstellen Sie einen Befehl, um eine Automatisierung als registrierte Aufgabe für ein Wartungsfenster zu konfigurieren. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   Linux & macOS

```
aws ssm register-task-with-maintenance-window \
--window-id window ID \
--name task name \
--task-arn runbook name \
--targets Key=targets,Values=value \
--service-role-arn IAM role arn \
--task-type AUTOMATION \
--task-invocation-parameters task parameters \
```

```
--priority task priority \
--max-concurrency 10% \
--max-errors 5
```

> **Note**
>
> Wenn Sie eine Automatisierung mithilfe von als registrierte Aufgabe konfigurieren AWS CLI, verwenden Sie den Parameter, um `--Task-Invocation-Parameters` Parameter anzugeben, die an eine Aufgabe übergeben werden, wenn sie ausgeführt wird. Verwenden Sie nicht den Parameter `--Task-Parameters`. Der Parameter `--Task-Parameters` ist veraltet.
>
> Bei Wartungsfensteraufgaben ohne festgelegtes Ziel können Sie keine Werte für `--max-errors` und `--max-concurrency` bereitstellen. Stattdessen fügt das System den Platzhalterwert 1 ein, der in der Antwort auf Befehle wie describe-maintenance-window-tasks und get-maintenance-window-task gemeldet wird. Diese Werte wirken sich nicht auf die Ausführung Ihrer Aufgabe aus und können ignoriert werden. Informationen zu Wartungsfenster-Tasks, für die keine Ziele erforderlich sind, finden Sie unter Wartungsfenster-Tasks ohne Ziele registrieren.

Windows

```
aws ssm register-task-with-maintenance-window ^
--window-id window ID ^
--name task name ^
--task-arn runbook name ^
--targets Key=targets,Values=value ^
--service-role-arn IAM role arn ^
--task-type AUTOMATION ^
--task-invocation-parameters task parameters ^
--priority task priority ^
--max-concurrency 10% ^
--max-errors 5
```

> **Note**
>
> Wenn Sie eine Automatisierung mithilfe von als registrierte Aufgabe konfigurieren AWS CLI, verwenden Sie den Parameter, um `--task-invocation-parameters` Parameter anzugeben, die an eine Aufgabe übergeben werden, wenn sie ausgeführt

wird. Verwenden Sie nicht den Parameter `--task-parameters`. Der Parameter `--task-parameters` ist veraltet.
Bei Wartungsfensteraufgaben ohne festgelegtes Ziel können Sie keine Werte für `--max-errors` und `--max-concurrency` bereitstellen. Stattdessen fügt das System den Platzhalterwert 1 ein, der in der Antwort auf Befehle wie describe-maintenance-window-tasks und get-maintenance-window-task gemeldet wird. Diese Werte wirken sich nicht auf die Ausführung Ihrer Aufgabe aus und können ignoriert werden.
Informationen zu Wartungsfenster-Tasks, für die keine Ziele erforderlich sind, finden Sie unter Wartungsfenster-Tasks ohne Ziele registrieren.

PowerShell

```
Register-SSMTaskWithMaintenanceWindow `
-WindowId window ID `
-Name "task name" `
-TaskArn "runbook name" `
-Target @{ Key="targets";Values="value" } `
-ServiceRoleArn "IAM role arn" `
-TaskType "AUTOMATION" `
-Automation_Parameter @{ "task parameter"="task parameter value"} `
-Priority task priority `
-MaxConcurrency 10% `
-MaxError 5
```

Note

Wenn Sie eine Automatisierung mithilfe von als registrierte Aufgabe konfigurieren AWS -Tools für PowerShell, verwenden Sie den Parameter, um `-Automation_Parameter` Parameter anzugeben, die an eine Aufgabe übergeben werden, wenn die Aufgabe ausgeführt wird. Verwenden Sie nicht den Parameter `-TaskParameters`. Der Parameter `-TaskParameters` ist veraltet.
Bei Wartungsfensteraufgaben ohne festgelegtes Ziel können Sie keine Werte für `-MaxError` und `-MaxConcurrency` bereitstellen. Stattdessen fügt das System den Platzhalterwert 1 ein, der in der Antwort auf Befehle wie `Get-SSMMaintenanceWindowTaskList` und `Get-SSMMaintenanceWindowTask` gemeldet wird. Diese Werte wirken sich nicht auf die Ausführung Ihrer Aufgabe aus und können ignoriert werden.

Informationen zu Wartungsfenster-Tasks, für die keine Ziele erforderlich sind, finden Sie unter Wartungsfenster-Tasks ohne Ziele registrieren.

Im folgenden Beispiel wird eine Automatisierung als registrierte Aufgabe für ein Wartungsfenster mit Priorität 1 konfiguriert. Es zeigt auch, dass die `--targets`, `--max-errors` und `--max-concurrency`- Optionen für eine ziellose Wartungsfensteraufgabe weggelassen werden. Die Automatisierung verwendet das `AWS-StartEC2Instance` Runbook und die angegebene Automationsübernahmerolle, um EC2 Instanzen zu starten, die als Ziele für das Wartungsfenster registriert sind. Das Wartungsfenster führt die Automatisierung gleichzeitig auf maximal 5 Instances zu einem bestimmten Zeitpunkt aus. Die Ausführung dieser registrierten Aufgabe wird außerdem für ein bestimmtes Intervall auf weiteren Instances gestoppt, wenn die Fehlerzählung 1 überschreitet.

Linux & macOS

```
aws ssm register-task-with-maintenance-window \
--window-id mw-0c50858d01EXAMPLE \
--name StartEC2Instances \
--task-arn AWS-StartEC2Instance \
--service-role-arn arn:aws:iam::123456789012:role/MaintenanceWindowRole \
--task-type AUTOMATION \
--task-invocation-parameters "{\"Automation\":{\"Parameters\":{\"InstanceId\":
[\"{{TARGET_ID}}\"],\"AutomationAssumeRole\":[\"arn:aws:iam::123456789012:role/
AutomationAssumeRole\"]}}}" \
--priority 1
```

Windows

```
aws ssm register-task-with-maintenance-window ^
--window-id mw-0c50858d01EXAMPLE ^
--name StartEC2Instances ^
--task-arn AWS-StartEC2Instance ^
--service-role-arn arn:aws:iam::123456789012:role/MaintenanceWindowRole ^
--task-type AUTOMATION ^
--task-invocation-parameters "{\"Automation\":{\"Parameters\":{\"InstanceId\":
[\"{{TARGET_ID}}\"],\"AutomationAssumeRole\":[\"arn:aws:iam::123456789012:role/
AutomationAssumeRole\"]}}}" ^
--priority 1
```

PowerShell

```
Register-SSMTaskWithMaintenanceWindow `
-WindowId mw-0c50858d01EXAMPLE `
-Name "StartEC2" `
-TaskArn "AWS-StartEC2Instance" `
-ServiceRoleArn "arn:aws:iam::123456789012:role/MaintenanceWindowRole" `
-TaskType "AUTOMATION" `
-Automation_Parameter
 @{ "InstanceId"="{{TARGET_ID}}";"AutomationAssumeRole"="arn:aws:iam::123456789012:role/
AutomationAssumeRole" } `
-Priority 1
```

Der Befehl gibt Details für die neue registrierte Aufgabe zurück, die den folgenden ähneln.

Linux & macOS

```
{
"WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
}
```

Windows

```
{
"WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
}
```

PowerShell

```
4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE
```

3. Um die registrierte Aufgabe anzuzeigen, führen Sie den folgenden Befehl aus. Ersetzen Sie es *maintenance windows ID* durch Ihre eigenen Informationen.

   Linux & macOS

   ```
   aws ssm describe-maintenance-window-tasks \
   --window-id maintenance window ID
   ```

Windows

```
aws ssm describe-maintenance-window-tasks ^
--window-id maintenance window ID
```

PowerShell

```
Get-SSMMaintenanceWindowTaskList `
-WindowId maintenance window ID
```

Das System gibt unter anderem folgende Informationen zurück

Linux & macOS

```
{
"Tasks": [
    {
        "ServiceRoleArn": "arn:aws:iam::123456789012:role/
MaintenanceWindowRole",
        "MaxErrors": "1",
        "TaskArn": "AWS-StartEC2Instance",
        "MaxConcurrency": "1",
        "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
        "TaskParameters": {},
        "Priority": 1,
        "WindowId": "mw-0c50858d01EXAMPLE",
        "Type": "AUTOMATION",
        "Targets": [
        ],
        "Name": "StartEC2"
    }
]
}
```

Windows

```
{
"Tasks": [
    {
```

```
        "ServiceRoleArn": "arn:aws:iam::123456789012:role/
MaintenanceWindowRole",
        "MaxErrors": "1",
        "TaskArn": "AWS-StartEC2Instance",
        "MaxConcurrency": "1",
        "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
        "TaskParameters": {},
        "Priority": 1,
        "WindowId": "mw-0c50858d01EXAMPLE",
        "Type": "AUTOMATION",
        "Targets": [
        ],
        "Name": "StartEC2"
    }
]
}
```

PowerShell

```
Description    :
LoggingInfo    :
MaxConcurrency : 5
MaxErrors      : 1
Name           : StartEC2
Priority       : 1
ServiceRoleArn : arn:aws:iam::123456789012:role/MaintenanceWindowRole
Targets        : {}
TaskArn        : AWS-StartEC2Instance
TaskParameters : {}
Type           : AUTOMATION
WindowId       : mw-0c50858d01EXAMPLE
WindowTaskId   : 4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE
```

Systems Manager Automation Aktionen-Referenz

Diese Referenz beschreibt die Automation-Aktionen, die Sie in einem Runbook angeben können. Automation ist ein Tool in AWS Systems Manager. Diese Aktionen können nicht in anderen Arten von Systems Manager (SSM)-Dokumenten verwendet werden. Weitere Informationen zu Plug-Ins für andere Arten von SSM-Dokumente finden Sie unter Referenz für Befehlsdokument-Plugins.

Die Systems Manager Automation führt Schritte aus, die in Automation-Runbooks definiert sind. Jeder Schritt ist einer bestimmten Aktion zugeordnet. Die Aktion bestimmt die Eingaben, das Verhalten und die Ausgaben des Schritts. Die Schritte sind im `mainSteps`-Bereich Ihres Runbooks definiert.

Sie müssen die Ausgaben einer Aktivität oder eines Schritts nicht angeben. Die Ausgaben werden im Voraus durch die dem Schritt zugeordnete Aktivität bestimmt. Wenn Sie Schritteingaben in Ihren Runbooks festlegen, können Sie auf mindestens eine Ausgabe aus einem früheren Schritt verweisen. Beispielsweise können Sie die Ausgabe von `aws:runInstances` für eine spätere `aws:runCommand`-Aktion verfügbar machen. Sie können auch auf Ausgaben aus früheren Schritten im Abschnitt `Output` des Runbooks verweisen.

Important

Wenn Sie einen automatisierten Workflow ausführen, der andere Services mithilfe einer AWS Identity and Access Management -(IAM)-Servicerolle aufruft, muss die Servicerolle mit der Berechtigung zum Aufrufen dieser Services konfiguriert sein. Diese Anforderung gilt für alle AWS Automatisierungs-Runbooks (`AWS-*`Runbooks) wie die`AWS-ConfigureS3BucketLogging`, und `AWS-RestartEC2Instance` Runbooks`AWS-CreateDynamoDBBackup`, um nur einige zu nennen. Diese Anforderung gilt auch für alle benutzerdefinierten Automatisierungs-Runbooks, die Sie erstellen und die andere mithilfe AWS-Services von Aktionen aufrufen, die andere Dienste aufrufen. Wenn Sie unter anderem `aws:executeAwsApi`-, `aws:createStack`- oder `aws:copyImage`-Aktionen verwenden, konfigurieren Sie die Dienstrolle mit der Berechtigung zum Aufrufen solcher Services. Sie können anderen AWS-Services Berechtigungen erteilen, indem Sie der Rolle eine eingebundene IAM-Richtlinie hinzufügen. Weitere Informationen finden Sie unter (Optional) Fügen Sie eine Inline-Automatisierungsrichtlinie oder eine vom Kunden verwaltete Richtlinie hinzu, um andere aufzurufen AWS-Services.

Themen

- Von allen Aktionen gemeinsam genutzte Eigenschaften
- aws:approve - Unterbrechen einer Automatisierung zur manuellen Genehmigung
- aws:assertAwsResourceProperty— Bestätigt einen AWS Ressourcen- oder Ereignisstatus
- aws:branch - Ausführen bedingter Automatisierungsschritte
- aws:changeInstanceState – Instance-Status ändern oder geltend machen

- aws:copyImage - Kopieren oder Verschlüsseln eines Amazon Machine Image
- aws:createImage - Erstellen eines Amazon Machine Image
- aws:createStack— Einen CloudFormation Stapel erstellen
- aws:createTags— Erstelle Tags für AWS Ressourcen
- aws:deleteImage - Löschen eines Amazon Machine Image
- aws:deleteStack— Löscht einen CloudFormation Stapel
- aws:executeAutomation - Führen Sie eine weitere Automatisierung durch
- aws:executeAwsApi— AWS API-Operationen aufrufen und ausführen
- aws:executeScript - Führen Sie ein Skript aus
- aws:executeStateMachine— Führen Sie eine AWS Step Functions Zustandsmaschine aus
- aws:invokeWebhook – Automation-Webhook-Integration aufrufen
- aws:invokeLambdaFunction— Ruft eine Funktion auf AWS Lambda
- aws:loop – Über Schritte in einer Automatisierung iterieren
- aws:pause - Pausieren einer Automatisierung
- aws:runCommand - Führt einen Befehl auf einer verwalteten Instance aus
- aws:runInstances— Starten Sie eine EC2 Amazon-Instance
- aws:sleep - Verzögerung einer Automatisierung
- aws:updateVariable – Aktualisiert einen Wert für eine Runbook-Variable
- aws:waitForAwsResourceProperty— Warte auf eine AWS Ressourceneigenschaft
- Systemvariablen für Automation

Von allen Aktionen gemeinsam genutzte Eigenschaften

Allgemeine Eigenschaften sind Parameter oder Optionen, die in allen Aktionen gefunden werden. Einige Optionen definieren das Verhalten für einen Schritt, etwa wie lange auf den Abschluss eines Schritts gewartet werden muss und was zu tun ist, wenn der Schritt fehlschlägt. Die folgenden Eigenschaften sind allen Aktionen gemeinsam.

description

Informationen, die Sie angeben, um den Zweck eines Runbooks oder eines Schritts zu beschreiben.

Typ: Zeichenfolge

Erforderlich: Nein

name

Ein Bezeichner, der für alle Schrittnamen im Runbook eindeutig sein muss.

Typ: Zeichenfolge

Zulässiges Muster: [a-zA-Z0-9_]+$

Erforderlich: Ja

action

Der Name der Aktion, die der Schritt ausführt. `aws:runCommand` - Führt einen Befehl auf einer verwalteten Instance aus ist ein Beispiel für eine Aktion, die Sie hier angeben können. Dieses Dokument enthält detaillierte Informationen über alle verfügbaren Aktionen.

Typ: Zeichenfolge

Erforderlich: Ja

maxAttempts

Die Anzahl der Wiederholungen des Schritt bei einem Fehler. Wenn der Wert größer als 1 ist, wird der Schritt erst als fehlgeschlagen betrachtet, wenn alle Wiederholungsversuche fehlgeschlagen sind. Der Standardwert ist 1.

Typ: Ganzzahl

Erforderlich: Nein

timeoutSeconds

Der Wert für das Timeout des Schritts. Wenn das Timeout erreicht ist und der Wert von `maxAttempts` größer als 1 ist, wird der Schritt erst als abgelaufen betrachtet, wenn alle Wiederholungen durchgeführt wurden.

Typ: Ganzzahl

Erforderlich: Nein

onFailure

Gibt an, ob die Automatisierung bei einem Fehler abgebrochen, fortgesetzt oder bis zu einem bestimmten Schritt übersprungen werden soll. Der Standardwert für diese Option ist "abort".

Typ: Zeichenfolge

Gültige Werte: Abort | Continue | step: *step_name*

Erforderlich: Nein

onCancel

Gibt an, zu welchem Schritt die Automatisierung gehen soll, falls ein Benutzer die Automatisierung abbricht. Die Automatisierung führt den Stornierungs-Workflow für maximal zwei Minuten aus.

Typ: Zeichenfolge

Gültige Werte: Abort | Schritt: *step_name*

Erforderlich: Nein

Die `onCancel`-Eigenschaft unterstützt das Verschieben zu den folgenden Aktionen nicht:

- `aws:approve`
- `aws:copyImage`
- `aws:createImage`
- `aws:createStack`
- `aws:createTags`
- `aws:loop`
- `aws:pause`
- `aws:runInstances`
- `aws:sleep`

isEnd

Diese Option stoppt eine Automatisierung am Ende eines bestimmten Schrittes. Die Automatisierung stoppt, egal ob der Schritt erfolgreich oder gar nicht ausgeführt werden konnte. Der Standardwert von "false".

Typ: Boolesch

Zulässige Werte: true | false

Erforderlich: Nein

nextStep

Gibt an, welcher Schritt in einer Automatisierung nach dem erfolgreichem Abschluss eines Schritts als nächster auszuführen ist.

Typ: Zeichenfolge

Erforderlich: Nein

isCritical

Bezeichnet einen Schritt als kritisch für den erfolgreichen Abschluss der Automation. Wenn ein Schritt mit dieser Bezeichnung fehlschlägt, dann wird der endgültige Status der Automation als fehlgeschlagen gemeldet. Diese Eigenschaft wird nur ausgewertet, wenn Sie diese explizit in Ihrem Schritt definieren. Wenn die `onFailure`-Eigenschaft auf `Continue` in einem Schritt gesetzt ist, lautet der Standardwert „false“. Der Standardwert für diese Option ist sonst „true“.

Typ: Boolesch

Zulässige Werte: true | false

Erforderlich: Nein

inputs

Die für die Aktivität spezifischen Eigenschaften.

Typ: Zuordnung

Erforderlich: Ja

Beispiel

```
---
description: "Custom Automation Example"
schemaVersion: '0.3'
assumeRole: "{{ AutomationAssumeRole }}"
parameters:
  AutomationAssumeRole:
    type: String
    description: "(Required) The ARN of the role that allows Automation to perform
      the actions on your behalf. If no role is specified, Systems Manager Automation
      uses your IAM permissions to run this runbook."
    default: ''
```

```
  InstanceId:
      type: String
      description: "(Required) The Instance Id whose root EBS volume you want to
 restore the latest Snapshot."
      default: ''
mainSteps:
- name: getInstanceDetails
  action: aws:executeAwsApi
  onFailure: Abort
  inputs:
    Service: ec2
    Api: DescribeInstances
    InstanceIds:
    - "{{ InstanceId }}"
  outputs:
    - Name: availabilityZone
      Selector: "$.Reservations[0].Instances[0].Placement.AvailabilityZone"
      Type: String
    - Name: rootDeviceName
      Selector: "$.Reservations[0].Instances[0].RootDeviceName"
      Type: String
  nextStep: getRootVolumeId
- name: getRootVolumeId
  action: aws:executeAwsApi
  maxAttempts: 3
  onFailure: Abort
  inputs:
    Service: ec2
    Api: DescribeVolumes
    Filters:
    -  Name: attachment.device
       Values: ["{{ getInstanceDetails.rootDeviceName }}"]
    -  Name: attachment.instance-id
       Values: ["{{ InstanceId }}"]
  outputs:
    - Name: rootVolumeId
      Selector: "$.Volumes[0].VolumeId"
      Type: String
  nextStep: getSnapshotsByStartTime
- name: getSnapshotsByStartTime
  action: aws:executeScript
  timeoutSeconds: 45
  onFailure: Abort
  inputs:
```

```
    Runtime: python3.8
    Handler: getSnapshotsByStartTime
    InputPayload:
      rootVolumeId : "{{ getRootVolumeId.rootVolumeId }}"
    Script: |-
      def getSnapshotsByStartTime(events,context):
        import boto3

        #Initialize client
        ec2 = boto3.client('ec2')
        rootVolumeId = events['rootVolumeId']
        snapshotsQuery = ec2.describe_snapshots(
          Filters=[
            {
              "Name": "volume-id",
              "Values": [rootVolumeId]
            }
          ]
        )
        if not snapshotsQuery['Snapshots']:
          noSnapshotFoundString = "NoSnapshotFound"
          return { 'noSnapshotFound' : noSnapshotFoundString }
        else:
          jsonSnapshots = snapshotsQuery['Snapshots']
          sortedSnapshots = sorted(jsonSnapshots, key=lambda k: k['StartTime'],
 reverse=True)
          latestSortedSnapshotId = sortedSnapshots[0]['SnapshotId']
          return { 'latestSnapshotId' : latestSortedSnapshotId }
  outputs:
  - Name: Payload
    Selector: $.Payload
    Type: StringMap
  - Name: latestSnapshotId
    Selector: $.Payload.latestSnapshotId
    Type: String
  - Name: noSnapshotFound
    Selector: $.Payload.noSnapshotFound
    Type: String
  nextStep: branchFromResults
- name: branchFromResults
  action: aws:branch
  onFailure: Abort
  onCancel: step:startInstance
  inputs:
```

```
    Choices:
    - NextStep: createNewRootVolumeFromSnapshot
      Not:
        Variable: "{{ getSnapshotsByStartTime.noSnapshotFound }}"
        StringEquals: "NoSnapshotFound"
  isEnd: true
- name: createNewRootVolumeFromSnapshot
  action: aws:executeAwsApi
  onFailure: Abort
  inputs:
    Service: ec2
    Api: CreateVolume
    AvailabilityZone: "{{ getInstanceDetails.availabilityZone }}"
    SnapshotId: "{{ getSnapshotsByStartTime.latestSnapshotId }}"
  outputs:
    - Name: newRootVolumeId
      Selector: "$.VolumeId"
      Type: String
  nextStep: stopInstance
- name: stopInstance
  action: aws:executeAwsApi
  onFailure: Abort
  inputs:
    Service: ec2
    Api: StopInstances
    InstanceIds:
    - "{{ InstanceId }}"
  nextStep: verifyVolumeAvailability
- name: verifyVolumeAvailability
  action: aws:waitForAwsResourceProperty
  timeoutSeconds: 120
  inputs:
    Service: ec2
    Api: DescribeVolumes
    VolumeIds:
    - "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
    PropertySelector: "$.Volumes[0].State"
    DesiredValues:
    - "available"
  nextStep: verifyInstanceStopped
- name: verifyInstanceStopped
  action: aws:waitForAwsResourceProperty
  timeoutSeconds: 120
  inputs:
```

```
    Service: ec2
    Api: DescribeInstances
    InstanceIds:
    - "{{ InstanceId }}"
    PropertySelector: "$.Reservations[0].Instances[0].State.Name"
    DesiredValues:
    - "stopped"
  nextStep: detachRootVolume
- name: detachRootVolume
  action: aws:executeAwsApi
  onFailure: Abort
  isCritical: true
  inputs:
    Service: ec2
    Api: DetachVolume
    VolumeId: "{{ getRootVolumeId.rootVolumeId }}"
  nextStep: verifyRootVolumeDetached
- name: verifyRootVolumeDetached
  action: aws:waitForAwsResourceProperty
  timeoutSeconds: 30
  inputs:
    Service: ec2
    Api: DescribeVolumes
    VolumeIds:
    - "{{ getRootVolumeId.rootVolumeId }}"
    PropertySelector: "$.Volumes[0].State"
    DesiredValues:
    - "available"
  nextStep: attachNewRootVolume
- name: attachNewRootVolume
  action: aws:executeAwsApi
  onFailure: Abort
  inputs:
    Service: ec2
    Api: AttachVolume
    Device: "{{ getInstanceDetails.rootDeviceName }}"
    InstanceId: "{{ InstanceId }}"
    VolumeId: "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
  nextStep: verifyNewRootVolumeAttached
- name: verifyNewRootVolumeAttached
  action: aws:waitForAwsResourceProperty
  timeoutSeconds: 30
  inputs:
    Service: ec2
```

```
    Api: DescribeVolumes
    VolumeIds:
    - "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
    PropertySelector: "$.Volumes[0].Attachments[0].State"
    DesiredValues:
    - "attached"
  nextStep: startInstance
- name: startInstance
  action: aws:executeAwsApi
  onFailure: Abort
  inputs:
    Service: ec2
    Api: StartInstances
    InstanceIds:
    - "{{ InstanceId }}"
```

`aws:approve` - Unterbrechen einer Automatisierung zur manuellen Genehmigung

Hält eine Automatisierung zeitweise an, bis die Aktion von designierten Prinzipalen genehmigt oder abgelehnt wird. Nach Erreichen der erforderlichen Anzahl an Genehmigungen wird die Automatisierung fortgesetzt. Sie können den Genehmigungsschritt an jeder beliebigen Stelle im `mainSteps`-Bereich Ihres Runbooks ansetzen.

> **Note**
>
> Diese Aktion unterstützt keine Automatisierungen für mehrere Konten und Regionen. Das Standard-Timeout für diese Aktion beträgt 7 Tage (604 800 Sekunden) und der Höchstwert ist 30 Tage (2 592 000 Sekunden). Sie können die Zeitüberschreitung über den Parameter `timeoutSeconds` für einen `aws:approve`-Schritt anpassen.

Im folgenden Beispiel hält die Aktion `aws:approve` die Automatisierung vorübergehend an, bis ein Genehmiger die Automatisierung entweder akzeptiert oder ablehnt. Bei entsprechender Genehmigung führt die Automatisierung einen einfachen PowerShell-Befehl aus.

YAML

```
---
description: RunInstancesDemo1
schemaVersion: '0.3'
assumeRole: "{{ assumeRole }}"
```

```
parameters:
  assumeRole:
    type: String
  message:
    type: String
mainSteps:
- name: approve
  action: aws:approve
  timeoutSeconds: 1000
  onFailure: Abort
  inputs:
    NotificationArn: arn:aws:sns:us-east-2:12345678901:AutomationApproval
    Message: "{{ message }}"
    MinRequiredApprovals: 1
    Approvers:
    - arn:aws:iam::12345678901:user/AWS-User-1
- name: run
  action: aws:runCommand
  inputs:
    InstanceIds:
    - i-1a2b3c4d5e6f7g
    DocumentName: AWS-RunPowerShellScript
    Parameters:
      commands:
      - date
```

JSON

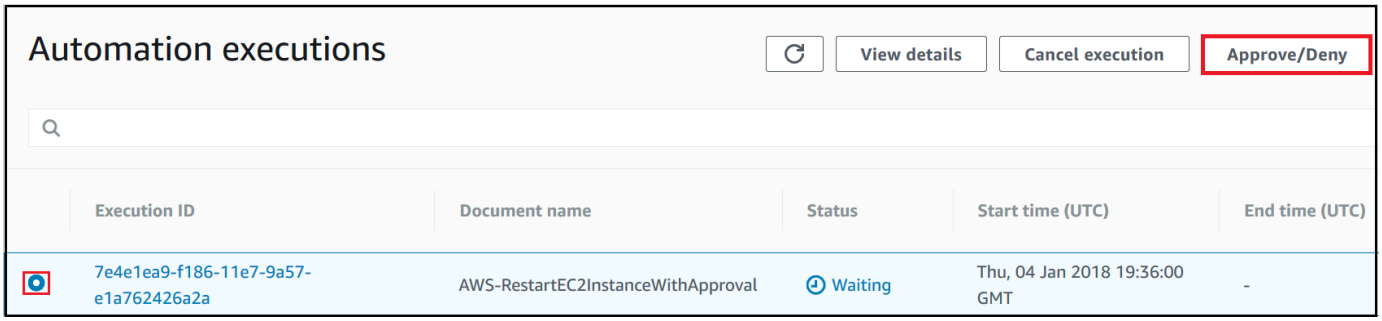
```
{
   "description":"RunInstancesDemo1",
   "schemaVersion":"0.3",
   "assumeRole":"{{ assumeRole }}",
   "parameters":{
      "assumeRole":{
         "type":"String"
      },
      "message":{
         "type":"String"
      }
   },
   "mainSteps":[
      {
         "name":"approve",
```

```
         "action":"aws:approve",
         "timeoutSeconds":1000,
         "onFailure":"Abort",
         "inputs":{
            "NotificationArn":"arn:aws:sns:us-
east-2:12345678901:AutomationApproval",
            "Message":"{{ message }}",
            "MinRequiredApprovals":1,
            "Approvers":[
               "arn:aws:iam::12345678901:user/AWS-User-1"
            ]
         }
      },
      {
         "name":"run",
         "action":"aws:runCommand",
         "inputs":{
            "InstanceIds":[
               "i-1a2b3c4d5e6f7g"
            ],
            "DocumentName":"AWS-RunPowerShellScript",
            "Parameters":{
               "commands":[
                  "date"
               ]
            }
         }
      }
   ]
}
```

Sie können Automatisierungen, die in der Konsole noch nicht genehmigt wurden, genehmigen oder ablehnen.

So genehmigen Sie Automatisierungen oder lehnen sie ab

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation.
3. Wählen Sie die Option neben einer Automation mit dem Status Waiting (Warten).



4. Wählen Sie Approve/Deny aus.
5. Überprüfen Sie die Details der Automation.
6. Wählen Sie Approve (Genehmigen) oder Deny (Verweigern), geben Sie einen optionalen Kommentar ein und wählen Sie dann Submit (Absenden) aus.

Eingabebeispiel

YAML

```
NotificationArn: arn:aws:sns:us-west-1:12345678901:Automation-ApprovalRequest
Message: Please approve this step of the Automation.
MinRequiredApprovals: 3
Approvers:
- IamUser1
- IamUser2
- arn:aws:iam::12345678901:user/IamUser3
- arn:aws:iam::12345678901:role/IamRole
```

JSON

```
{
   "NotificationArn":"arn:aws:sns:us-west-1:12345678901:Automation-ApprovalRequest",
   "Message":"Please approve this step of the Automation.",
   "MinRequiredApprovals":3,
   "Approvers":[
      "IamUser1",
      "IamUser2",
      "arn:aws:iam::12345678901:user/IamUser3",
      "arn:aws:iam::12345678901:role/IamRole"
   ]
}
```

NotificationArn

Der Amazon Resource Name (ARN) eines Amazon Simple Notification Service (Amazon SNS) Themas für Automation-Genehmigungen. Wenn Sie einen `aws:approve`-Schritt in einer Automatisierung festlegen, sendet Automation eine Nachricht an dieses Thema und informiert die Prinzipale darüber, dass sie einen Automation-Schritt entweder genehmigen oder zurückweisen müssen. Die Bezeichnung des Amazon-SNS-Themas muss das Präfix „Automatisierung“ aufweisen.

Typ: Zeichenfolge

Erforderlich: Nein

Fehlermeldung

Die Informationen, die Sie in das Amazon-SNS-Thema einbeziehen möchten, wenn die Genehmigungsanforderung gesendet wird. Die maximale Länge der Nachricht beträgt 4096 Zeichen.

Typ: Zeichenfolge

Erforderlich: Nein

MinRequiredApprovals

Die erforderliche Mindestanzahl an Genehmigungen zum Fortsetzen der Automatisierung. Wenn Sie keinen Wert angeben, verwendet das System standardmäßig den Wert 1. Der Wert für diesen Parameter muss eine positive Zahl sein. Der Wert für diesen Parameter darf nicht größer sein als die Anzahl der Genehmiger, die anhand des `Approvers`-Parameters definiert sind.

Typ: Ganzzahl

Erforderlich: Nein

Genehmiger

Eine Liste der authentifizierten AWS-Prinzipale, die die Aktion genehmigen oder ablehnen können. Die maximale Anzahl an Genehmigern ist 10. Sie können Prinzipale anhand eines der folgenden Formate festlegen:

- Ein Benutzername
- Ein Benutzer-ARN
- Ein IAM-Rollen-ARN
- Ein IAM-Rollenübernahme-ARN

Typ: StringList

Erforderlich: Ja

Verbesserte Genehmigungen

Diese Eingabe wird nur für Change Manager-Vorlagen verwendet. Eine Liste der AWS authentifizierten Prinzipals, die die Aktion entweder genehmigen oder ablehnen können, den Typ des IAM-Prinzipals und die Mindestanzahl von Genehmiger. Im Folgenden wird ein Beispiel gezeigt:

```
schemaVersion: "0.3"
emergencyChange: false
autoApprovable: false
mainSteps:
    - name: ApproveAction1
    action: aws:approve
    timeoutSeconds: 604800
    inputs:
        Message: Please approve this change request
        MinRequiredApprovals: 3
        EnhancedApprovals:
        Approvers:
            - approver: John Stiles
            type: IamUser
            minRequiredApprovals: 0
            - approver: Ana Carolina Silva
            type: IamUser
            minRequiredApprovals: 0
            - approver: GroupOfThree
            type: IamGroup
            minRequiredApprovals: 0
            - approver: RoleOfTen
            type: IamRole
            minRequiredApprovals: 0
```

Typ: StringList

Erforderlich: Ja

Ausgabe

ApprovalStatus

Der Genehmigungsstatus des Schritts. Der Status kann einer der folgenden sein: Genehmigt, Abgelehnt oder Warten. Warten bedeutet, dass Automation auf eine Eingabe der Genehmiger wartet.

Typ: Zeichenfolge

ApproverDecisions

Eine JSON-Karte, die den Genehmigungsbescheid der einzelnen Genehmiger enthält.

Typ: MapList

`aws:assertAwsResourceProperty`— Bestätigt einen AWS Ressourcen- oder Ereignisstatus

Die Aktion `aws:assertAwsResourceProperty` erlaubt Ihnen, einen bestimmten Ressourcen- oder Ereignisstatus für einen bestimmten Automation-Schritt zu prüfen.

> **Note**
>
> Die Aktion `aws:assertAwsResourceProperty` unterstützt die automatische Drosselungswiederholung. Weitere Informationen finden Sie unter Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge.

Weitere Beispiele zur Verwendung dieser Aktion finden Sie unter Weitere Runbook-Beispiele.

Eingabe

Eingaben werden von der ausgewählten API-Operation bestimmt.

YAML

```
action: aws:assertAwsResourceProperty
inputs:
  Service: The official namespace of the service
  Api: The API operation or method name
  API operation inputs or parameters: A value
  PropertySelector: Response object
  DesiredValues:
  - Desired property values
```

JSON

```
{
  "action": "aws:assertAwsResourceProperty",
  "inputs": {
    "Service":"The official namespace of the service",
    "Api":"The API operation or method name",
    "API operation inputs or parameters":"A value",
    "PropertySelector": "Response object",
    "DesiredValues": [
      "Desired property values"
    ]
  }
}
```

Service

Der AWS-Service Namespace, der die API-Operation enthält, die Sie ausführen möchten. Beispielsweise lautet der Namespace für Systems Manager `ssm`. Der Namespace für Amazon EC2 ist`ec2`. Sie finden eine Liste der unterstützten AWS-Service -Namespaces im Abschnitt Verfügbare Services der AWS CLI -Befehlsreferenz.

Typ: Zeichenfolge

Erforderlich: Ja

Api

Der Name der API-Operation, die Sie ausführen möchten. Sie können die API-Operationen (auch als Methoden bezeichnet) anzeigen, indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise werden alle API-Vorgänge (Methoden) für Amazon Relational Database Service (Amazon RDS) auf der folgenden Seite aufgelistet: Amazon RDS-Methoden.

Typ: Zeichenfolge

Erforderlich: Ja

API-Operation-Eingaben

Eine oder mehrere API-Eingaben. Sie können die verfügbaren Eingaben (auch als Parameter bezeichnet) anzeigen, indem Sie einen Service in der linken Navigationsleiste auf der folgenden

Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise sind alle Methoden für Amazon RDS auf der folgenden Seite aufgeführt: Amazon RDS-Methoden. Wählen Sie die Methode describe_db_instances und scrollen Sie nach unten, um die verfügbaren Parameter wie DBInstanceIdentifier, Name und Values zu sehen. Verwenden Sie das folgende Format, um mehr als eine Eingabe anzugeben.

YAML

```
inputs:
  Service: The official namespace of the service
  Api: The API operation name
  API input 1: A value
  API Input 2: A value
  API Input 3: A value
```

JSON

```
"inputs":{
      "Service":"The official namespace of the service",
      "Api":"The API operation name",
      "API input 1":"A value",
      "API Input 2":"A value",
      "API Input 3":"A value"
}
```

Typ: Abhängig von der gewählten API-Operation

Erforderlich: Ja

PropertySelector

Das JSONPath zu einem bestimmten Attribut im Antwortobjekt. Sie können die Antwortobjekte anzeigen indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise sind alle Methoden für Amazon RDS auf der folgenden Seite aufgeführt: Amazon RDS-Methoden. Wählen Sie die Methode describe_db_instances und scrollen Sie nach unten zum Abschnitt Antwortstruktur. DBInstancesist als Antwortobjekt aufgeführt.

Typ: Zeichenfolge

Erforderlich: Ja

DesiredValues

Die erwartete Status oder Zustand, bei dem die Automatisierung fortgesetzt werden soll. Wenn Sie einen booleschen Wert angeben, müssen Sie einen Großbuchstaben verwenden, wie z. B. True oder False.

Typ: StringList

Erforderlich: Ja

`aws:branch` - Ausführen bedingter Automatisierungsschritte

Die Aktion `aws:branch` erlaubt das Erstellen einer dynamischen Automatisierung, der verschiedene Auswahlmöglichkeiten in einem einzigen Schritt evaluiert und dann auf der Grundlage dieser Evaluierung zu einem anderen Schritt in dem Runbook springt.

Wenn Sie die Aktion `aws:branch` für einen Schritt angeben, geben Sie die `Choices` an, die die Automatisierung evaluieren muss. Die `Choices` können auf einem Wert basieren, den Sie im Abschnitt `Parameters` des Runbooks angegeben haben, oder auf einem als Ausgabe von dem vorherigen Schritt generierten dynamischen Wert basieren. Die Automatisierung evaluiert jede Auswahl mithilfe eines booleschen Ausdrucks. Wenn die erste Auswahl „wahr“ ist, springt die Automatisierung zu dem für diese Auswahl vorgesehenen Schritt. Wenn die erste Auswahl „false“ ist, evaluiert die Automatisierung die nächste Auswahl. Die Automatisierung evaluiert weiterhin jede Auswahl, bis eine Auswahl als „true“ verarbeitet wird. Die Automatisierung springt dann zu dem für die als „true“ evaluierte Auswahl angegebenen Schritt.

Wenn keine Auswahl als „true“ evaluiert wird, prüft die Automatisierung, ob der Schritt einen `default`-Wert enthält. Ein Default-Wert definiert einen Schritt, zu dem die Automatisierung springen soll, wenn keine der Auswahlmöglichkeiten als „true“ evaluiert wird. Wenn kein `default`-Wert für den Schritt definiert ist, verarbeitet die Automatisierung den nächsten Schritt in dem Runbook.

Die Aktion `aws:branch` unterstützt komplexe Auswahlevaluierungen durch Verwendung einer Kombination der Operatoren `And`, `Not` und `Or`. Weitere Informationen über die Verwendung von `aws:branch`, mit Beispielrunbooks und Beispielen, die unterschiedliche Operatoren verwenden, finden Sie unter Verwendung bedingter Anweisungen in Runbooks.

Eingabe

Geben Sie eine oder mehrere `Choices` in einem Schritt an. Die `Choices` können auf einem Wert basieren, den Sie im Abschnitt `Parameters` des Runbooks angegeben haben, oder auf einem als Ausgabe von dem vorherigen Schritt generierten dynamischen Wert basieren. Hier ist ein YAML-Beispiel, das einen Parameter evaluiert.

```
mainSteps:
- name: chooseOS
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runWindowsCommand
      Variable: "{{Name of a parameter defined in the Parameters section. For example:
 OS_name}}"
      StringEquals: windows
    - NextStep: runLinuxCommand
      Variable: "{{Name of a parameter defined in the Parameters section. For example:
 OS_name}}"
      StringEquals: linux
    Default:
      sleep3
```

Hier ist ein YAML-Beispiel, das die Ausgabe von einem vorherigen Schritt evaluiert.

```
mainSteps:
- name: chooseOS
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runPowerShellCommand
      Variable: "{{Name of a response object. For example: GetInstance.platform}}"
      StringEquals: Windows
    - NextStep: runShellCommand
      Variable: "{{Name of a response object. For example: GetInstance.platform}}"
      StringEquals: Linux
    Default:
      sleep3
```

Auswahlen

Ein oder mehrere Ausdrücke, die die Automatisierung evaluieren soll, wenn der nächste zu verarbeitende Schritt bestimmt wird. Auswahlen werden mit einem booleschen Ausdruck evaluiert. Jede Auswahl muss die folgenden Optionen definieren:

- NextStep: Der nächste Schritt in dem Runbook, der zu verarbeiten ist, wenn die betreffende Auswahl „true“ ist.
- Variable: Geben Sie entweder den Namen eines Parameters an, der im Abschnitt `Parameters` des Runbooks definiert ist, Oder geben Sie ein Ausgabeobjekt von einem vorherigen Schritt im Runbook an. Weitere Informationen zum Erstellen von Variablen für `aws:branch` finden Sie unter Informationen zum Erstellen der Ausgabevariable.
- Operation: Die Kriterien für die Evaluierung der Auswahl. Die Aktion `aws:branch` unterstützt die folgenden Operationen:

  Zeichenfolgenoperationen

  - StringEquals
  - EqualsIgnoreCase
  - StartsWith
  - EndsWith
  - Enthält

  Numerische Operationen

  - NumericEquals
  - NumericGreater
  - NumericLesser
  - NumericGreaterOrEquals
  - NumericLesser
  - NumericLesserOrEquals

  Boolesche Operation

  - BooleanEquals

  > **Important**
  >
  > Wenn Sie ein Runbook erstellen, validiert das System alle Operationen im Runbook. Wenn eine Operation nicht unterstützt wird, gibt das System einen Fehler aus, wenn Sie versuchen, das Runbook zu erstellen.

Standard

Der Name eines Schritts, zu dem die Automatisierung springen soll, wenn keine der `Choices` „true“ ist.

Typ: Zeichenfolge

Erforderlich: Nein

> **Note**
>
> Die Aktion `aws:branch` unterstützt die Operatoren `And`, `Or` und `Not`. Beispiele für `aws:branch` unter Verwendung von Operatoren finden Sie unter Verwendung bedingter Anweisungen in Runbooks.

`aws:changeInstanceState` – Instance-Status ändern oder geltend machen

Ändert oder klärt den Status der Instance.

Diese Aktivität kann im Assert-Modus verwendet werden (führt jedoch die API nicht aus, um den Status zu ändern, sondern prüft, ob die Instance den gewünschten Status aufweist.) Um den Assert-Modus zu verwenden, setzen Sie den Parameter `CheckStateOnly` auf "true". Dieser Modus ist nützlich, wenn der Sysprep-Befehl unter Windows Server ausgeführt wird. Bei diesem Befehl handelt es sich um einen asynchronen Befehl, der lange Zeit im Hintergrund ausgeführt werden kann. Sie können sicherstellen, dass die Instance angehalten wird, bevor Sie ein Amazon Machine Image (AMI) erstellen.

> **Note**
>
> Der Standardwert für die Zeitüberschreitung für diese Aktion beträgt 3 600 Sekunden (eine Stunde). Sie können die Zeitüberschreitung über den Parameter `timeoutSeconds` für einen `aws:changeInstanceState`-Schritt anpassen.

> **Note**
>
> Die Aktion `aws:changeInstanceState` unterstützt die automatische Drosselungswiederholung. Weitere Informationen finden Sie unter Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge.

Eingabe

YAML

```
name: stopMyInstance
action: aws:changeInstanceState
maxAttempts: 3
timeoutSeconds: 3600
onFailure: Abort
inputs:
  InstanceIds:
  - i-1234567890abcdef0
  CheckStateOnly: true
  DesiredState: stopped
```

JSON

```
{
    "name":"stopMyInstance",
    "action": "aws:changeInstanceState",
    "maxAttempts": 3,
    "timeoutSeconds": 3600,
    "onFailure": "Abort",
    "inputs": {
        "InstanceIds": ["i-1234567890abcdef0"],
        "CheckStateOnly": true,
        "DesiredState": "stopped"
    }
}
```

InstanceIds

Die IDs der Instanzen.

Typ: StringList

Erforderlich: Ja

CheckStateOnly

Wenn „false“, wird der Instance-Status auf den gewünschten Status festgelegt. Wenn „true“, wird der gewünschte Status anhand einer Abfrage überprüft.

Standard: `false`

Typ: Boolesch

Erforderlich: Nein

DesiredState

Der gewünschte Status. Wenn diese Option auf gesetzt ist`running`, wartet sie darauf, dass der EC2 Amazon-Status`Running`, der Instance-Status und der Systemstatus erreicht sind`OK`, `OK` bevor sie abgeschlossen wird.

Typ: Zeichenfolge

Zulässige Werte: `running` | `stopped` | `terminated`

Erforderlich: Ja

Force

Wenn festgelegt, wird das Anhalten der Instances erzwungen. Die Instances haben keine Gelegenheit, die Caches oder Metadaten des Dateisystems zu leeren. Wenn Sie diese Option verwenden, müssen Sie eine Überprüfung und Reparatur des Dateisystems durchführen. Diese Option wird nicht für EC2 Instances für Windows Server empfohlen.

Typ: Boolesch

Erforderlich: Nein

AdditionalInfo

Reserved Instances.

Typ: Zeichenfolge

Erforderlich: Nein

Output

Keine

`aws:copyImage` - Kopieren oder Verschlüsseln eines Amazon Machine Image

Kopiert ein Amazon Machine Image (AMI) aus einer beliebigen Region AWS-Region in die aktuelle Region. Diese Aktion kann auch das neue AMI verschlüsseln.

> **Note**
>
> Die Aktion `aws:copyImage` unterstützt die automatische Drosselungswiederholung. Weitere Informationen finden Sie unter Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge.

Eingabe

Diese Aktion unterstützt die meisten `CopyImage`-Parameter. Weitere Informationen finden Sie unter CopyImage.

Das folgende Beispiel erstellt eine Kopie eines AMI in der Region Seoul (`SourceImageID`: ami-0fe10819. `SourceRegion`: ap-northeast-2). Das neue AMI wird in die Region kopiert, in der Sie die Automation-Aktivität gestartet haben. Das kopierte AMI wird verschlüsselt, da das optionale `Encrypted`-Flag auf `true` gesetzt ist.

YAML

```
name: createEncryptedCopy
action: aws:copyImage
maxAttempts: 3
onFailure: Abort
inputs:
  SourceImageId: ami-0fe10819
  SourceRegion: ap-northeast-2
  ImageName: Encrypted Copy of LAMP base AMI in ap-northeast-2
  Encrypted: true
```

JSON

```
{
    "name": "createEncryptedCopy",
```

```
    "action": "aws:copyImage",
    "maxAttempts": 3,
    "onFailure": "Abort",
    "inputs": {
        "SourceImageId": "ami-0fe10819",
        "SourceRegion": "ap-northeast-2",
        "ImageName": "Encrypted Copy of LAMP base AMI in ap-northeast-2",
        "Encrypted": true
    }
}
```

SourceRegion

Die Region, in der Quell-AMI derzeit vorhanden ist.

Typ: Zeichenfolge

Erforderlich: Ja

SourceImageId

Die AMI-ID, die aus der Quellregion kopiert werden soll.

Typ: Zeichenfolge

Erforderlich: Ja

ImageName

Der Name für das neue Image.

Typ: Zeichenfolge

Erforderlich: Ja

ImageDescription

Eine Beschreibung des Ziel-Image.

Typ: Zeichenfolge

Erforderlich: Nein

Verschlüsselt

Verschlüsseln Sie das AMI.

Typ: Boolesch

Erforderlich: Nein

KmsKeyId

Der vollständige Amazon-Ressourcenname (ARN), der beim Verschlüsseln der Snapshots eines Images während eines Kopiervorgangs verwendet werden AWS KMS key soll. Weitere Informationen finden Sie unter CopyImage.

Typ: Zeichenfolge

Erforderlich: Nein

ClientToken

Ein eindeutiger Bezeichner, bei dem die Groß- und Kleinschreibung beachtet werden muss, um die Idempotenz der Anforderung sicherzustellen. Weitere Informationen finden Sie unter CopyImage.

Typ: Zeichenfolge

Erforderlich: Nein

Output

ImageId

Die ID des kopierten Image.

ImageState

Der Status des kopierten Image.

Zulässige Werte: `available` | `pending` | `failed`

`aws:createImage` - Erstellen eines Amazon Machine Image

Erstellt ein Amazon Machine Image (AMI) aus einer Instance, die entweder ausgeführt wird, angehalten wird oder angehalten ist, und fragt ab, ob `ImageState` gleich `available` ist.

> **Note**
>
> Die Aktion `aws:createImage` unterstützt die automatische Drosselungswiederholung. Weitere Informationen finden Sie unter Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge.

Eingabe

Diese Aktion unterstützt die folgenden `CreateImage`-Parameter. Weitere Informationen finden Sie unter CreateImage.

YAML

```
name: createMyImage
action: aws:createImage
maxAttempts: 3
onFailure: Abort
inputs:
  InstanceId: i-1234567890abcdef0
  ImageName: AMI Created on{{global:DATE_TIME}}
  NoReboot: true
  ImageDescription: My newly created AMI
```

JSON

```
{
    "name": "createMyImage",
    "action": "aws:createImage",
    "maxAttempts": 3,
    "onFailure": "Abort",
    "inputs": {
        "InstanceId": "i-1234567890abcdef0",
        "ImageName": "AMI Created on{{global:DATE_TIME}}",
        "NoReboot": true,
        "ImageDescription": "My newly created AMI"
    }
}
```

InstanceId

Die ID der Instance.

Typ: Zeichenfolge

Erforderlich: Ja

ImageName

Der Name für das Image.

Typ: Zeichenfolge

Erforderlich: Ja

ImageDescription

Eine Beschreibung des Image.

Typ: Zeichenfolge

Erforderlich: Nein

NoReboot

Ein boolesches Literal.

Standardmäßig versucht Amazon Elastic Compute Cloud (Amazon EC2), die Instance herunterzufahren und neu zu starten, bevor das Image erstellt wird. Wenn die Option Kein Neustart auf gesetzt ist`true`, fährt Amazon die Instance EC2 nicht herunter, bevor das Image erstellt wird. Wenn diese Option verwendet wird, kann die Integrität des Dateisystems auf dem erstellten Image nicht garantiert werden.

Wenn Sie nicht möchten, dass die Instance ausgeführt wird, nachdem Sie eine AMI daraus erstellt haben, verwenden Sie zuerst die `aws:changeInstanceState` – Instance-Status ändern oder geltend machen Aktion, um die Instance zu beenden, und verwenden Sie dann diese `aws:createImage` Aktion mit der NoRebootOption auf`true`.

Typ: Boolesch

Erforderlich: Nein

BlockDeviceMappings

Die Blockgeräte für die Instance.

Typ: Zuordnung

Erforderlich: Nein

Output

ImageId

Die ID des neu erstellten Image.

Typ: Zeichenfolge

ImageState

Der aktuelle Status des Image. Wenn der Status verfügbar ist, wird das Image erfolgreich registriert und kann zum Starten einer Instance verwendet werden.

Typ: Zeichenfolge

`aws:createStack`— Einen CloudFormation Stapel erstellen

Erzeugt einen AWS CloudFormation Stapel aus einer Vorlage.

> **Note**
>
> Die Aktion `aws:createStack` unterstützt die automatische Drosselungswiederholung. Weitere Informationen finden Sie unter Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge.

Zusätzliche Informationen zum Erstellen von CloudFormation Stacks finden Sie CreateStackin der AWS CloudFormation API-Referenz.

Eingabe

YAML

```
name: makeStack
action: aws:createStack
maxAttempts: 1
onFailure: Abort
inputs:
```

```
  Capabilities:
  - CAPABILITY_IAM
  StackName: myStack
  TemplateURL: http://s3.amazonaws.com/amzn-s3-demo-bucket/myStackTemplate
  TimeoutInMinutes: 5
  Parameters:
    - ParameterKey: LambdaRoleArn
      ParameterValue: "{{LambdaAssumeRole}}"
    - ParameterKey: createdResource
      ParameterValue: createdResource-{{automation:EXECUTION_ID}}
```

JSON

```
{
    "name": "makeStack",
    "action": "aws:createStack",
    "maxAttempts": 1,
    "onFailure": "Abort",
    "inputs": {
        "Capabilities": [
            "CAPABILITY_IAM"
        ],
        "StackName": "myStack",
        "TemplateURL": "http://s3.amazonaws.com/amzn-s3-demo-bucket/
myStackTemplate",
        "TimeoutInMinutes": 5,
        "Parameters": [
          {
            "ParameterKey": "LambdaRoleArn",
            "ParameterValue": "{{LambdaAssumeRole}}"
          },
          {
            "ParameterKey": "createdResource",
            "ParameterValue": "createdResource-{{automation:EXECUTION_ID}}"
          }
    }
}
```

Capabilities

Mit einer Liste von Werten, die Sie zuvor angegeben haben, CloudFormation können Sie bestimmte Stacks erstellen. Einige Stack-Vorlagen enthalten Ressourcen, die sich auf Ihre AWS-

Konto Berechtigungen auswirken können. Für einige Stacks müssen Sie deren Fähigkeiten mithilfe dieses Parameters explizit bestätigen.

Gültige Werte sind: `CAPABILITY_IAM`, `CAPABILITY_NAMED_IAM` und `CAPABILITY_AUTO_EXPAND`.

CAPABILITY_IAM und CAPABILITY_NAMED_IAM

Wenn Sie &IAM;-Ressourcen besitzen, können Sie jede Fähigkeit angeben.
Wenn Sie IAM-Ressourcen mit benutzerdefinierten Namen besitzen, müssen Sie `CAPABILITY_NAMED_IAM` angeben. Wenn Sie diesen Parameter angeben, gibt die Aktivität einen `InsufficientCapabilities`-Fehler zurück. Für die folgenden Ressourcen müssen Sie entweder `CAPABILITY_IAM` oder `CAPABILITY_NAMED_IAM` angeben.

- `AWS::IAM::AccessKey`
- `AWS::IAM::Group`
- `AWS::IAM::InstanceProfile`
- `AWS::IAM::Policy`
- `AWS::IAM::Role`
- `AWS::IAM::User`
- `AWS::IAM::UserToGroupAddition`

Wenn Ihre Stack-Vorlage diese Ressourcen enthält, empfehlen wir, dass Sie alle ihnen zugeordneten Berechtigungen überprüfen und ihre Berechtigungen bei Bedarf bearbeiten.

Weitere Informationen finden Sie unter Bestätigung von IAM-Ressourcen in CloudFormation Vorlagen.

CAPABILITY_AUTO_EXPAND

Einige Vorlagen enthalten Makros. Makros führen eine benutzerdefinierte Verarbeitung von Vorlagen durch. Dies kann einfache Aktionen wie find-and-replace Operationen bis hin zu umfangreichen Transformationen ganzer Vorlagen umfassen. Aus diesem Grund erstellt der Benutzer normalerweise einen Änderungssatz aus der verarbeiteten Vorlage, sodass er die aus den Makros resultierenden Änderungen überprüfen kann, bevor er den Stack tatsächlich erstellt. Wenn Ihre Stack-Vorlage ein oder mehrere Makros enthält und Sie sich dafür entscheiden, einen Stack direkt aus der verarbeiteten Vorlage zu erstellen, ohne vorher die resultierenden Änderungen in einem Änderungssatz zu überprüfen, müssen Sie diese Funktion berücksichtigen.

Weitere Informationen finden Sie im Benutzerhandbuch unter Verwenden von AWS CloudFormation Makros zur benutzerdefinierten Verarbeitung von Vorlagen.AWS CloudFormation

Typ: Zeichenfolge-Array

Zulässige Werte: `CAPABILITY_IAM | CAPABILITY_NAMED_IAM | CAPABILITY_AUTO_EXPAND`

Erforderlich: Nein

ClientRequestToken

Eine eindeutige Kennung für diese CreateStack Anfrage. Geben Sie dieses Token an, wenn Sie maxAttempts in diesem Schritt auf einen Wert größer als 1 festlegen. Durch die Angabe dieses Tokens CloudFormation weiß, dass Sie nicht versuchen, einen neuen Stack mit demselben Namen zu erstellen.

Typ: Zeichenfolge

Erforderlich: Nein

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 128 Zeichen.

Muster: [a-zA-Z0-9][-a-zA-Z0-9]*

DisableRollback

Legen Sie den Parameter auf `true` fest, um ein Rollback des Stacks zu deaktivieren, wenn ein Fehler bei der Erstellung des Stacks aufgetreten ist.

Bedingt: Sie können entweder den `DisableRollback`-Parameter oder den `OnFailure`-Parameter festlegen, aber nicht beide.

Standard: `false`

Typ: Boolesch

Erforderlich: Nein

Benachrichtigung ARNs

Das Thema Amazon Simple Notification Service (Amazon SNS) ARNs zur Veröffentlichung von Ereignissen im Zusammenhang mit Stacks. Sie finden das SNS-Thema ARNs in der Amazon SNS SNS-Konsole, https://console.aws.amazon.com/sns/ v3/home.

Typ: Zeichenfolge-Array

Array-Mitglieder: Maximale Anzahl von 5 Elementen.

Erforderlich: Nein

OnFailure

Bestimmt die Aktion, die ergriffen werden muss, wenn ein Fehler am Stack auftritt. Sie müssen `DO_NOTHING`, `ROLLBACK` oder `DELETE` angeben.

Bedingt: Sie können entweder den `OnFailure`-Parameter oder den `DisableRollback`-Parameter festlegen, aber nicht beide.

Standard: `ROLLBACK`

Typ: Zeichenfolge

Zulässige Werte: `DO_NOTHING | ROLLBACK | DELETE`

Erforderlich: Nein

Parameters

Eine Liste der `Parameter`-Strukturen, die Eingabeparameter für den Stack angeben. Weitere Informationen finden Sie im Datentyp Parameter.

Typ: Array von Parameter-Objekten

Erforderlich: Nein

ResourceTypes

Die Vorlagenressourcentypen für diese Aktion zum Erstellen von Stacks, für die Sie über Berechtigungen verfügen. Beispiel: `AWS::EC2::Instance`, `AWS::EC2::*` oder `Custom::MyCustomInstance`. Verwenden Sie die folgende Syntax zum Beschreiben von Vorlagenressourcentypen.

- Für alle Ressourcen: AWS

```
AWS::*
```

- Für alle benutzerdefinierten Ressourcen:

```
Custom::*
```

- Für eine bestimmte benutzerdefinierte Ressource:

```
Custom::logical_ID
```

- Für alle Ressourcen eines bestimmten AWS-Service:

```
AWS::service_name::*
```

- Für eine bestimmte AWS Ressource:

```
AWS::service_name::resource_logical_ID
```

Wenn die Liste der Ressourcentypen keine Ressource enthält, die Sie erstellen, schlägt die Erstellung des Stacks fehl. CloudFormation Gewährt standardmäßig Berechtigungen für alle Ressourcentypen. IAM verwendet diesen Parameter für CloudFormation -spezifische Bedingungsschlüssel in IAM-Richtlinien. Weitere Informationen finden Sie unter Zugriffskontrolle mit. AWS Identity and Access Management

Typ: Zeichenfolge-Array

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

RoleARN

Der Amazon-Ressourcenname (ARN) einer IAM-Rolle, die CloudFormation davon ausgeht, den Stack zu erstellen. CloudFormation verwendet die Anmeldeinformationen der Rolle, um in Ihrem Namen Anrufe zu tätigen. CloudFormation verwendet diese Rolle immer für alle future Operationen auf dem Stack. Wenn Benutzer die Berechtigung für Vorgänge am Stack besitzen, verwendet CloudFormation diese Rolle auch dann, wenn die Benutzer nicht über die Berechtigung zur Weitergabe verfügen. Stellen Sie sicher, dass die Rolle die geringstmögliche Menge an Berechtigungen gewährt.

Wenn Sie keinen Wert angeben, CloudFormation verwendet die Rolle, die zuvor dem Stack zugeordnet war. Wenn keine Rolle verfügbar ist, CloudFormation verwendet eine temporäre Sitzung, die anhand Ihrer Benutzeranmeldedaten generiert wird.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 20. Maximale Länge beträgt 2048 Zeichen.

Erforderlich: Nein

StackName

Der dem Stack zugeordnete Name. Der Name muss in der Region eindeutig sein, in der Sie den Stack erstellen.

> **Note**
>
> Ein Stack-Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Er muss mit einem alphabetischen Zeichen beginnen und darf nicht mehr als 128 Zeichen umfassen.

Typ: Zeichenfolge

Erforderlich: Ja

StackPolicyBody

Struktur, die die Stack-Richtlinie enthält. Weitere Informationen finden Sie unter Verhindern von Aktualisierungen der Stack-Ressourcen.

Bedingt: Sie können entweder den `StackPolicyBody`-Parameter oder den `StackPolicyURL`-Parameter festlegen, aber nicht beide.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 16384 Zeichen.

Erforderlich: Nein

StackPolicyURL

Speicherort einer Datei, die die Stack-Richtlinie enthält. Die URL muss auf eine Richtlinie in einem S3-Bucket in derselben Region wie der Stack verweisen. Die maximal zulässige Dateigröße für die Stack-Richtlinie ist 16 KB.

Bedingt: Sie können entweder den `StackPolicyBody`-Parameter oder den `StackPolicyURL`-Parameter festlegen, aber nicht beide.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 1350 Zeichen.

Erforderlich: Nein

Tags (Markierungen)

Schlüssel-Wert-Paare, die diesem Stack zugeordnet werden sollen. CloudFormation überträgt diese Tags auch auf die im Stack erstellten Ressourcen. Sie können höchstens 10 Tags angeben.

Typ: Array von Tag-Objekten

Erforderlich: Nein

TemplateBody

Struktur, die den Vorlagetext mit einer Mindestlänge von 1 Byte und einer Höchstlänge von 51.200 Byte enthält. Weitere Informationen finden Sie unter Aufbau einer Vorlage.

Bedingt: Sie können entweder den `TemplateBody`-Parameter oder den `TemplateURL`-Parameter festlegen, aber nicht beide.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen.

Erforderlich: Nein

TemplateURL

Speicherort einer Datei, die den Vorlagentext enthält. Die URL muss auf eine Vorlage verweisen, die sich in einem S3-Bucket befindet. Die maximal zulässige Größe für die Vorlage ist 460.800 Byte. Weitere Informationen finden Sie unter Aufbau einer Vorlage.

Bedingt: Sie können entweder den `TemplateBody`-Parameter oder den `TemplateURL`-Parameter festlegen, aber nicht beide.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Erforderlich: Nein

TimeoutInMinutes

Die Zeit, die verstreichen kann, bevor der Stack-Status zu `CREATE_FAILED` wird. Falls `DisableRollback` nicht festgelegt ist oder auf `false` festgelegt ist, wird für den Stack ein Rollback ausgeführt.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1.

Erforderlich: Nein

Outputs

StackId

Eindeutiger Bezeichner des Stacks.

Typ: Zeichenfolge

StackStatus

Aktueller Status des Stacks.

Typ: Zeichenfolge

Zulässige Werte: `CREATE_IN_PROGRESS | CREATE_FAILED | CREATE_COMPLETE | ROLLBACK_IN_PROGRESS | ROLLBACK_FAILED | ROLLBACK_COMPLETE | DELETE_IN_PROGRESS | DELETE_FAILED | DELETE_COMPLETE | UPDATE_IN_PROGRESS | UPDATE_COMPLETE_CLEANUP_IN_PROGRESS | UPDATE_COMPLETE | UPDATE_ROLLBACK_IN_PROGRESS | UPDATE_ROLLBACK_FAILED | UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS | UPDATE_ROLLBACK_COMPLETE | REVIEW_IN_PROGRESS`

Erforderlich: Ja

StackStatusReason

Erfolgs- oder Fehlermeldung im Zusammenhang mit dem Stack-Status.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter CreateStack.

Sicherheitsüberlegungen

Bevor Sie die Aktion `aws:createStack` verwenden können, müssen Sie folgende Richtlinie der IAM-Automation-Assume-Role zuweisen. Weitere Informationen über die Übernahmerolle finden Sie unter Aufgabe 1: Erstellen einer Servicerolle für Automation.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "sqs:*",
            "cloudformation:CreateStack",
            "cloudformation:DescribeStacks"
         ],
         "Resource":"*"
      }
   ]
}
```

`aws:createTags`— Erstelle Tags für AWS Ressourcen

Erstellt neue Tags für Amazon Elastic Compute Cloud (Amazon EC2) -Instances oder AWS Systems Manager verwaltete Instances.

> **Note**
>
> Die Aktion `aws:createTags` unterstützt die automatische Drosselungswiederholung. Weitere Informationen finden Sie unter Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge.

Eingabe

Diese Aktion unterstützt die meisten Amazon EC2 `CreateTags` - und Systems Manager `AddTagsToResource` Manager-Parameter. Weitere Informationen erhalten Sie unter CreateTags und AddTagsToResource.

Das folgende Beispiel zeigt, wie Sie ein Amazon Machine Image (AMI) und eine Instance als Produktionsressourcen für eine bestimmte Abteilung taggen.

YAML

```
name: createTags
action: aws:createTags
maxAttempts: 3
onFailure: Abort
inputs:
  ResourceType: EC2
  ResourceIds:
  - ami-9a3768fa
  - i-02951acd5111a8169
  Tags:
  - Key: production
    Value: ''
  - Key: department
    Value: devops
```

JSON

```
{
    "name": "createTags",
    "action": "aws:createTags",
    "maxAttempts": 3,
    "onFailure": "Abort",
    "inputs": {
        "ResourceType": "EC2",
        "ResourceIds": [
            "ami-9a3768fa",
            "i-02951acd5111a8169"
        ],
        "Tags": [
            {
                "Key": "production",
                "Value": ""
```

```
            },
            {
                "Key": "department",
                "Value": "devops"
            }
        ]
    }
}
```

ResourceIds

Die IDs Ressource (n), die markiert werden sollen. Wenn der Ressourcentyp nicht „EC2“ ist, kann dieses Feld nur ein einzelnes Element enthalten.

Typ: StringList

Erforderlich: Ja

Tags (Markierungen)

Die Tags, die der/den Ressource(n) zugeordnet werden sollen.

Typ: Liste von Karten

Erforderlich: Ja

ResourceType

Der Typ der Ressource(n), die getaggt werden soll(en). Wenn nicht angegeben, wird der Standardwert „EC2“ verwendet.

Typ: Zeichenfolge

Erforderlich: Nein

Zulässige Werte: `EC2` | `ManagedInstance` | `MaintenanceWindow` | `Parameter`

Output

Keine

`aws:deleteImage` - Löschen eines Amazon Machine Image

Löschen Sie das angegebene Amazon Machine Image (AMI) und alle dazugehörigen Snapshots.

> **Note**
>
> Die Aktion `aws:deleteImage` unterstützt die automatische Drosselungswiederholung. Weitere Informationen finden Sie unter Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge.

Eingabe

Diese Aktion unterstützt nur einen Parameter. Weitere Informationen finden Sie in der Dokumentation zu DeregisterImageund DeleteSnapshot.

YAML

```
name: deleteMyImage
action: aws:deleteImage
maxAttempts: 3
timeoutSeconds: 180
onFailure: Abort
inputs:
  ImageId: ami-12345678
```

JSON

```
{
    "name": "deleteMyImage",
    "action": "aws:deleteImage",
    "maxAttempts": 3,
    "timeoutSeconds": 180,
    "onFailure": "Abort",
    "inputs": {
        "ImageId": "ami-12345678"
    }
}
```

ImageId

Die ID des Image, das zerstört werden soll.

Typ: Zeichenfolge

Erforderlich: Ja

Output

Keine

`aws:deleteStack`— Löscht einen CloudFormation Stapel

Löscht einen AWS CloudFormation Stapel.

> **Note**
>
> Die Aktion `aws:deleteStack` unterstützt die automatische Drosselungswiederholung. Weitere Informationen finden Sie unter Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge.

Eingabe

YAML

```
name: deleteStack
action: aws:deleteStack
maxAttempts: 1
onFailure: Abort
inputs:
  StackName: "{{stackName}}"
```

JSON

```
{
   "name":"deleteStack",
   "action":"aws:deleteStack",
   "maxAttempts":1,
   "onFailure":"Abort",
   "inputs":{
      "StackName":"{{stackName}}"
   }
}
```

ClientRequestToken

Ein eindeutiger Bezeichner für diese `DeleteStack`-Anfrage. Geben Sie dieses Token an, wenn Sie beabsichtigen, Anfragen erneut zu versuchen, damit das System CloudFormation weiß, dass Sie nicht versuchen, einen Stack mit demselben Namen zu löschen. Sie können `DeleteStack`-Anfragen wiederholen, um zu verifizieren, ob CloudFormation sie empfangen hat.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 128 Zeichen.

Muster: [a-zA-Z][-a-zA-Z0-9]*

Erforderlich: Nein

RetainResources.Mitglied.N

Diese Eingabe gilt nur für Stacks im Status `DELETE_FAILED`. Eine Liste logischer Ressourcen IDs für die Ressourcen, die Sie behalten möchten. CloudFormation Löscht beim Löschen den Stapel, löscht jedoch nicht die zurückbehaltenen Ressourcen.

Das Aufbewahren der Ressourcen ist nützlich, wenn Sie eine Ressource nicht löschen können, wie etwa einen nicht leeren S3-Bucket, Sie aber den Stack löschen möchten.

Typ: Zeichenfolge-Array

Erforderlich: Nein

RoleARN

Der Amazon-Ressourcenname (ARN) einer AWS Identity and Access Management (IAM) - Rolle, die die Erstellung des Stacks CloudFormation übernimmt. CloudFormation verwendet die Anmeldeinformationen der Rolle, um in Ihrem Namen Anrufe zu tätigen. CloudFormation verwendet diese Rolle immer für alle future Operationen auf dem Stack. Solange Benutzer berechtigt sind, auf dem Stack zu arbeiten, CloudFormation verwendet diese Rolle auch dann, wenn die Benutzer nicht berechtigt sind, sie weiterzugeben. Stellen Sie sicher, dass die Rolle die geringstmögliche Menge an Berechtigungen gewährt.

Wenn Sie keinen Wert angeben, CloudFormation verwendet die Rolle, die zuvor dem Stack zugeordnet war. Wenn keine Rolle verfügbar ist, CloudFormation verwendet eine temporäre Sitzung, die anhand Ihrer Benutzeranmeldedaten generiert wird.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 20. Maximale Länge beträgt 2048 Zeichen.

Erforderlich: Nein

StackName

Der Name oder die eindeutige Stack-ID, die dem Stack zugeordnet ist.

Typ: Zeichenfolge

Erforderlich: Ja

Sicherheitsüberlegungen

Bevor Sie die Aktion `aws:deleteStack` verwenden können, müssen Sie folgende Richtlinie der IAM-Automation-Assume-Role zuweisen. Weitere Informationen über die Übernahmerolle finden Sie unter Aufgabe 1: Erstellen einer Servicerolle für Automation.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "sqs:*",
            "cloudformation:DeleteStack",
            "cloudformation:DescribeStacks"
         ],
         "Resource":"*"
      }
   ]
}
```

`aws:executeAutomation` - Führen Sie eine weitere Automatisierung durch

Führt eine sekundäre Automatisierung durch Aufrufen eines sekundären Runbooks aus. Mit dieser Aktion können Sie Runbooks für die gängigsten Vorgänge erstellen und während einer Automatisierung auf diese Runbooks verweisen. Mit dieser Aktion können Sie Ihre Runbooks vereinfachen, indem Sie die Notwendigkeit für wiederholte Schritte bei ähnlichen Runboks entfernen.

Die sekundäre Automatisierung wird im Kontext des Benutzers ausgeführt, der die primäre Automatisierung gestartet hat. Dies bedeutet, dass die sekundäre Automatisierung dieselbe AWS Identity and Access Management (IAM)-Rolle oder denselben Benutzer verwendet wie der Benutzer, der die erste Automatisierung gestartet hat.

> **Important**
>
> Wenn Sie Parameter in einer sekundären Automatisierung festlegen, die eine Übernahmerolle verwenden (eine Rolle, die die iam:passRole-Richtlinie verwendet), muss der Benutzer oder die Rolle, der/die die primäre Automatisierung gestartet hat, über die Berechtigung zur Weitergabe der Übernahmerolle an die sekundäre Automatisierung verfügen. Weitere Informationen zum Einrichten einer Übernahmerolle für Automation finden Sie unter Erstellen Sie die Servicerollen für Automation mithilfe der Konsole.

Eingabe

YAML

```
name: Secondary_Automation
action: aws:executeAutomation
maxAttempts: 3
timeoutSeconds: 3600
onFailure: Abort
inputs:
  DocumentName: secondaryAutomation
  RuntimeParameters:
    instanceIds:
    - i-1234567890abcdef0
```

JSON

```
{
   "name":"Secondary_Automation",
   "action":"aws:executeAutomation",
   "maxAttempts":3,
   "timeoutSeconds":3600,
   "onFailure":"Abort",
   "inputs":{
      "DocumentName":"secondaryAutomation",
      "RuntimeParameters":{
```

```
         "instanceIds":[
            "i-1234567890abcdef0"
         ]
      }
   }
}
```

DocumentName

Der Name des sekundären Runbooks, das während des Schritts ausgeführt werden soll. Geben Sie für Runbooks im gleichen AWS-Konto den Namen des Runbooks an. Geben Sie für Runbooks, die von einem anderen AWS-Konto geteilt wurden den Amazon-Ressourcennamen (ARN) des Runbooks an. Weitere Informationen zur Verwendung von freigegebenen Runbooks finden Sie unter Verwenden von freigegebenen SSM-Dokumenten.

Typ: Zeichenfolge

Erforderlich: Ja

DocumentVersion

Die Version des sekundären Runbooks, das ausgeführt werden soll. Falls nicht festgelegt, führt Automation die Standardrunbookversion aus.

Typ: Zeichenfolge

Erforderlich: Nein

MaxConcurrency

Die maximale Anzahl von Zielen, für die diese Aufgabe parallel ausgeführt werden dürfen. Sie können eine Zahl, z. B. 10, oder einen Prozentsatz, z. B. 10 %, angeben.

Typ: Zeichenfolge

Erforderlich: Nein

MaxErrors

Die Anzahl der Fehler, die zulässig sind, bevor das System die Automatisierung auf zusätzlichen Zielen stoppt. Sie können entweder eine absolute Anzahl an Fehlern, z. B. 10, oder einen Prozentsatz des festgelegten Ziels, beispielsweise 10 % festlegen. Wenn Sie z. B. 3 angeben, führt das System keine Automatisierung mehr aus, wenn der vierte Fehler empfangen wird.

Wenn Sie 0 angeben, führt das System keine weitere Automatisierung auf zusätzlichen Zielen aus, nachdem das erste Fehlerergebnis zurückgegeben wird. Wenn Sie eine Automatisierung auf 50 Ressourcen ausführen und `MaxErrors` auf 10 % setzen, hört das System auf, dass die Automatisierung auf zusätzlichen Zielen auszuführen, sobald der sechste Fehler empfangen wurde.

Automatisierung, die bereits eine ausgeführt werden, wenn der `MaxErrors`-Fehlerschwellenwert erreicht wird, können abgeschlossen werden, einige dieser Automatisierungen können jedoch dennoch fehlschlagen. Wenn Sie sicherstellen müssen, dass es nicht mehr fehlgeschlagene Automatisierungen als die angegebenen `MaxErrors` geben wird, setzen Sie `MaxConcurrency` auf 1, sodass die Automatisierungen nacheinander ausgeführt werden.

Typ: Zeichenfolge

Erforderlich: Nein

RuntimeParameters

Erforderliche Parameter für das sekundäre Runbook. Das Mapping verwendet das folgende Format: {"parameter1" : "value1", "parameter2" : "value2" }

Typ: Zuordnung

Erforderlich: Nein

Tags

Optionale Metadaten, die Sie einer Ressource zuweisen. Sie können maximal fünf Tags für eine Automatisierung festlegen.

Typ: MapList

Erforderlich: Nein

TargetLocations

Ein Standort ist eine Kombination aus AWS-Regionen und/oder AWS-Konten, wo Sie die Automatisierung ausführen möchten. Es muss eine Mindestanzahl von 1 Element angegeben werden und eine maximale Anzahl von 100 Elementen kann angegeben werden. Wenn Sie einen Wert für diesen Parameter angeben, werden die Ausgaben nicht an die übergeordnete Automatisierung zurückgegeben. Falls erforderlich, müssen Sie nachfolgende API-Vorgänge aufrufen, um die Ausgabe von untergeordneten Automatisierungen abzurufen.

Typ: MapList

Erforderlich: Nein

TargetMaps

Eine Liste von Schlüssel-Wert-Zuweisungen von Dokumentparametern zu Zielressourcen. Sowohl `Targets` als auch `TargetMaps` kann nicht zusammen angegeben werden.

Typ: MapList

Erforderlich: Nein

TargetParameterName

Der Name des Parameters, der als Zielressource für die ratengesteuerte Automatisierung verwendet wird. Erforderlich, wenn Sie `Targets` angeben.

Typ: Zeichenfolge

Erforderlich: Nein

Targets (Ziele)

Eine Liste von Schlüssel-Wert-Zuordnungen zu Zielressourcen. Erforderlich, wenn Sie `TargetParameterName` angeben.

Typ: MapList

Erforderlich: Nein

Output

Output

Die von der sekundären Automatisierung generierte Ausgabe. Sie können anhand des folgenden Formats auf die Ausgabe verweisen: *`Secondary_Automation_Step_Name`*.Output

Typ: StringList

Ein Beispiel:

```
- name: launchNewWindowsInstance
  action: 'aws:executeAutomation'
```

```
  onFailure: Abort
  inputs:
    DocumentName: launchWindowsInstance
  nextStep: getNewInstanceRootVolume
- name: getNewInstanceRootVolume
  action: 'aws:executeAwsApi'
  onFailure: Abort
  inputs:
    Service: ec2
    Api: DescribeVolumes
    Filters:
    - Name: attachment.device
      Values:
      - /dev/sda1
    - Name: attachment.instance-id
      Values:
      - '{{launchNewWindowsInstance.Output}}'
  outputs:
  - Name: rootVolumeId
    Selector: '$.Volumes[0].VolumeId'
    Type: String
  nextStep: snapshotRootVolume
- name: snapshotRootVolume
  action: 'aws:executeAutomation'
  onFailure: Abort
  inputs:
    DocumentName: AWS-CreateSnapshot
    RuntimeParameters:
    VolumeId:
    - '{{getNewInstanceRootVolume.rootVolumeId}}'
    Description:
    - 'Initial root snapshot for {{launchNewWindowsInstance.Output}}'
```

ExecutionId

Die ID der sekundären Automatisierung.

Typ: Zeichenfolge

Status

Der Status der sekundären Automatisierung.

Typ: Zeichenfolge

`aws:executeAwsApi`— AWS API-Operationen aufrufen und ausführen

Ruft AWS API-Operationen auf und führt sie aus. Die meisten API-Operationen werden unterstützt, es wurden jedoch nicht alle API-Operationen getestet. Streaming-API-Operationen, wie der GetObjectVorgang, werden nicht unterstützt. Wenn Sie sich nicht sicher sind, ob einen API-Vorgang, den Sie verwenden möchten, eine Streaming-Operation ist, lesen Sie die Boto3-Dokumentation für den Service, um festzustellen, ob eine API-Streaming-Eingaben oder -Ausgaben erfordert. Wir aktualisieren regelmäßig die von dieser Aktion verwendete Boto3-Version. Nach der Veröffentlichung einer neuen Boto3-Version kann es jedoch bis zu mehreren Wochen dauern, bis sich die Änderungen in dieser Aktion niederschlagen. Jede `aws:executeAwsApi`-Aktion kann bis zu einer maximalen Dauer von 25 Sekunden dauern. Weitere Beispiele zur Verwendung dieser Aktion finden Sie unter Weitere Runbook-Beispiele.

> **Note**
>
> Die Aktion `aws:executeAwsApi` unterstützt die automatische Drosselungswiederholung. Weitere Informationen finden Sie unter Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge.

Eingaben

Eingaben werden von der ausgewählten API-Operation bestimmt.

YAML

```
action: aws:executeAwsApi
inputs:
  Service: The official namespace of the service
  Api: The API operation or method name
  API operation inputs or parameters: A value
outputs: # These are user-specified outputs
- Name: The name for a user-specified output key
  Selector: A response object specified by using jsonpath format
  Type: The data type
```

JSON

```
{
   "action":"aws:executeAwsApi",
```

```
   "inputs":{
      "Service":"The official namespace of the service",
      "Api":"The API operation or method name",
      "API operation inputs or parameters":"A value"
   },
   "outputs":[ These are user-specified outputs
      {
         "Name":"The name for a user-specified output key",
         "Selector":"A response object specified by using JSONPath format",
         "Type":"The data type"
      }
   ]
}
```

Service

Der AWS-Service Namespace, der den API-Vorgang enthält, den Sie ausführen möchten. Eine Liste der unterstützten AWS-Service Namespaces finden Sie unter Verfügbare Dienste von. AWS SDK für Python (Boto3) Der Namespace befindet sich im Abschnitt Client . Beispielsweise lautet der Namespace für Systems Manager `ssm`. Der Namespace für Amazon Elastic Compute Cloud (Amazon EC2) lautet`ec2`.

Typ: Zeichenfolge

Erforderlich: Ja

Api

Der Name der API-Operation, die Sie ausführen möchten. Sie können die API-Operationen (auch als Methoden bezeichnet) anzeigen, indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise werden alle API-Vorgänge (Methoden) für Amazon Relational Database Service (Amazon RDS) auf der folgenden Seite aufgelistet: Amazon RDS-Methoden.

Typ: Zeichenfolge

Erforderlich: Ja

API-Operation-Eingaben

Eine oder mehrere API-Eingaben. Sie können die verfügbaren Eingaben (auch als Parameter bezeichnet) anzeigen, indem Sie einen Service in der linken Navigationsleiste auf der folgenden

Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise sind alle Methoden für Amazon RDS auf der folgenden Seite aufgeführt: Amazon RDS-Methoden. Wählen Sie die Methode describe_db_instances und scrollen Sie nach unten, um die verfügbaren Parameter wie DBInstanceIdentifier, Name und Values zu sehen.

YAML

```
inputs:
  Service: The official namespace of the service
  Api: The API operation name
  API input 1: A value
  API Input 2: A value
  API Input 3: A value
```

JSON

```
"inputs":{
      "Service":"The official namespace of the service",
      "Api":"The API operation name",
      "API input 1":"A value",
      "API Input 2":"A value",
      "API Input 3":"A value"
}
```

Typ: Abhängig von der gewählten API-Operation

Erforderlich: Ja

Outputs

Die Ausgaben werden vom Benutzer basierend auf der Antwort des ausgewählten API-Vorgangs angegeben.

Name

Ein Name für die Ausgabe.

Typ: Zeichenfolge

Erforderlich: Ja

Selector

Das JSONPath zu einem bestimmten Attribut im Antwortobjekt. Sie können die Antwortobjekte anzeigen indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise sind alle Methoden für Amazon RDS auf der folgenden Seite aufgeführt: Amazon RDS-Methoden. Wählen Sie die Methode describe_db_instances und scrollen Sie nach unten zum Abschnitt Antwortstruktur. DBInstancesist als Antwortobjekt aufgeführt.

Typ: Integer, Boolean, String, StringList, oder StringMap MapList

Erforderlich: Ja

Type (Typ)

Der Datentyp für das Antwortelement.

Typ: Unterschiedlich

Erforderlich: Ja

`aws:executeScript` - Führen Sie ein Skript aus

Führt das bereitgestellte Python- oder PowerShell Skript mit der angegebenen Laufzeit und dem angegebenen Handler aus. Jede `aws:executeScript`-Aktion kann bis zu einer maximalen Dauer von 600 Sekunden (10 Minuten) laufen. Sie können die Zeitüberschreitung über den Parameter `timeoutSeconds` für einen `aws:executeScript`-Schritt limitieren.

Verwenden Sie Rückgabe-Anweisungen in Ihrer Funktion, um Ihrer Ausgabenutzlast Ausgaben hinzuzufügen. Für Beispiele zum Definieren von Ausgaben für Ihre `aws:executeScript`-Aktion, siehe Beispiel 2: Skriptbasiertes Runbook. Sie können auch die Ausgabe von `aws:executeScript` Aktionen in Ihren Runbooks an die von Ihnen angegebene Amazon CloudWatch Logs-Protokollgruppe senden. Weitere Informationen finden Sie unter Protokollierung der Automation-Aktionsausgabe mit CloudWatch Logs.

Wenn Sie die Ausgabe von `aws:executeScript` Aktionen an CloudWatch Logs senden möchten oder wenn die Skripts, die Sie für `aws:executeScript` Aktionen angeben, AWS API-Operationen aufrufen, ist für die Ausführung des Runbooks immer eine AWS Identity and Access Management (IAM-) Service-Rolle (oder Übernahme einer Rolle) erforderlich.

> **Note**
>
> Die Aktion `aws:executeScript` unterstützt die automatische Drosselungswiederholung nicht. Wenn Ihr Skript AWS API-Aufrufe durchführt, die möglicherweise gedrosselt werden, müssen Sie Ihre eigene Wiederholungslogik in Ihren Skriptcode implementieren.

Die `aws:executeScript` Aktion enthält die folgenden PowerShell vorinstallierten Core-Module:

- Microsoft. PowerShell. Gastgeber
- Microsoft. PowerShell. Verwaltung
- Microsoft. PowerShell. Sicherheit
- Microsoft. PowerShell. Hilfsprogramm
- PackageManagement
- PowerShellGet

Um PowerShell Core-Module zu verwenden, die nicht vorinstalliert sind, muss Ihr Skript das Modul mit der `-Force` Markierung installieren, wie im folgenden Befehl gezeigt. Das `AWSPowerShell.NetCore`-Modul wird nicht unterstützt. *`ModuleName`*Ersetzen Sie es durch das Modul, das Sie installieren möchten.

```
Install-Module ModuleName -Force
```

Um PowerShell Core-Cmdlets in Ihrem Skript zu verwenden, empfehlen wir die Verwendung der `AWS.Tools` Module, wie in den folgenden Befehlen gezeigt. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen.

- Amazon S3 Cmdlets.

  ```
  Install-Module AWS.Tools.S3 -Force
  Get-S3Bucket -BucketName amzn-s3-demo-bucket
  ```

- EC2 Amazon-Cmdlets.

  ```
  Install-Module AWS.Tools.EC2 -Force
  Get-EC2InstanceStatus -InstanceId instance-id
  ```

- Allgemeine oder dienstunabhängige AWS Tools for Windows PowerShell Cmdlets.

```
Install-Module AWS.Tools.Common -Force
Get-AWSRegion
```

Wenn Ihr Skript zusätzlich zur Verwendung von PowerShell Core-Cmdlets neue Objekte initialisiert, müssen Sie das Modul auch importieren, wie im folgenden Befehl gezeigt.

```
Install-Module AWS.Tools.EC2 -Force
Import-Module AWS.Tools.EC2

$tag = New-Object Amazon.EC2.Model.Tag
$tag.Key = "Tag"
$tag.Value = "TagValue"

New-EC2Tag -Resource i-02573cafcfEXAMPLE -Tag $tag
```

Beispiele für die Installation und den Import von `AWS.Tools` Modulen und die Verwendung von PowerShell Core-Cmdlets in Runbooks finden Sie unter. Visuelle Designerfahrung für Automation-Runbooks

Eingabe

Geben Sie die zum Ausführen Ihres Skripts erforderlichen Informationen an. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

> **Note**
>
> Der Anhang für ein Python-Skript kann eine .py-Datei oder eine .zip-Datei sein, die das Skript enthält. PowerShell Skripten müssen in ZIP-Dateien gespeichert werden.

YAML

```
action: "aws:executeScript"
inputs:
 Runtime: runtime
 Handler: "functionName"
 InputPayload:
```

```
  scriptInput: '{{parameterValue}}'
 Script: |-
   def functionName(events, context):
   ...
 Attachment: "scriptAttachment.zip"
```

JSON

```
{
    "action": "aws:executeScript",
    "inputs": {
        "Runtime": "runtime",
        "Handler": "functionName",
        "InputPayload": {
            "scriptInput": "{{parameterValue}}"
        },
        "Attachment": "scriptAttachment.zip"
    }
}
```

Laufzeit

Die Laufzeitsprache, die für die Ausführung des bereitgestellten Skripts verwendet werden soll. `aws:executeScript`unterstützt die Laufzeiten in der folgenden Tabelle.

Laufzeitname	Laufzeitwert für die Eingabe ***runtime***
Python 3.10	**`python3.10`**
Python 3.11	**`python3.11`**
PowerShell 7.4 (`dotnet8`)	**`PowerShell 7.4`**

Typ: Zeichenfolge

Erforderlich: Ja

> **Note**
>
> Für Python-Laufzeiten bietet die Umgebung 512 MB Arbeitsspeicher und 512 MB Speicherplatz. Für PowerShell Laufzeiten stellt die Umgebung 1024 MB Arbeitsspeicher und 512 MB Festplattenspeicher zur Verfügung.

Handler

Der Name Ihrer Funktion. Sie müssen sicherstellen, dass die im Handler definierte Funktion über zwei Parameter verfügt: `events` und `context`. Die PowerShell Laufzeit unterstützt diesen Parameter nicht.

Typ: Zeichenfolge

Erforderlich: Ja (Python) | Nicht unterstützt (PowerShell)

InputPayload

Ein JSON- oder YAML-Objekt, das an den ersten Parameter des Handlers übergeben wird. Dies kann verwendet werden, um Eingabedaten an das Skript zu übergeben.

Typ: Zeichenfolge

Erforderlich: Nein

Python

```
description: Tag an instance
schemaVersion: '0.3'
assumeRole: '{{AutomationAssumeRole}}'
parameters:
    AutomationAssumeRole:
        type: String
        description: '(Required) The Amazon Resource Name (ARN) of the IAM role
 that allows Automation to perform the actions on your behalf. If no role is
 specified, Systems Manager Automation uses your IAM permissions to operate this
 runbook.'
    InstanceId:
        type: String
        description: (Required) The ID of the EC2 instance you want to tag.
mainSteps:
  - name: tagInstance
```

```
    action: 'aws:executeScript'
    inputs:
        Runtime: "python3.11"
        Handler: tagInstance
        InputPayload:
            instanceId: '{{InstanceId}}'
        Script: |-
          def tagInstance(events,context):
            import boto3

            #Initialize client
            ec2 = boto3.client('ec2')
            instanceId = events['instanceId']
            tag = {
                "Key": "Env",
                "Value": "ExamplePython"
            }
            print(f"Adding tag {tag} to instance id {instanceId}")
            ec2.create_tags(
                Resources=[instanceId],
                Tags=[tag]
            )
            return tag
    outputs:
      - Type: String
        Name: TagKey
        Selector: $.Payload.Key
outputs:
  - tagInstance.TagKey
```

PowerShell

```
description: Tag an instance
schemaVersion: '0.3'
assumeRole: '{{AutomationAssumeRole}}'
parameters:
  AutomationAssumeRole:
    type: String
    description: (Required) The Amazon Resource Name (ARN) of the IAM role
 that allows Automation to perform the actions on your behalf. If no role is
 specified, Systems Manager Automation uses your IAM permissions to operate this
 runbook.
  InstanceId:
```

```
    type: String
    description: (Required) The ID of the EC2 instance you want to tag.
mainSteps:
  - name: tagInstance
    action: aws:executeScript
    isEnd: true
    inputs:
      Runtime: PowerShell 7.4
      InputPayload:
        instanceId: '{{InstanceId}}'
      Script: |-
        Install-Module AWS.Tools.EC2 -Force
        Import-Module AWS.Tools.EC2

        $input = $env:InputPayload | ConvertFrom-Json

        $tag = New-Object Amazon.EC2.Model.Tag
        $tag.Key = "Env"
        $tag.Value = "ExamplePowerShell"

        Write-Information "Adding tag key: $($tag.Key) and value: $($tag.Value)
 to instance id $($input.instanceId)"
        New-EC2Tag -Resource $input.instanceId -Tag $tag

        return $tag
    outputs:
      - Type: String
        Name: TagKey
        Selector: $.Payload.Key
outputs:
  - tagInstance.TagKey
```

Script

Ein eingebettetes Skript, das während der Automatisierung ausgeführt werden soll.

Typ: Zeichenfolge

Erforderlich: Nein (Python) | Ja (PowerShell)

Attachment

Der Name einer eigenständigen Skriptdatei oder einer ZIP-Datei, die von der Aktion aufgerufen werden kann. Geben Sie denselben Wert wie den `Name` der Dokument-Anhangsdatei an, den

Sie im Anforderungsparameter `Attachments` angeben. Weitere Informationen finden Sie unter Anhänge in der API-Referenz für AWS Systems Manager . Wenn Sie ein Skript mithilfe einer Anlage bereitstellen, müssen Sie auch einen `files`-Abschnitt in den Elementen der obersten Ebene Ihres Runbooks definieren. Weitere Informationen finden Sie unter Schema der Version 0.3.

Um eine Datei für Python aufzurufen, verwenden Sie das `filename.method_name`-Format in `Handler`.

> **Note**
>
> Der Anhang für ein Python-Skript kann eine .py-Datei oder eine .zip-Datei sein, die das Skript enthält. PowerShell Skripten müssen in ZIP-Dateien gespeichert werden.

Wenn Sie Python-Bibliotheken in Ihren Anhang einfügen, empfehlen wir, eine leere `__init__.py`-Datei in jedem Modulverzeichnis hinzuzufügen. Auf diese Weise können Sie die Module aus der Bibliothek in Ihrem Anhang innerhalb Ihres Skriptinhalts importieren. Beispiel: `from library import module`

Typ: Zeichenfolge

Erforderlich: Nein

Output

Nutzlast

Die JSON-Darstellung des Objekts, das von Ihrer Funktion zurückgegeben wird. Bis zu 100 KB werden zurückgegeben. Wenn Sie eine Liste ausgeben, werden maximal 100 Elemente zurückgegeben.

Verwenden von Anhängen mit aws:executeScript

Anhänge bieten eine leistungsstarke Möglichkeit, komplexe Skripts, mehrere Module und externe Abhängigkeiten zu verpacken und für `aws:executeScript`-Aktionen wiederzuverwenden. Sie können Anhänge in folgenden Fällen verwenden:

- Packen Sie mehrere Python-Module oder PowerShell -Skripte zusammen.

- Wiederverwenden derselben Skriptlogik in mehreren Runbooks
- Einfügen externer Bibliotheken oder Abhängigkeiten in Skripts
- Runbook-Definition übersichtlich halten, indem Sie Elemente komplexer Skriptlogik trennen
- Freigeben von Skriptpaketen für Teams oder Automatisierungs-Workflows

Struktur und Verpackung von Anhängen

Sie können einzelne Dateien oder ZIP-Pakete mit mehreren Dateien anhängen. Die Struktur hängt von Ihrem Anwendungsfall ab:

Einzelne Dateianhänge

Bei einfachen Skripten können Sie eine einzelne `.py` Datei (Python) oder eine `.zip` Datei, die ein einzelnes PowerShell Skript enthält, anhängen.

Pakete mit mehreren Modulen

Für komplexe Automatisierungen, die mehrere Module erfordern, erstellen Sie ein ZIP-Paket mit der folgenden empfohlenen Struktur:

```
my-automation-package.zip
### main.py                     # Entry point script
### utils/
#   ### __init__.py            # Required for Python module imports
#   ### helper_functions.py    # Utility functions
#   ### aws_operations.py      # AWS-specific operations
### config/
#   ### __init__.py
#   ### settings.py            # Configuration settings
### requirements.txt           # Optional: document dependencies
```

> **Important**
>
> Für Python-Pakete müssen Sie in jedes Verzeichnis, das Python-Module enthält, eine leere `__init__.py`-Datei einfügen. Dadurch können Sie Module mit einer standardmäßigen Python-Importsyntax importieren, z. B. `from utils import helper_functions`.

PowerShell Struktur des Pakets

PowerShell Anlagen müssen in ZIP-Dateien mit der folgenden Struktur gepackt werden:

```
my-powershell-package.zip
### Main.ps1                  # Entry point script
### Modules/
#   ### HelperFunctions.ps1   # Utility functions
#   ### AWSOperations.ps1     # AWS-specific operations
### Config/
    ### Settings.ps1          # Configuration settings
```

Erstellen von Runbooks mit Anhängen

Gehen Sie wie folgt vor, um Runbooks zu erstellen, die Anhänge verwenden:

1. Anhang in Amazon S3 hochladen

   Laden Sie Ihre Skriptdatei oder Ihr ZIP-Paket in einen S3-Bucket hoch, auf den Ihre Automatisierungsrolle Zugriff hat. Notieren Sie den S3-URI für den nächsten Schritt.

   ```
   aws s3 cp my-automation-package.zip s3://my-automation-bucket/scripts/
   ```

2. Prüfsumme des Anhangs berechnen

   Berechnen Sie die SHA-256-Prüfsumme Ihrer Anhangsdatei zur Sicherheitsüberprüfung:

   ```
   # Linux/macOS
   shasum -a 256 my-automation-package.zip

   # Windows PowerShell
   Get-FileHash -Algorithm SHA256 my-automation-package.zip
   ```

3. Dateibereich im Runbook definieren

   Fügen Sie auf der obersten Ebene Ihres Runbooks einen `files`-Abschnitt hinzu, der auf Ihren Anhang verweist:

   ```
   files:
     my-automation-package.zip:
       checksums:
         sha256: "your-calculated-checksum-here"
   ```

4. Im executeScript-Schritt auf den Anhang verweisen

Verwenden Sie den Parameter `Attachment`, um auf die hochgeladene Datei zu verweisen:

```
- name: runMyScript
  action: aws:executeScript
  inputs:
    Runtime: python3.11
    Handler: main.process_data
    Attachment: my-automation-package.zip
    InputPayload:
      inputData: "{{InputParameter}}"
```

Beispiele für aws:executeScript-Anhänge

Die folgenden Beispiele zeigen verschiedene Möglichkeiten, Anhänge mit der Aktion `aws:executeScript` zu verwenden.

Beispiel 1: einzelner Dateianhang

Dieses Beispiel zeigt, wie eine einzelne Python-Datei als Anlage zur Verarbeitung von EC2 Instanzdaten verwendet wird.

Anhangsdatei: process_instance.py

Erstellen Sie eine Python-Datei mit folgendem Inhalt:

```
import boto3
import json

def process_instance_data(events, context):
    """Process EC2 instance data and return formatted results."""
    try:
        instance_id = events.get('instanceId')
        if not instance_id:
            raise ValueError("instanceId is required")

        ec2 = boto3.client('ec2')

        # Get instance details
        response = ec2.describe_instances(InstanceIds=[instance_id])
        instance = response['Reservations'][0]['Instances'][0]

        # Format the response
```

```
        result = {
            'instanceId': instance_id,
            'instanceType': instance['InstanceType'],
            'state': instance['State']['Name'],
            'availabilityZone': instance['Placement']['AvailabilityZone'],
            'tags': {tag['Key']: tag['Value'] for tag in instance.get('Tags', [])}
        }

        print(f"Successfully processed instance {instance_id}")
        return result

    except Exception as e:
        print(f"Error processing instance: {str(e)}")
        raise
```

Vollständiges Runbook

Hier ist das vollständige Runbook, das den einzelnen Dateianhang verwendet:

```
description: Process EC2 instance data using single file attachment
schemaVersion: '0.3'
assumeRole: '{{AutomationAssumeRole}}'
parameters:
  AutomationAssumeRole:
    type: String
    description: (Required) IAM role for automation execution
  InstanceId:
    type: String
    description: (Required) EC2 instance ID to process

files:
  process_instance.py:
    checksums:
      sha256: "abc123def456..."

mainSteps:
  - name: processInstance
    action: aws:executeScript
    inputs:
      Runtime: python3.11
      Handler: process_instance.process_instance_data
      Attachment: process_instance.py
      InputPayload:
        instanceId: '{{InstanceId}}'
```

```
    outputs:
      - Type: StringMap
        Name: InstanceData
        Selector: $.Payload

outputs:
  - processInstance.InstanceData
```

Beispiel 2: Paket mit mehreren Modulen

In diesem Beispiel wird ein ZIP-Paket, das mehrere Python-Module enthält, für komplexe S3-Bucket-Vorgänge verwendet.

Paketstruktur

Erstellen Sie ein ZIP-Paket mit der folgenden Struktur:

```
s3-operations.zip
### main.py
### utils/
#   ### __init__.py
#   ### s3_helper.py
#   ### validation.py
### config/
    ### __init__.py
    ### settings.py
```

main.py (Eintrittspunkt)

Das Hauptskript, das die Vorgänge orchestriert:

```
from utils.s3_helper import S3Operations
from utils.validation import validate_bucket_name
from config.settings import get_default_settings

def cleanup_s3_bucket(events, context):
    """Clean up S3 bucket based on specified criteria."""
    try:
        bucket_name = events.get('bucketName')
        max_age_days = events.get('maxAgeDays', 30)

        # Validate inputs
        if not validate_bucket_name(bucket_name):
```

```
            raise ValueError(f"Invalid bucket name: {bucket_name}")

        # Initialize S3 operations
        s3_ops = S3Operations()
        settings = get_default_settings()

        # Perform cleanup
        deleted_objects = s3_ops.delete_old_objects(
            bucket_name,
            max_age_days,
            settings['dry_run']
        )

        result = {
            'bucketName': bucket_name,
            'deletedCount': len(deleted_objects),
            'deletedObjects': deleted_objects[:10],  # Return first 10 for brevity
            'dryRun': settings['dry_run']
        }

        print(f"Cleanup completed for bucket {bucket_name}")
        return result

    except Exception as e:
        print(f"Error during S3 cleanup: {str(e)}")
        raise
```

Fehlerbehebung bei aws:executeScript-Anhängen

Beachten Sie die folgenden Informationen, um häufig auftretende Probleme mit `aws:executeScript`-Anhängen zu lösen:

Modulimportfehler

Wenn Sie bei Paketen mit mehreren Modulen Importfehler erhalten:

- Stellen Sie sicher, dass jedes Verzeichnis mit Python-Modulen eine leere `__init__.py`-Datei enthält.
- Prüfen Sie, ob Ihre Importanweisungen der tatsächlichen Datei- und Verzeichnisstruktur im ZIP-Paket entsprechen.
- Verwenden Sie konsistent relative Importe (z. B. `from .utils import helper`) oder absolute Importe (z. B. `from utils import helper`).

Anhang kann nicht gefunden werden

Wenn Ihre Automatisierung den Anhang nicht finden kann:

- Stellen Sie sicher, dass der `Attachment`-Parameterwert dem Schlüssel in Ihrem `files`-Abschnitt entspricht.
- Vergewissern Sie sich, dass Ihr S3-Bucket-Pfad und Ihr Dateiname im `files`-Abschnitt korrekt sind.
- Prüfen Sie, ob Ihre Automatisierungsrolle die Berechtigung `s3:GetObject` für den S3-Speicherort des Anhangs hat.
- Stellen Sie sicher, dass die Prüfsumme in Ihrem Runbook mit der tatsächlichen Dateiprüfsumme übereinstimmt.

Handler-Funktionsfehler

Wenn Fehler im Zusammenhang mit dem Handler auftreten:

- Für Python: Verwenden Sie das Format `filename.function_name` im `Handler`-Parameter (z. B. `main.process_data`).
- Sorgen Sie dafür, dass Ihre Handler-Funktion genau zwei Parameter akzeptiert: `events` und `context`.
- Für PowerShell: Geben Sie keinen `Handler` Parameter an; das Skript wird direkt ausgeführt.

Fehler bei der Skriptausführung

Wenn Ihr Skript während der Ausführung fehlschlägt:

- Im Ausführungsverlauf der Automatisierung finden Sie detaillierte Fehlermeldungen und Stack-Traces.
- Verwenden Sie `print()` Anweisungen (Python) oder `Write-Information` (PowerShell), um Debugging-Ausgaben hinzuzufügen.
- Stellen Sie sicher, dass Ihrer Automatisierungsrolle alle erforderlichen AWS Berechtigungen erteilt wurden.
- Testen Sie Ihre Skriptlogik lokal, bevor Sie sie als Anlage verpacken.

Beendigungscodes und Fehlerbehandlung

Gehen Sie wie folgt vor, um Fehler zu bearbeiten und Beendigungscodes zurückzugeben:

- In Python: Verwenden Sie `raise Exception("error message")`, um auf einen Skriptfehler hinzuweisen.
- In PowerShell: Verwenden Sie `throw "error message"` oder`Write-Error`, um auf einen Fehler hinzuweisen.
- Geben Sie strukturierte Daten aus Ihren Funktionen zurück, um detaillierte success/failure Informationen bereitzustellen.
- Verwenden Sie try-catch-Blöcke, um Ausnahmen ordnungsgemäß zu bearbeiten und aussagekräftige Fehlermeldungen bereitzustellen.

`aws:executeStateMachine`— Führen Sie eine AWS Step Functions Zustandsmaschine aus

Führt eine AWS Step Functions Zustandsmaschine aus.

> **Note**
>
> Die Aktion `aws:executeStateMachine` unterstützt die automatische Drosselungswiederholung. Weitere Informationen finden Sie unter Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge.

Eingabe

Diese Aktion unterstützt die meisten Parameter für den Step Functions StartExecutionFunctions-API-Vorgang.

Erforderliche AWS Identity and Access Management (IAM-) Berechtigungen

- `states:DescribeExecution`
- `states:StartExecution`
- `states:StopExecution`

YAML

```
name: executeTheStateMachine
action: aws:executeStateMachine
inputs:
```

```
  stateMachineArn: StateMachine_ARN
  input: '{"parameters":"values"}'
  name: name
```

JSON

```
{
    "name": "executeTheStateMachine",
    "action": "aws:executeStateMachine",
    "inputs": {
        "stateMachineArn": "StateMachine_ARN",
        "input": "{\"parameters\":\"values\"}",
        "name": "name"
    }
}
```

stateMachineArn

Der Amazon-Ressourcenname (ARN) der Step Functions State-Machine.

Typ: Zeichenfolge

Erforderlich: Ja

name

Der Name der Ausführung.

Typ: Zeichenfolge

Erforderlich: Nein

input

Eine Zeichenfolge, die die JSON-Eingabedaten für die Ausführung enthält.

Typ: Zeichenfolge

Erforderlich: Nein

Outputs

Die folgenden Ausgaben sind für diese Aktion vordefiniert.

executionArn

Der ARN der Ausführung.

Typ: Zeichenfolge

input

Die Zeichenfolge, die die JSON-Eingabedaten der Ausführung enthält. Längenbeschränkungen gelten für die Nutzlastgröße und werden als Bytes in UTF-8-Codierung ausgedrückt.

Typ: Zeichenfolge

name

Der Name der Ausführung.

Typ: Zeichenfolge

output

Die JSON-Ausgabedaten der Ausführung. Längenbeschränkungen gelten für die Nutzlastgröße und werden als Bytes in UTF-8-Codierung ausgedrückt.

Typ: Zeichenfolge

startDate

Das Datum, an dem die Ausführung gestartet wird.

Typ: Zeichenfolge

stateMachineArn

Der ARN des ausgeführten angegebenen Computers.

Typ: Zeichenfolge

Status

Der aktuelle Status der Ausführung.

Typ: Zeichenfolge

stopDate

Wenn die Ausführung bereits beendet wurde, das Datum, an dem die Ausführung beendet wurde.

Typ: Zeichenfolge

`aws:invokeWebhook` – Automation-Webhook-Integration aufrufen

Ruft die angegebene Automation-Webhook-Integration auf. Weitere Informationen zum Erstellen von Automation-Integrationen finden Sie unter Erstellen von Webhook-Integrationen für Automation.

> **Note**
>
> Die Aktion `aws:invokeWebhook` unterstützt die automatische Drosselungswiederholung. Weitere Informationen finden Sie unter Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge.

> **Note**
>
> Um die `aws:invokeWebhook`-Aktion zu verwenden, muss Ihre Benutzer- oder Servicerolle die folgenden Aktionen zulassen:
>
> - ssm: GetParameter
> - kms:Decrypt
>
> Die Genehmigung für den `Decrypt` Vorgang AWS Key Management Service (AWS KMS) ist nur erforderlich, wenn Sie einen vom Kunden verwalteten Schlüssel verwenden, um den Parameter für Ihre Integration zu verschlüsseln.

Eingabe

Geben Sie die Informationen für die aufzurufende Automation-Integration an.

YAML

```
action: "aws:invokeWebhook"
inputs:
 IntegrationName: "exampleIntegration"
 Body: "Request body"
```

JSON

```
{
    "action": "aws:invokeWebhook",
```

```
    "inputs": {
        "IntegrationName": "exampleIntegration",
        "Body": "Request body"
    }
}
```

IntegrationName

Der Name der Automation-Integration. Beispiel, `exampleIntegration`. Die von Ihnen angegebene Integration muss bereits vorhanden sein.

Typ: Zeichenfolge

Erforderlich: Ja

Fließtext

Die Nutzlast, die Sie beim Aufrufen der Webhook-Integration senden möchten.

Typ: Zeichenfolge

Erforderlich: Nein

Output

Antwort

Der Text aus der Antwort des Webhook-Anbieters.

ResponseCode

Der HTTP-Statuscode aus der Antwort des Webhook-Anbieters.

`aws:invokeLambdaFunction`— Ruft eine Funktion auf AWS Lambda

Ruft die angegebene AWS Lambda Funktion auf.

> **Note**
>
> Jede `aws:invokeLambdaFunction`-Aktion kann bis zu einer maximalen Dauer von 300 Sekunden (5 Minuten) laufen. Sie können die Zeitüberschreitung über den Parameter `timeoutSeconds` für einen `aws:invokeLambdaFunction`-Schritt limitieren.

> **Note**
>
> Die Aktion `aws:invokeLambdaFunction` unterstützt die automatische Drosselungswiederholung. Weitere Informationen finden Sie unter Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge.

Eingabe

Diese Aktion unterstützt die meisten aufgerufenen Parameter für den Lambda-Service. Weitere Informationen finden Sie unter Aufrufen.

YAML

```
name: invokeMyLambdaFunction
action: aws:invokeLambdaFunction
maxAttempts: 3
timeoutSeconds: 120
onFailure: Abort
inputs:
  FunctionName: MyLambdaFunction
```

JSON

```
{
    "name": "invokeMyLambdaFunction",
    "action": "aws:invokeLambdaFunction",
    "maxAttempts": 3,
    "timeoutSeconds": 120,
    "onFailure": "Abort",
    "inputs": {
        "FunctionName": "MyLambdaFunction"
    }
}
```

FunctionName

Der Name der Lambda-Funktion. Diese Funktion muss vorhanden sein.

Typ: Zeichenfolge

Erforderlich: Ja

Qualifier

Die Version oder der Aliasname der Funktion.

Typ: Zeichenfolge

Erforderlich: Nein

InvocationType

Der Aufruftyp. Der Standardwert ist `RequestResponse`.

Typ: Zeichenfolge

Zulässige Werte: `Event` | `RequestResponse` | `DryRun`

Erforderlich: Nein

LogType

Wenn der Standardwert `Tail` ist, muss der Aufruftyp `RequestResponse` sein. Lambda gibt die letzten 4 KB von Protokolldaten mit base64 verschlüsselt zurück, die von Ihrer Lambda-Funktion vorliegen.

Typ: Zeichenfolge

Zulässige Werte: `None` | `Tail`

Erforderlich: Nein

ClientContext

Die Client-spezifischen Informationen.

Erforderlich: Nein

InputPayload

Ein YAML- oder JSON-Objekt, das an den ersten Parameter des Handlers übergeben wird. Sie können diese Eingabe verwenden, um Daten an die Funktion zu übergeben. Diese Eingabe bietet mehr Flexibilität und Unterstützung als die Legacy-`Payload`-Eingabe. Wenn Sie sowohl `InputPayload` als auch `Payload` für die Aktion definieren, hat `InputPayload` Vorrang, und der `Payload`-Wert wird nicht verwendet.

Typ: StringMap

Erforderlich: Nein

Nutzlast

Eine JSON-Zeichenfolge, die an den ersten Parameter des Handlers übergeben wird. Dies kann verwendet werden, um Eingabedaten an die Funktion zu übergeben. Wir empfehlen die Verwendung der `InputPayload`-Eingabe für zusätzliche Funktionen.

Typ: Zeichenfolge

Erforderlich: Nein

Output

StatusCode

Den HTTP-Statuscode .

FunctionError

Falls vorhanden, weist es darauf hin, dass währen der Ausführung der Funktion ein Fehler aufgetreten ist. Fehlerdetails sind in der Antwortnutzlast enthalten.

LogResult

Die mit base64 verschlüsselten Protokolle zum Aufrufen der Lambda-Funktion. Protokolle sind nur dann vorhanden, wenn der Aufrufen-Typ `RequestResponse` ist und die Protokolle angefragt wurden.

Nutzlast

Die JSON-Darstellung des Objekts, das von der Lambda-Funktion zurückgegeben wird. Die Nutzlast ist nur vorhanden, wenn der Aufrufen-Typ `RequestResponse` ist.

Das Folgende ist ein Teil des `AWS-PatchInstanceWithRollback`-Runbooks, der zeigt, wie auf Ausgaben der `aws:invokeLambdaFunction`-Aktion verwiesen wird.

YAML

```
- name: IdentifyRootVolume
  action: aws:invokeLambdaFunction
```

```
  inputs:
    FunctionName: "IdentifyRootVolumeLambda-{{automation:EXECUTION_ID}}"
    Payload: '{"InstanceId": "{{InstanceId}}"}'
- name: PrePatchSnapshot
  action: aws:executeAutomation
  inputs:
    DocumentName: "AWS-CreateSnapshot"
    RuntimeParameters:
      VolumeId: "{{IdentifyRootVolume.Payload}}"
      Description: "ApplyPatchBaseline restoration case contingency"
```

JSON

```
{
    "name": "IdentifyRootVolume",
    "action": "aws:invokeLambdaFunction",
    "inputs": {
      "FunctionName": "IdentifyRootVolumeLambda-{{automation:EXECUTION_ID}}",
      "Payload": "{\"InstanceId\": \"{{InstanceId}}\"}"
    }
  },
  {
    "name": "PrePatchSnapshot",
    "action": "aws:executeAutomation",
    "inputs": {
      "DocumentName": "AWS-CreateSnapshot",
      "RuntimeParameters": {
        "VolumeId": "{{IdentifyRootVolume.Payload}}",
        "Description": "ApplyPatchBaseline restoration case contingency"
      }
    }
  }
```

`aws:loop` – Über Schritte in einer Automatisierung iterieren

Diese Aktion wiederholt sich über eine Teilmenge von Schritten in einem Automation-Runbook. Sie können einen Schleifenstil `do while` oder `for each` eine Schleife wählen. Verwenden Sie den `LoopCondition`-Eingabeparameter, um eine `do while`-Schleife zu erstellen. Verwenden Sie die Eingabeparameter `Iterators` und `IteratorDataType`, um eine `for each`-Schleife zu erstellen. Wenn Sie eine `aws:loop`-Aktion verwenden, geben Sie nur entweder den Eingabeparameter `Iterators` oder `LoopCondition` an. Die maximale Anzahl von Iterationen beträgt 100.

Die `onCancel`-Eigenschaft kann nur für Schritte definiert werden, die innerhalb einer Schleife genutzt sind. Die `onCancel`-Eigenschaft wird für die `aws:loop`-Aktion nicht unterstützt. Die `onFailure`-Eigenschaft kann für eine `aws:loop`-Aktion verwendet werden, sie wird jedoch nur verwendet, wenn ein unerwarteter Fehler auftritt, der dazu führt, dass der Schritt fehlschlägt. Wenn Sie `onFailure`-Eigenschaften für die Schritte innerhalb einer Schleife definieren, erbt die `aws:loop`-Aktion diese Eigenschaften und reagiert entsprechend, wenn ein Fehler auftritt.

Beispiele

Im Folgenden finden Sie Beispiele für die Erstellung der verschiedenen Typen von Loop-Aktionen.

do while

```
name: RepeatMyLambdaFunctionUntilOutputIsReturned
action: aws:loop
inputs:
    Steps:
    - name: invokeMyLambda
        action: aws:invokeLambdaFunction
        inputs:
        FunctionName: LambdaFunctionName
        outputs:
        - Name: ShouldRetry
            Selector: $.Retry
            Type: Boolean
    LoopCondition:
        Variable: "{{ invokeMyLambda.ShouldRetry }}"
        BooleanEquals: true
    MaxIterations: 3
```

for each

```
name: stopAllInstancesWithWaitTime
action: aws:loop
inputs:
    Iterators: "{{ DescribeInstancesStep.InstanceIds }}"
    IteratorDataType: "String"
    Steps:
    - name: stopOneInstance
        action: aws:changeInstanceState
        inputs:
        InstanceIds:
```

```
            - "{{stopAllInstancesWithWaitTime.CurrentIteratorValue}}"
        CheckStateOnly: false
        DesiredState: stopped
    - name: wait10Seconds
        action: aws:sleep
        inputs:
        Duration: PT10S
```

Eingabe

Die Eingabe ist wie folgt.

Iteratoren

Die Liste der Elemente, über die die Schritte iteriert werden sollen. Die maximale Anzahl von Iteratoren beträgt 100.

Typ: StringList

Erforderlich: Nein

IteratorDataType

Ein optionaler Parameter zur Angabe des Datentyps von `Iterators`. Ein Wert für diesen Parameter kann zusammen mit dem `Iterators`-Eingabeparameter angegeben werden. Wenn Sie keinen Wert für diesen Parameter und `Iterators` angeben, müssen Sie einen Wert für den `LoopCondition`-Parameter angeben.

Typ: Zeichenfolge

Gültige Werte: Boolesch | Ganzzahl | Zeichenfolge | StringMap

Standard: Zeichenfolge

Erforderlich: Nein

LoopCondition

Besteht aus `Variable` und einer auszuwertenden Operatorbedingung. Wenn Sie keinen Wert für diesen Parameter angeben, müssen Sie einen Wert für die `Iterators`- und `IteratorDataType`-Parameter angeben. Sie können komplexe Operatorauswertungen verwenden, indem Sie eine Kombination aus Operatoren `And`, `Not` und `Or` verwenden. Die

Bedingung wird bewertet, nachdem die Schritte in der Schleife abgeschlossen sind. Wenn die Bedingung `true` ist und der `MaxIterations`-Wert nicht erreicht wurde, werden die Schritte in der Schleife erneut ausgeführt. Die Bedingungen für den Operator lauten wie folgt:

Zeichenfolgenoperationen

- StringEquals
- EqualsIgnoreCase
- StartsWith
- EndsWith
- Enthält

Numerische Operationen

- NumericEquals
- NumericGreater
- NumericLesser
- NumericGreaterOrEquals
- NumericLesser
- NumericLesserOrEquals

Boolesche Operation

- BooleanEquals

Typ: StringMap

Erforderlich: Nein

MaxIterations

Gibt an, wie oft die Schritte in der Schleife maximal ausgeführt werden. Sobald der für diese Eingabe angegebene Wert erreicht ist, stoppt die Schleife, auch wenn `LoopCondition` immer noch `true` ist oder im `Iterators`-Parameter verbleibende Objekte vorhanden sind.

Typ: Ganzzahl

Zulässige Werte: 1–100

Erforderlich: Nein

Schritte

Die Liste der auszuführenden Schritte. Diese funktionieren wie ein verschachteltes Runbook. In diesen Schritten können Sie mithilfe der `{{loopStepName.CurrentIteratorValue}}`-Syntax auf den aktuellen Iteratorwert für eine `for each`-Schleife zugreifen. Sie können mithilfe der `{{loopStepName.CurrentIteration}}`-Syntax auch auf einen Integer-Wert der aktuellen Iteration für beide Schleifentypen zugreifen.

Typ: Liste der Schritte

Erforderlich: Ja

Output

CurrentIteration

Die aktuelle Schleifeniteration als Ganzzahl. Iterationswerte beginnen bei 1.

Typ: Ganzzahl

Aktueller Iterator-Wert

Der Wert des aktuellen Iterators als Zeichenfolge. Diese Ausgabe ist nur in `for each`-Schleifen vorhanden.

Typ: Zeichenfolge

`aws:pause` - Pausieren einer Automatisierung

Mit dieser Aktion wird die Ausführung der Automatisierung unterbrochen. Nach der Unterbrechung lautet der Automation-Status Waiting. Um die Automatisierung fortzusetzen, verwenden Sie die API-Operation SendAutomationSignal mit dem Signaltyp `Resume`. Wir empfehlen die Verwendung von der `aws:sleep`- oder `aws:approve`-Aktion zur genaueren Kontrolle Ihrer Workflows.

> **Note**
>
> Das Standard-Timeout für diese Aktion beträgt 7 Tage (604 800 Sekunden) und der Höchstwert ist 30 Tage (2 592 000 Sekunden). Sie können die Zeitüberschreitung über den Parameter `timeoutSeconds` für einen `aws:pause`-Schritt anpassen.

Eingabe

Die Eingabe ist wie folgt.

YAML

```
name: pauseThis
action: aws:pause
timeoutSeconds: 1209600
inputs: {}
```

JSON

```
{
    "name": "pauseThis",
    "action": "aws:pause",
    "timeoutSeconds": "1209600",
    "inputs": {}
}
```

Output

Keine

`aws:runCommand` - Führt einen Befehl auf einer verwalteten Instance aus

Führt die angegebenen Befehle aus.

> **Note**
>
> Automation unterstützt nur die Ausgabe einer AWS Systems Manager Run Command-Aktion. Ein Runbook kann mehrere Run Command-Aktionen und enthalten, die Ausgabe wird allerdings nur für je eine Aktion unterstützt.

Eingabe

Diese Aktion unterstützt die meisten Befehlsendeparameter. Weitere Informationen finden Sie unter SendCommand.

YAML

```
- name: checkMembership
  action: 'aws:runCommand'
  inputs:
    DocumentName: AWS-RunPowerShellScript
    InstanceIds:
      - '{{InstanceIds}}'
    Parameters:
      commands:
        - (Get-WmiObject -Class Win32_ComputerSystem).PartOfDomain
```

JSON

```
{
    "name": "checkMembership",
    "action": "aws:runCommand",
    "inputs": {
        "DocumentName": "AWS-RunPowerShellScript",
        "InstanceIds": [
            "{{InstanceIds}}"
        ],
        "Parameters": {
            "commands": [
                "(Get-WmiObject -Class Win32_ComputerSystem).PartOfDomain"
            ]
        }
    }
}
```

DocumentName

Geben Sie den Namen des Dokuments an, wenn das Befehlstyp-Dokument Ihnen oder AWS gehört. Geben Sie den Amazon-Ressourcennamen (ARN) des Dokuments an, wenn Sie ein Dokument verwenden, das von einem anderen AWS-Konto mit Ihnen geteilt wird. Weitere Informationen zur Verwendung von geteilten Dokumenten finden Sie unter Verwenden von freigegebenen SSM-Dokumenten.

Typ: Zeichenfolge

Erforderlich: Ja

InstanceIds

Die Instance-IDs, auf denen Sie den Befehl ausführen möchten. Sie können maximal 50 IDs angeben.

Sie können auch Pseudoparameter `{{RESOURCE_ID}}` anstelle von Instance-IDs verwenden, um den Befehl auf allen Instances in der Zielgruppe auszuführen. Weitere Informationen zu Pseudoparametern finden Sie unter Verwendung von Pseudo-Parametern bei der Registrierung von Aufgaben im Wartungsfenster.

Alternativ können Sie Befehle mit dem Parameter `Targets` an eine Instance-Flotte senden. Die `Targets`-Parameter akzeptiert Amazon Elastic Compute Cloud (Amazon EC2)-Tags. Weitere Informationen zur Verwendung des Parameters `Targets` finden Sie unter Ausführen von Befehlen in großem Maßstab.

Typ: StringList

Erforderlich: Nein (Wenn Sie keine InstanceIds angeben oder den -`{{RESOURCE_ID}}`Pseudoparameter verwenden, müssen Sie den Parameter `Targets` angeben.)

Targets (Ziele)

Ein Array von Suchkriterien, die Instances mit einer Schlüssel-Wert-Kombination adressieren, die Sie angeben. Der Wert `Targets` ist erforderlich, wenn Sie nicht eine oder mehrere Instance-IDs im Aufruf angeben. Weitere Informationen zur Verwendung des Parameters `Targets` finden Sie unter Ausführen von Befehlen in großem Maßstab.

Typ: MapList (Das Schema der Map in der Liste muss mit dem Objekt übereinstimmen.) Informationen finden Sie unter Target in der AWS Systems Manager-API-Referenz.

Erforderlich: Nein (Wenn Sie nicht `Targets` angeben, müssen Sie InstanceIds angeben oder den `{{RESOURCE_ID}}`-Pseudoparameter verwenden.)

Im Folgenden sehen Sie ein Beispiel.

YAML

```
- name: checkMembership
  action: aws:runCommand
  inputs:
    DocumentName: AWS-RunPowerShellScript
    Targets:
```

```
      - Key: tag:Stage
        Values:
          - Gamma
          - Beta
      - Key: tag-key
        Values:
          - Suite
    Parameters:
      commands:
        - (Get-WmiObject -Class Win32_ComputerSystem).PartOfDomain
```

JSON

```
{
    "name": "checkMembership",
    "action": "aws:runCommand",
    "inputs": {
        "DocumentName": "AWS-RunPowerShellScript",
        "Targets": [
            {
                "Key": "tag:Stage",
                "Values": [
                    "Gamma", "Beta"
                ]
            },
            {
                "Key": "tag:Application",
                "Values": [
                    "Suite"
                ]
            }
        ],
        "Parameters": {
            "commands": [
                "(Get-WmiObject -Class Win32_ComputerSystem).PartOfDomain"
            ]
        }
    }
}
```

Parameter

Die erforderlichen und optionalen Parameter, die im Dokument angegeben sind.

Typ: Zuordnung

Erforderlich: Nein

CloudWatchOutputConfig

Konfigurationsoptionen für das Senden der Befehlsausgabe an Amazon CloudWatch Logs. Weitere Informationen zum Senden von Befehlsausgaben an CloudWatch Logs finden Sie unter Konfigurieren von Amazon CloudWatch Logs für Run Command.

Typ: StringMap (Das Schema der Map muss dem Objekt entsprechen. Weitere Informationen finden Sie unter CloudWatchOutputConfig in der AWS Systems ManagerAPI-Referenz).

Erforderlich: Nein

Im Folgenden sehen Sie ein Beispiel.

YAML

```
- name: checkMembership
  action: aws:runCommand
  inputs:
    DocumentName: AWS-RunPowerShellScript
    InstanceIds:
      - "{{InstanceIds}}"
    Parameters:
      commands:
        - "(Get-WmiObject -Class Win32_ComputerSystem).PartOfDomain"
    CloudWatchOutputConfig:
      CloudWatchLogGroupName: CloudWatchGroupForSSMAutomationService
      CloudWatchOutputEnabled: true
```

JSON

```
{
    "name": "checkMembership",
    "action": "aws:runCommand",
    "inputs": {
        "DocumentName": "AWS-RunPowerShellScript",
        "InstanceIds": [
            "{{InstanceIds}}"
        ],
        "Parameters": {
            "commands": [
```

```
                "(Get-WmiObject -Class Win32_ComputerSystem).PartOfDomain"
            ]
        },
        "CloudWatchOutputConfig" : {
                "CloudWatchLogGroupName":
 "CloudWatchGroupForSSMAutomationService",
                "CloudWatchOutputEnabled": true
        }
    }
}
```

Kommentar

Benutzerdefinierte Informationen über den Befehl.

Typ: Zeichenfolge

Erforderlich: Nein

DocumentHash

Der Hash für das Dokument.

Typ: Zeichenfolge

Erforderlich: Nein

DocumentHashType

Der Typ des Hash.

Typ: Zeichenfolge

Zulässige Werte: `Sha256` | `Sha1`

Erforderlich: Nein

NotificationConfig

Die Konfigurationen für das Senden von Benachrichtigungen.

Erforderlich: Nein

OutputS3BucketName

Der Name des S3-Buckets für Befehlsausgabeantworten. Ihr verwalteter Knoten muss Berechtigungen haben, damit der S3-Bucket die Ausgabe erfolgreich protokollieren kann.

Typ: Zeichenfolge

Erforderlich: Nein

OutputS3KeyPrefix

Das Präfix.

Typ: Zeichenfolge

Erforderlich: Nein

ServiceRoleArn

Der ARN der AWS Identity and Access Management-IAM-Rolle.

Typ: Zeichenfolge

Erforderlich: Nein

TimeoutSeconds

Die Menge an Zeit in Sekunden, die auf die Übermittlung eines Befehls an den AWS Systems Manager SSM Agent auf einer Instance vergeht. Wenn der Befehl vom SSM Agent auf der Instance vor dem Erreichen des angegebenen Werts nicht empfangen wurde, ändert sich der Status des Befehls in `Delivery Timed Out`.

Typ: Ganzzahl

Erforderlich: Nein

Zulässige Werte: 30 bis 2 592 000

Output

CommandId

Die ID des Befehls.

Status

Der Status des Befehls.

ResponseCode

Der Antwortcode des Befehls. Wenn das Dokument, das Sie ausführen, mehr als einen Schritt umfasst, wird für diese Ausgabe kein Wert zurückgegeben.

Output

Die Ausgabe des Befehls. Wenn Sie mit Ihrem Befehl auf ein Tag oder mehrere Instances abzielen, wird kein Ausgabewert zurückgegeben. Sie können die API-Vorgänge `GetCommandInvocation` und `ListCommandInvocations` verwenden, um Ausgaben für einzelne Instances abzurufen.

`aws:runInstances`— Starten Sie eine EC2 Amazon-Instance

Startet eine neue Amazon Elastic Compute Cloud (Amazon EC2) -Instance.

> **Note**
>
> Die Aktion `aws:runInstances` unterstützt die automatische Drosselungswiederholung. Weitere Informationen finden Sie unter Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge.

Eingabe

Die Aktion unterstützt die meisten API-Parameter. Weitere Informationen finden Sie in der RunInstancesAPI-Dokumentation.

YAML

```
name: launchInstance
action: aws:runInstances
maxAttempts: 3
timeoutSeconds: 1200
onFailure: Abort
inputs:
  ImageId: ami-12345678
  InstanceType: t2.micro
  MinInstanceCount: 1
  MaxInstanceCount: 1
  IamInstanceProfileName: myRunCmdRole
  TagSpecifications:
  - ResourceType: instance
    Tags:
    - Key: LaunchedBy
      Value: SSMAutomation
```

```
    - Key: Category
      Value: HighAvailabilityFleetHost
```

JSON

```
{
   "name":"launchInstance",
   "action":"aws:runInstances",
   "maxAttempts":3,
   "timeoutSeconds":1200,
   "onFailure":"Abort",
   "inputs":{
      "ImageId":"ami-12345678",
      "InstanceType":"t2.micro",
      "MinInstanceCount":1,
      "MaxInstanceCount":1,
      "IamInstanceProfileName":"myRunCmdRole",
      "TagSpecifications":[
         {
            "ResourceType":"instance",
            "Tags":[
               {
                  "Key":"LaunchedBy",
                  "Value":"SSMAutomation"
               },
               {
                  "Key":"Category",
                  "Value":"HighAvailabilityFleetHost"
               }
            ]
         }
      ]
   }
}
```

AdditionalInfo

Reserved Instances.

Typ: Zeichenfolge

Erforderlich: Nein

BlockDeviceMappings

Die Blockgeräte für die Instance.

Typ: MapList

Erforderlich: Nein

ClientToken

Der Bezeichner, um die Idempotenz der Anfrage sicherzustellen.

Typ: Zeichenfolge

Erforderlich: Nein

DisableApiTermination

Aktiviert oder deaktiviert die Instance-API-Beendigung.

Typ: Boolesch

Erforderlich: Nein

EbsOptimized

Aktiviert oder deaktiviert die Amazon Elastic Block Store (Amazon EBS)-Optimierung.

Typ: Boolesch

Erforderlich: Nein

IamInstanceProfileArn

Der Amazon-Ressourcenname (ARN) des AWS Identity and Access Management (IAM) - Instance-Profils für die Instance.

Typ: Zeichenfolge

Erforderlich: Nein

IamInstanceProfileName

Der Name des IAM-Instance-Profils für die Instance.

Typ: Zeichenfolge

Erforderlich: Nein

ImageId

Die ID des Amazon Machine Image (AMI).

Typ: Zeichenfolge

Erforderlich: Ja

InstanceInitiatedShutdownBehavior

Gibt an, ob die Instance beim Herunterfahren des Systems angehalten oder beendet wird.

Typ: Zeichenfolge

Erforderlich: Nein

InstanceType

Der Instance-Typ.

> **Note**
>
> Wenn kein Wert für den Instance-Typ angegeben wird, wird der Instance-Typ m1.small verwendet.

Typ: Zeichenfolge

Erforderlich: Nein

KernelId

Die ID des Kernels.

Typ: Zeichenfolge

Erforderlich: Nein

KeyName

Der Name des Schlüsselpaars.

Typ: Zeichenfolge

Erforderlich: Nein

MaxInstanceCount

Die Höchstanzahl zu startender Instances.

Typ: Zeichenfolge

Erforderlich: Nein

MetadataOptions

Die Metadatenoptionen für die Instance. Weitere Informationen finden Sie unter InstanceMetadataOptionsRequest.

Typ: StringMap

Erforderlich: Nein

MinInstanceCount

Die Mindestanzahl zu startender Instances.

Typ: Zeichenfolge

Erforderlich: Nein

Überwachen

Aktiviert oder deaktiviert die detaillierte Überwachung.g

Typ: Boolesch

Erforderlich: Nein

NetworkInterfaces

Die Netzwerkschnittstellen.

Typ: MapList

Erforderlich: Nein

Placement

Die Platzierung für die Instance.

Typ: StringMap

Erforderlich: Nein

PrivateIpAddress

Die primäre IPv4 Adresse.

Typ: Zeichenfolge

Erforderlich: Nein

RamdiskId

Die ID des RAM-Datenträgers.

Typ: Zeichenfolge

Erforderlich: Nein

SecurityGroupIds

Die IDs der Sicherheitsgruppen für die Instance.

Typ: StringList

Erforderlich: Nein

SecurityGroups

Die Namen der Sicherheitsgruppen für die Instance.

Typ: StringList

Erforderlich: Nein

SubnetId

Die Subnetz-ID.

Typ: Zeichenfolge

Erforderlich: Nein

TagSpecifications

Die Tags, die beim Start auf die Ressourcen angewendet werden. Instances und Volumes können nur beim Start mit Tags versehen werden. Die angegebenen Tags werden auf alle Instances bzw. Volumes angewendet, die beim Start erstellt werden. Um eine Instance nach dem Start mit Tags zu versehen, verwenden Sie die Aktion `aws:createTags`— Erstelle Tags für AWS Ressourcen.

Typ: MapList (Weitere Informationen finden Sie unter TagSpecification.)

Erforderlich: Nein

UserData

Ein Skript, das als Zeichenfolgenliteralwert bereitgestellt wird. Wenn ein Literalwert eingegeben wird, muss er Base64-kodiert sein.

Typ: Zeichenfolge

Erforderlich: Nein

Output

InstanceIds

Die IDs der Instanzen.

InstanceStates

Der Status der Instance.

`aws:sleep` - Verzögerung einer Automatisierung

Verzögert eine Automatisierung um eine bestimmte Zeit. Diese Aktion verwendet das Datums- und Uhrzeitformat der International Organization for Standardization (ISO) 8601. Weitere Informationen zu diesem Datums- und Uhrzeitformat finden Sie unter ISO 8601.

Eingabe

Sie können eine Automatisierung um eine festgelegte Dauer verzögern.

YAML

```
name: sleep
action: aws:sleep
inputs:
  Duration: PT10M
```

JSON

```
{
```

```
   "name":"sleep",
   "action":"aws:sleep",
   "inputs":{
      "Duration":"PT10M"
   }
}
```

Sie können eine Automatisierung auch bis zu einem festgelegten Zeitpunkt verzögern. Wenn das Datum und die Uhrzeit verstrichen sind, erfolgt die Aktion unmittelbar.

YAML

```
name: sleep
action: aws:sleep
inputs:
  Timestamp: '2020-01-01T01:00:00Z'
```

JSON

```
{
    "name": "sleep",
    "action": "aws:sleep",
    "inputs": {
        "Timestamp": "2020-01-01T01:00:00Z"
    }
}
```

> **Note**
>
> Automation unterstützt eine maximale Verzögerung von 604799 Sekunden (7 Tage).

Dauer

Ein ISO 8601-Dauer. Sie können keine negative Dauer angeben.

Typ: Zeichenfolge

Erforderlich: Nein

Zeitstempel

Ein ISO 8601-Zeitstempel. Wenn Sie keinen Wert für diesen Parameter angeben, müssen Sie einen Wert für den `Duration`-Parameter angeben.

Typ: Zeichenfolge

Erforderlich: Nein

Output

Keine

`aws:updateVariable` – Aktualisiert einen Wert für eine Runbook-Variable

Diese Aktion aktualisiert einen Wert für eine Runbook-Variable. Der Datentyp des Werts muss dem Datentyp der Variable entsprechen, die Sie aktualisieren möchten. Datentypkonvertierungen werden nicht unterstützt. Die `onCancel`-Eigenschaft wird für die `aws:updateVariable`-Aktion nicht unterstützt.

Eingabe

Die Eingabe ist wie folgt.

YAML

```
name: updateStringList
action: aws:updateVariable
inputs:
    Name: variable:variable name
    Value:
    - "1"
    - "2"
```

JSON

```
{
    "name": "updateStringList",
    "action": "aws:updateVariable",
    "inputs": {
```

```
        "Name": "variable:variable name",
        "Value": ["1","2"]
    }
}
```

Name

Der Name der Variable, deren Wert Sie aktualisieren möchten. Sie müssen das Format `variable:variable name` verwenden

Typ: Zeichenfolge

Erforderlich: Ja

Wert

Der neue Wert, der der Variable zugewiesen werden soll. Der Wert muss mit dem Datentyp der Variable übereinstimmen. Datentypkonvertierungen werden nicht unterstützt.

Typ: Boolesch | Ganzzahl | MapList | Zeichenfolge | StringList | StringMap

Erforderlich: Ja

Einschränkungen:

- MapList kann eine maximale Anzahl von 200 Elementen enthalten.
- Schlüssellängen können eine Mindestlänge von 1 und eine Maximallänge von 50 haben.
- StringList kann eine Mindestanzahl von 0 Elementen und eine maximale Anzahl von 50 Elementen haben.
- Die Länge einer Zeichenfolge kann eine Mindestlänge von 1 und eine Maximallänge von 512 haben.

Output

Keine

`aws:waitForAwsResourceProperty`— Warte auf eine AWS Ressourceneigenschaft

Die `aws:waitForAwsResourceProperty`-Aktion erlaubt Ihrer Automatisierung auf einen bestimmten Ressourcenstatus oder Ereignisstatus zu warten, bevor Sie die Automatisierung

fortsetzen. Weitere Beispiele zur Verwendung dieser Aktion finden Sie unter Weitere Runbook-Beispiele.

> **Note**
>
> Der Standardwert für die Zeitüberschreitung für diese Aktion beträgt 3 600 Sekunden (eine Stunde). Sie können die Zeitüberschreitung über den Parameter `timeoutSeconds` für einen `aws:waitForAwsResourceProperty`-Schritt anpassen. Weitere Informationen und Beispiele zur Verwendung dieser Aktion finden Sie unter Behandeln von Timeouts in Runbooks.

> **Note**
>
> Die Aktion `aws:waitForAwsResourceProperty` unterstützt die automatische Drosselungswiederholung. Weitere Informationen finden Sie unter Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge.

Eingabe

Eingaben werden von der ausgewählten API-Operation bestimmt.

YAML

```
action: aws:waitForAwsResourceProperty
inputs:
  Service: The official namespace of the service
  Api: The API operation or method name
  API operation inputs or parameters: A value
  PropertySelector: Response object
  DesiredValues:
  - Desired property value
```

JSON

```
{
  "action": "aws:waitForAwsResourceProperty",
  "inputs": {
    "Service":"The official namespace of the service",
```

```
    "Api":"The API operation or method name",
    "API operation inputs or parameters":"A value",
    "PropertySelector": "Response object",
    "DesiredValues": [
      "Desired property value"
    ]
  }
}
```

Service

Der AWS-Service Namespace, der die API-Operation enthält, die Sie ausführen möchten. Der Namespace für ist beispielsweise. AWS Systems Manager `ssm` Der Namespace für Amazon Elastic Compute Cloud (Amazon EC2) lautet`ec2`. Eine Liste der unterstützten AWS-Service Namespaces finden Sie im Abschnitt Verfügbare Dienste der Befehlsreferenz.AWS CLI

Typ: Zeichenfolge

Erforderlich: Ja

Api

Der Name der API-Operation, die Sie ausführen möchten. Sie können die API-Operationen (auch als Methoden bezeichnet) anzeigen, indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise werden alle API-Vorgänge (Methoden) für Amazon Relational Database Service (Amazon RDS) auf der folgenden Seite aufgelistet: Amazon RDS-Methoden.

Typ: Zeichenfolge

Erforderlich: Ja

API-Operation-Eingaben

Eine oder mehrere API-Eingaben. Sie können die verfügbaren Eingaben (auch als Parameter bezeichnet) anzeigen, indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise sind alle Methoden für Amazon RDS auf der folgenden Seite aufgeführt: Amazon RDS-Methoden. Wählen Sie die Methode describe_db_instances und scrollen Sie nach unten, um die verfügbaren Parameter wie Identifier, Name und Values zu sehen. DBInstance

YAML

```
inputs:
  Service: The official namespace of the service
  Api: The API operation name
  API input 1: A value
  API Input 2: A value
  API Input 3: A value
```

JSON

```
"inputs":{
      "Service":"The official namespace of the service",
      "Api":"The API operation name",
      "API input 1":"A value",
      "API Input 2":"A value",
      "API Input 3":"A value"
}
```

Typ: Abhängig von der gewählten API-Operation

Erforderlich: Ja

PropertySelector

Das JSONPath zu einem bestimmten Attribut im Antwortobjekt. Sie können die Antwortobjekte anzeigen indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise sind alle Methoden für Amazon RDS auf der folgenden Seite aufgeführt: Amazon RDS-Methoden. Wählen Sie die Methode describe_db_instances und scrollen Sie nach unten zum Abschnitt Antwortstruktur. DBInstancesist als Antwortobjekt aufgeführt.

Typ: Zeichenfolge

Erforderlich: Ja

DesiredValues

Die erwartete Status oder Zustand, bei dem die Automatisierung fortgesetzt werden soll.

Typ: MapList, StringList

Erforderlich: Ja

Systemvariablen für Automation

AWS Systems Manager-Automation-Runbooks verwenden die folgenden Variablen. Ein Beispiel für die Verwendung dieser Variablen erhalten Sie, wenn Sie die JSON-Quelle des `AWS-UpdateWindowsAmi`-Runbooks anzeigen.

So zeigen Sie die JSON-Quelle des **`AWS-UpdateWindowsAmi`**-Runbooks an

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie in der Dokumentliste entweder über die Suchleiste oder die Zahlen rechts neben der Suchleiste das Runbook **`AWS-UpdateWindowsAmi`** aus.
4. Wählen Sie die Registerkarte Content aus.

Systemvariablen

Automation-Runbooks unterstützen die folgenden Variablen.

Variable	Details
`global:ACCOUNT_ID`	Die AWS-Konto-ID des Benutzers oder der Rolle, in dem die Automatisierung ausgeführt wird.
`global:DATE`	Das Datum (zur Ausführungszeit) im Format yyyy-MM-dd.
`global:DATE_TIME`	Das Datum und die Uhrzeit (zur Ausführun gszeit) im Format yyyy-MM-dd_HH.mm.ss.
`global:AWS_PARTITION`	Die Partition, in der sich die Ressource befindet. Für Standard-AWS-Regionen lautet die Partition `aws`. Für Ressourcen in anderen Partitionen lautet die Partition `aws-`*`partition name`* . Beispielsweise ist die Partition für Ressourcen in der AWS-GovCloud (US-West)-Region `aws-us-gov` .

Variable	Details
`global:REGION`	Die Region, in der das Runbook ausgeführt wird. Beispiel: us-east-2.

Variablen für Automation

Automation-Runbooks unterstützen die folgenden Automatisierungsvariablen.

Variable	Details
`automation:EXECUTION_ID`	Die eindeutige ID, die der aktuellen Automatisierung zugewiesen ist. Beispiel, `1a2b3c-1a2b3c-1a2b3c-1a2b3c1a2b3c1a2b3c` .

Themen

- Terminologie
- Unterstützte Szenarien
- Nicht unterstützte Szenarien

Terminologie

Die folgenden Bedingungen beschreiben, wie Variablen und Parameter gelöst werden.

Begriff	Definition	Beispiel
Konstanter ARN	Ein gültiger Amazon-Ressourcenname (ARN) ohne Variablen.	`arn:aws:iam::123456789012:role/roleName`
Runbook-Parameter	Ein auf der Runbook-Ebene definierter Parameter (z. B. `instanceId`). Der Parameter wird in einer grundlegenden Zeichenfo	`{ "description": "Create Image Demo", "version": "0.3",`

Begriff	Definition	Beispiel
	lgenersetzung verwendet. Sein Wert wird zur Startausf ührungszeit bereitgestellt.	<pre>"assumeRole": "*Your_Automation_Assume_Role_ARN* ", "parameters":{ **"instanceId": { "type": "String", "description": "Instance to create image from"** } }</pre>

Begriff	Definition	Beispiel
Systemvariable	Eine allgemeine Variable, die in das Runbook eingefügt wird, wenn ein beliebiger Teil des Runbooks bewertet wird.	`"activities": [{ "id": "copyImage", "activityType": "AWS-CopyImage", "maxAttempts": 1, "onFailure": "Continue", "inputs": { "ImageName": "{{imageName}}", "SourceImageId": "{{sourceImageId}}", "SourceRegion": "{{sourceRegion}}", "Encrypted": true, "ImageDescription": "Test CopyImage Description created on {{global:DATE}} " } }]`

Begriff	Definition	Beispiel

Variable für Automation	Eine Variable, die sich auf die Automatisierung bezieht, die in das Runbook eingefügt wird, wenn ein Teil des Runbooks bewertet wird.	{
		"name": "runFixedCmds",
		"action": "aws:runCommand",
		"maxAttempts": 1,
		"onFailure": "Continue",
		"inputs": {
		"DocumentName": "AWS-RunPowerShellScript",
		"InstanceIds": [
		"{{LaunchInstance.InstanceIds}}"
		],
		"Parameters": {
		"commands": [
		"dir",
		"date",
		"“{{outputFormat}}” -f “left”,”right”,”{{global:DATE}}”,” **{{automation:EXECUTION_ID}}** ”
		]
		}
		}
		}

Begriff	Definition	Beispiel

Systems Manager-Parameter	Eine in AWS Systems Manager Parameter Store definierte Variable. Eine direkte Referenzierung in der Schritteingabe ist nicht möglich. Eventuell sind für den Zugriff auf den Parameter Berechtigungen erforderlich.	<pre>description: Launch new Windows test instance
		schemaVersion: '0.3'
		assumeRole: '{{AutomationAssumeRole}}'
		parameters:
		  AutomationAssumeRole:
		    type: String
		    default: ''
		    description: >-
		      (Required) The ARN of the role that allows Automation to perform the
		      actions on your behalf. If no role is specified, Systems Manager
		      Automation uses your IAM permissions to run this runbook.
		  LatestAmi:
		    type: String
		    default: >-
		      {{ssm:/aws/service/ami-windows-latest/Windows_Server-2016-English-Full-Base}}
		    description: The latest Windows Server 2016 AMI queried from the public parameter.
		mainSteps:
		  - name: launchInstance
		    action: 'aws:runInstances'
		    maxAttempts: 3</pre>

Begriff	Definition	Beispiel
		timeoutSeconds: 1200 onFailure: Abort inputs: ImageId: '{{LatestAmi}}' ...

Unterstützte Szenarien

Szenario	Kommentare	Beispiel
Konstanter ARN `assumeRole` beim Erstellen.	Es wird eine Autorisierungsprüfung durchgeführt, um zu bestätigen, dass der aufrufende Benutzer über die Berechtigung zum Übergeben der Rolle `assumeRole` verfügt.	{ "description": "Test all Automation resolvable parameters", "schemaVersion": "0.3", "assumeRole": ***"arn:aws:iam::123456789012:role/roleName"*** , "parameters": { ...
Der Runbook-Parameter wird für `AssumeRole` bereitgestellt, wenn die Automatisierung gestartet wird.	Muss in der Parameterliste des Runbooks definiert werden.	{ "description": "Test all Automation resolvable parameters", "schemaVersion": "0.3", "assumeRole": **"{{dynamicARN}}"** , "parameters": { ...

Szenario	Kommentare	Beispiel

Für Runbookparameter beim Start bereitgestellter Wert.	Der Kunde stellt den für einen Parameter zu verwendenden Wert bereit. Alle zur bereitgestellten Eingaben müssen in der Parameterliste des Runbooks definiert sein.	<pre>...
		"parameters": {
		    "amiId": {
		      "type": "String",
		      "default":
		"ami-12345678 ",
		      "description":
		"list of commands to
		run as part of first
		step"
		    },
		...</pre> Eingaben zum Start der Automation-Ausführung umfassen : `{"amiId" : ["ami-12345678 "] }`

Szenario	Kommentare	Beispiel

Systems Manager Parameter , auf den im Runbook-Inhalt verwiesen wird.	Die Variable existiert im Kundenkonto oder ist ein öffentlich zugänglic her Parameter und die `AssumeRole` für das Runbook hat Zugriff auf die Variable. Beim Erstellen wird eine Überprüfung durchgefü hrt, um zu bestätigen, dass `AssumeRole` Zugriff hat. Der Parameter kann nicht direkt in der Schritteingabe referenziert werden.	```
		...
		parameters:
		    LatestAmi:
		    type: String
		    default: >-
		      {{ssm:/aws/
		service/ami-wind
		ows-latest/Windows
		_Server-2016-English-
		Full-Base}}
		    description: The
		 latest Windows Server
		 2016 AMI queried from
		 the public parameter.
		mainSteps:
		  - name: launchIns
		tance
		    action: 'aws:runI
		nstances'
		    maxAttempts: 3
		    timeoutSeconds:
		 1200
		    onFailure: Abort
		    inputs:
		      ImageId: '{{Latest
		Ami}}'
		...
		``

Szenario	Kommentare	Beispiel

Die Systemvariable, auf die in der Definition des Schritts verwiesen wird	Eine Systemvariable wird beim Start der Automatisierung in das Runbook eingefügt. Der in das Runbook eingefügte Wert steht in Relation zum Zeitpunkt des Einfügens. Das bedeutet, dass der Wert einer Zeitvariable, die in Schritt 1 eingefügt wurde, aufgrund der erforderlichen Zeit für die Ausführung der Schritte vom in Schritt 3 eingefügten Wert abweicht. Systemvariablen müssen nicht in der Parameter liste des Runbooks festgelegt werden.	```
		...
		  "mainSteps": [
		    {
		      "name": "RunSomeC
		ommands",
		      "action":
		 "aws:runCommand",
		      "maxAttempts": 1,
		      "onFailure":
		 "Continue",
		      "inputs": {
		        "DocumentName":
		 "AWS:RunPowerShell",
		        "InstanceIds":
		 ["{{LaunchInstance
		.InstanceIds}}"],
		        "Parameters": {
		            "commands
		" : [
		                "echo
		 {The time is now
		 {{global:DATE_TIME
		}}}"
		            ]
		        }
		    }
		}, ...
		``

Szenario	Kommentare	Beispiel

Die Automation-Variable, auf die in der Definition des Schritts verwiesen wird.	Automation-Variablen müssen nicht in der Parameterliste des Runbooks festgelegt werden. Die einzige unterstützte AAutomation-Variable ist automation:EXECUTION_ID.	<pre>...
		"mainSteps": [
		    {
		      "name": "invokeLambdaFunction",
		      "action": "aws:invokeLambdaFunction",
		      "maxAttempts": 1,
		      "onFailure": "Continue",
		      "inputs": {
		        "FunctionName": "Hello-World-LambdaFunction",
		
		"Payload" : "{ "executionId" : "{{automation:EXECUTION_ID}}" }"
		      }
		    }
		...</pre>

Szenario	Kommentare	Beispiel

Weitere Informationen finden Sie in der Ausgabe des vorherigen Schritts in der Definition des nächsten Schritts.	Dies ist die Parameter umleitung. Mithilfe der Syntax `{{stepName.OutputName}}` wird auf die Ausgabe eines vorherigen Schritts verwiesen. Diese Syntax kann vom Kunden nicht für Runbookparameter verwendet werden. Dies wird behoben, wenn der verweisende Schritt ausgeführt wird. Der Parameter ist nicht in der Liste der Parameter des Runbooks aufgeführt.	<pre>...
		"mainSteps": [
		    {
		      "name": "LaunchInstance",
		      "action": "aws:runInstances",
		      "maxAttempts": 1,
		      "onFailure": "Continue",
		      "inputs": {
		        "ImageId": "{{amiId}}",
		        "MinInstanceCount": 1,
		        "MaxInstanceCount": 2
		      }
		    },
		    {
		      "name":"changeState",
		      "action": "aws:changeInstanceState",
		      "maxAttempts": 1,
		      "onFailure": "Continue",
		      "inputs": {
		        "InstanceIds": ["{{LaunchInstance.InstanceIds}}"],
		        "DesiredState": "terminated"
		      }
		    }
		
		...</pre>

Nicht unterstützte Szenarien

Szenario	Kommentar	Beispiel

Systems Manager Parameter bereitgestellt für `assumeRole` beim Erstellen	Nicht unterstützt	`...`
		
		`{`
		`  "description": "Test all Automation resolvable parameters",`
		`  "schemaVersion": "0.3",`
		`  "assumeRole": "{{ssm:administratorRoleARN}}  ",`
		`  "parameters": {`
		
		`...`

System Manager-Parameter, der direkt in der Schritteingabe referenziert wird.	Gibt eine `InvalidDocumentContent` - Ausnahme zur Erstellungszeit zurück.	`...`
		`mainSteps:`
		`  - name: launchInstance`
		`    action: 'aws:runInstances'`
		`    maxAttempts: 3`
		`    timeoutSeconds: 1200`
		`    onFailure: Abort`
		`    inputs:`
		`      ImageId: '{{ssm:/aws/service/ami-windows-latest/Windows_Server-2016-English-Full-Base}}'`
		`...`

Szenario	Kommentar	Beispiel
Variablenschrittdefinition	Die Definition eines Schritts im Runbook wird anhand von Variablen zusammengestellt.	... "mainSteps": [{ "name": "LaunchInstance", "action": "aws:runInstances", "**{{attemptModel}}** ": 1, "onFailure": "Continue", "inputs": { "ImageId": "*ami-12345678* ", "MinInstanceCount": 1, "MaxInstanceCount": 2 } ... User supplies input : { "attemptModel" : "**minAttempts** " }

Szenario	Kommentar	Beispiel
Querverweise auf Runbook-Parameter	Der Benutzer liefert zur Startzeit einen Eingabeparameter, der ein Verweis auf einen anderen Parameter im Runbook ist.	... "parameters": { "amiId": { "type": "String", "default": "ami-7f2e6015 ", "description": "list of commands to run as part of first step" }, "alternateAmiId": { "type": "String", "description": "The alternate AMI to try if this first fails". "default" : "{{amiId}}" }, ...

Szenario	Kommentar	Beispiel
Multi-Level-Expansion	Das Runbook definiert eine Variable, die den Namen einer Variablen ergibt. Dieser befindet sich in den Variablen trennzeichen (d. h. {{ }}) und wird auf den Wert dieser Variable/dieses Parameters erweitert.	... "parameters": { "*firstParameter* ": { "type": "String", "default": "param2", "description": "The parameter to reference" }, "*secondParameter* ": { "type": "String", "default" : "echo {Hello world}", "description": "What to run" } }, "mainSteps": [{ "name": "runFixedCmds", "action": "aws:runCommand", "maxAttempts": 1, "onFailure": "Continue", "inputs": { "DocumentName": "AWS-RunPowerShellScript", "InstanceIds" : "{{LaunchInstance.InstanceIds}}", "Parameters": { "commands": ["{{ ***{{firstParameter}}*** }}"] }

Szenario	Kommentar	Beispiel
		``` ... Note: The customer intention here would be to run a command of "echo {Hello world}" ```

Szenario	Kommentar	Beispiel

Verweis auf die Ausgabe aus einem Runbook-Schritt, bei dem es sich um einen anderen Variablentyp handelt	Der Benutzer verweist auf die Ausgabe eines vorherige n Runbook-Schritts in einem späteren Schritt. Die Ausgabe ist ein Variablentyp, der nicht den Anforderungen der Aktion des nachfolgenden Schritts erfüllt.	```
		...
		mainSteps:
		- name: getImageId
		  action: aws:execu
		teAwsApi
		  inputs:
		    Service: ec2
		    Api: DescribeImages
		    Filters:
		    - Name: "name"
		      Values:
		      - "{{ImageName}}"
		  outputs:
		  - Name: ImageIdList
		    Selector: "$.Images
		"
		    Type: "StringList"
		- name: copyMyImages
		  action: aws:copyImage
		  maxAttempts: 3
		  onFailure: Abort
		  inputs:
		    SourceImageId:
		 {{getImageId.Image
		IdList}}
		    SourceRegion: ap-
		northeast-2
		    ImageName:
		 Encrypted Copies of
		 LAMP base AMI in ap-
		northeast-2
		    Encrypted: true
		...
		Note: You must provide
		 the type required
		 by the Automation
		 action.
		In this case, aws:copyI
		mage requires a
		 "String" type variable
		 but the preceding step
		``

Szenario	Kommentar	Beispiel
		`outputs a "StringList" type variable.`

Erstellen Ihrer eigenen Runbooks

Ein Automatisierungs-Runbook definiert die Aktionen, die Systems Manager auf Ihren verwalteten Instanzen und anderen AWS Ressourcen ausführt, wenn eine Automatisierung ausgeführt wird. Automation ist ein Tool in AWS Systems Manager. Ein Runbook enthält einen oder mehrere Schritte, die in sequentieller Reihenfolge ausgeführt werden. Jeder Schritt basiert auf einer einzigen Aktion. Die Ausgabe eines Schritts kann als Eingabe in einem späteren Schritt verwendet werden.

Der Prozess der Ausführung dieser Aktionen und ihrer Schritte wird als Automatisierung bezeichnet.

Mit den für Runbooks unterstützten Aktionstypen können Sie eine Vielzahl von Vorgängen in Ihrer AWS Umgebung automatisieren. Mithilfe des `executeScript` Aktionstyps können Sie beispielsweise eine Python oder ein PowerShell Skript direkt in Ihr Runbook einbetten. (Wenn Sie ein benutzerdefiniertes Runbook erstellen, können Sie Ihr Skript inline hinzufügen oder es von einem S3-Bucket oder von Ihrem lokalen Computer aus anhängen.) Sie können die Verwaltung Ihrer AWS CloudFormation Ressourcen automatisieren, indem Sie die `deleteStack` Aktionstypen `createStack` und verwenden. Darüber hinaus kann ein Schritt mithilfe des `executeAwsApi` Aktionstyps jede beliebige API-Operation ausführen AWS-Service, z. B. das Erstellen oder Löschen von AWS Ressourcen, das Starten anderer Prozesse, das Initiieren von Benachrichtigungen und vieles mehr.

Eine Liste aller 20 unterstützten Aktionstypen für Automation finden Sie unter Systems Manager Automation Aktionen-Referenz.

AWS Systems Manager Automation bietet mehrere Runbooks mit vordefinierten Schritten, mit denen Sie allgemeine Aufgaben wie den Neustart einer oder mehrerer Amazon Elastic Compute Cloud (Amazon EC2) -Instances oder das Erstellen einer Amazon Machine Image () ausführen können. AMI Sie können auch Ihre eigenen Runbooks erstellen und sie mit anderen AWS-Konten teilen oder sie für alle Automation-Benutzer veröffentlichen.

Runbooks werden mit YAML oder JSON geschrieben. Mit dem Document Builder in der Systems Manager-Automation-Konsole können Sie jedoch ein Runbook erstellen, ohne nativen JSON- oder YAML-Code erstellen zu müssen.

> **Important**
>
> Wenn Sie einen automatisierten Workflow ausführen, der andere Services mithilfe einer AWS Identity and Access Management -(IAM)-Servicerolle aufruft, muss die Servicerolle mit der Berechtigung zum Aufrufen dieser Services konfiguriert sein. Diese Anforderung gilt für alle AWS Automatisierungs-Runbooks (`AWS-*`Runbooks) wie die`AWS-ConfigureS3BucketLogging`, und `AWS-RestartEC2Instance` Runbooks`AWS-CreateDynamoDBBackup`, um nur einige zu nennen. Diese Anforderung gilt auch für alle benutzerdefinierten Automatisierungs-Runbooks, die Sie erstellen und die andere mithilfe AWS-Services von Aktionen aufrufen, die andere Dienste aufrufen. Wenn Sie unter anderem `aws:executeAwsApi`-, `aws:createStack`- oder `aws:copyImage`-Aktionen verwenden, konfigurieren Sie die Dienstrolle mit der Berechtigung zum Aufrufen solcher Services.
> Sie können anderen AWS-Services Berechtigungen erteilen, indem Sie der Rolle eine eingebundene IAM-Richtlinie hinzufügen. Weitere Informationen finden Sie unter (Optional) Fügen Sie eine Inline-Automatisierungsrichtlinie oder eine vom Kunden verwaltete Richtlinie hinzu, um andere aufzurufen AWS-Services.

Informationen zu den Aktionen, die Sie in einem Runbook angeben können, finden Sie unter Systems Manager Automation Aktionen-Referenz.

Informationen zur Verwendung von AWS Toolkit for Visual Studio Code zum Erstellen von Runbooks finden Sie unter Arbeiten mit Systems Manager Automation-Dokumenten im AWS Toolkit for Visual Studio Code Benutzerhandbuch.

Informationen zur Verwendung des visuellen Designers zum Erstellen eines benutzerdefinierten Runbooks finden Sie unter Visuelle Designerfahrung für Automation-Runbooks.

Inhalt

- Visuelle Designerfahrung für Automation-Runbooks
  - Überblick über die Benutzeroberfläche für visuelle Designerfahrung
    - Aktionsbrowser
    - Leinwand

    - Formular
    - Tastenkombination
  - Die visuelle Designerfahrung nutzen
    - Einen Runbook-Workflow erstellen
    - Ein Runbook entwerfen
    - Ihr Runbook aktualisieren
    - Ihr Runbook exportieren
  - Konfigurieren von Eingaben und Ausgaben für Ihre Aktionen
    - Eingabedaten für eine Aktion angeben
    - Die Ausgabedaten für eine Aktion definieren
  - Fehlerbehandlung bei der visuellen Designerfahrung
    - Bei einem Fehler die Aktion erneut versuchen
    - Timeouts
    - Fehlgeschlagene Aktionen
    - Abgebrochene Aktionen
    - Kritische Aktionen
    - Aktionen beenden
  - Tutorial: Ein Runbook mithilfe der visuellen Designerfahrung erstellen
    - Schritt 1: Zur visuellen Designerfahrung navigieren
    - Schritt 2: Einen Workflow erstellen
    - Schritt 3: Den automatisch generierten Code überprüfen
    - Schritt 4: Ihr neues Runbook ausführen
    - Schritt 5: Bereinigen
- Erstellen von Automation-Runbooks
  - Identifizieren Sie Ihren Anwendungsfall
  - Einrichten Ihrer Entwicklungsumgebung
  - Entwickeln von Runbook-Inhalten
  - Beispiel 1: Erstellen von über- und untergeordneten Runbooks
    - Erstellen des untergeordneten Runbooks
    - Erstellen des übergeordneten Runbooks

- Beispiel 2: Skriptbasiertes Runbook
- Weitere Runbook-Beispiele
  - Bereitstellung der VPC-Architektur und der Microsoft Active Directory-Domänencontroller
  - Wiederherstellen eines Root-Volumes aus dem letzten Snapshot
  - Erstellen eines AMI und einer regionenübergreifenden Kopie
- Erstellen von Eingabeparametern, die AWS-Ressourcen ausfüllen
- Verwenden von Document Builder zur Erstellung von Runbooks
  - Erstellen eines Runbooks mithilfe von Document Builder
  - Erstellen eines Runbooks, das Skripte ausführt
- Verwenden von Skripten in Runbooks
  - Berechtigungen für die Verwendung von Runbooks
  - Hinzufügen von Skripts zu Runbooks
  - Skripteinschränkungen für Runbooks
- Verwendung bedingter Anweisungen in Runbooks
  - Arbeiten mit der aws:branch-Aktion
    - Erstellen eines aws:branch-Schritts in einem Runbook
      - Informationen zum Erstellen der Ausgabevariable
    - Beispiel aws:branch-Runbooks
    - Erstellen komplexer verzweigender Automatisierungen mit Operatoren
  - Beispiele für die Verwendung von bedingten Optionen
- Verwenden von Aktionsausgaben als Eingaben
  - Verwendung JSONPath in Runbooks
- Erstellen von Webhook-Integrationen für Automation
  - Erstellen von Integrationen (Konsole)
  - Erstellen von Integrationen (Befehlszeile)
  - Erstellen von Webhooks für Integrationen
- Behandeln von Timeouts in Runbooks

Visuelle Designerfahrung für Automation-Runbooks

AWS Systems Manager Automation bietet ein visuelles Designerlebnis mit geringem Code-Aufwand, mit dem Sie Automatisierungs-Runbooks erstellen können. Das visuelle Designerlebnis bietet eine drag-and-drop Benutzeroberfläche mit der Option, Ihren eigenen Code hinzuzufügen, sodass Sie Runbooks einfacher erstellen und bearbeiten können. Mit der visuellen Designerfahrung können Sie Folgendes tun:

- Steuern Sie bedingte Anweisungen.
- Steuern Sie, wie Eingabe und Ausgabe für jede Aktion gefiltert oder transformiert werden.
- Konfigurieren Sie die Fehlerbehandlung.
- Erstellen Sie Prototypen für neue Runbooks.
- Verwenden Sie Ihre Prototyp-Runbooks als Ausgangspunkt für die lokale Entwicklung mit AWS Toolkit for Visual Studio Code.

Wenn Sie ein Runbook erstellen oder bearbeiten, können Sie über die Automation-Konsole auf die visuelle Designerfahrung zugreifen. Wenn Sie ein Runbook erstellen, überprüft die visuelle Designerfahrung Ihre Arbeit und generiert automatisch Code. Sie können den generierten Code überprüfen oder ihn für die lokale Entwicklung exportieren. Wenn Sie fertig sind, können Sie Ihr Runbook speichern, ausführen und die Ergebnisse in der Systems-Manager-Automation-Konsole überprüfen.

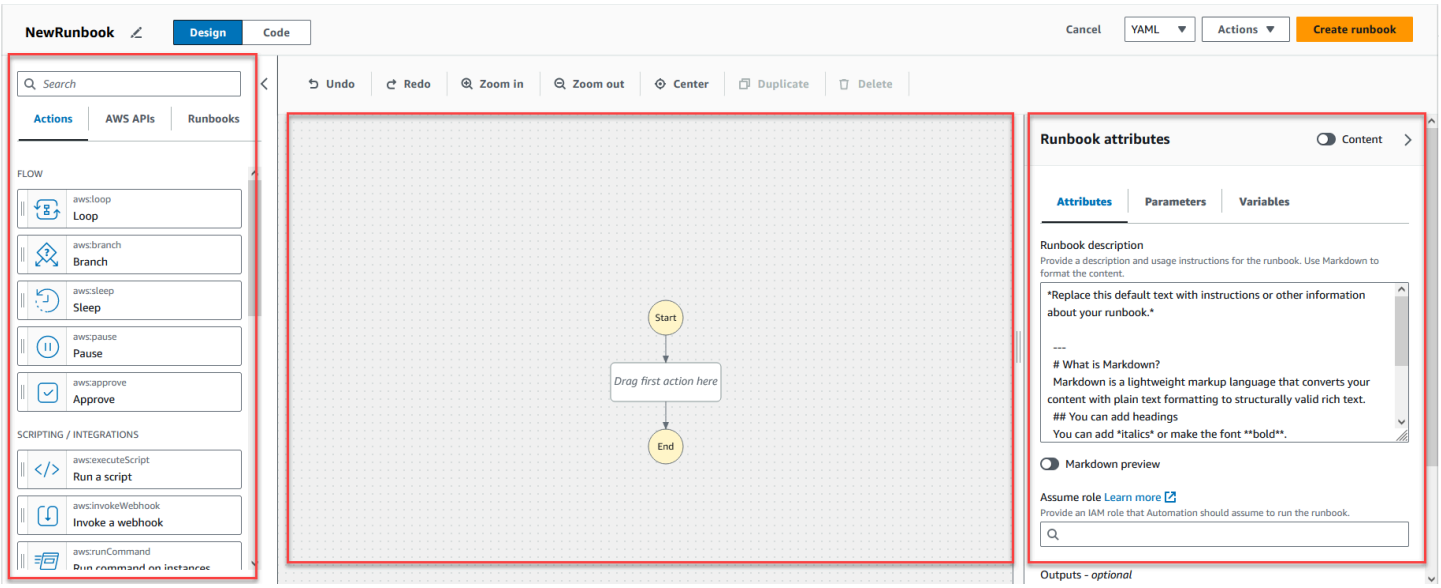
Themen

- Überblick über die Benutzeroberfläche für visuelle Designerfahrung
- Die visuelle Designerfahrung nutzen
- Konfigurieren von Eingaben und Ausgaben für Ihre Aktionen
- Fehlerbehandlung bei der visuellen Designerfahrung
- Tutorial: Ein Runbook mithilfe der visuellen Designerfahrung erstellen

Überblick über die Benutzeroberfläche für visuelle Designerfahrung

Die visuelle Designerfahrung für Systems Manager Automation ist ein visueller Workflow-Designer mit geringem Code-Aufwand, mit dem Sie Automation-Runbooks erstellen können.

Lernen Sie die visuelle Designerfahrung anhand eines Überblicks über die Komponenten der Benutzeroberfläche kennen:



- Der Aktionsbrowser enthält die Registerkarten Aktionen, AWS-APIs und Runbooks.
- Auf der Arbeitsfläche können Sie Aktionen per Drag-and-Drop in Ihr Workflow-Diagramm ziehen, die Reihenfolge der Aktionen ändern und Aktionen auswählen, die konfiguriert oder angezeigt werden sollen.
- Im Formularfenster können Sie die Eigenschaften jeder Aktion, die Sie auf der Arbeitsfläche ausgewählt haben, anzeigen und bearbeiten. Wählen Sie den Schalter Inhalt, um die YAML- oder JSON-Daten für Ihr Runbook anzuzeigen, wobei die aktuell ausgewählte Aktion hervorgehoben ist.

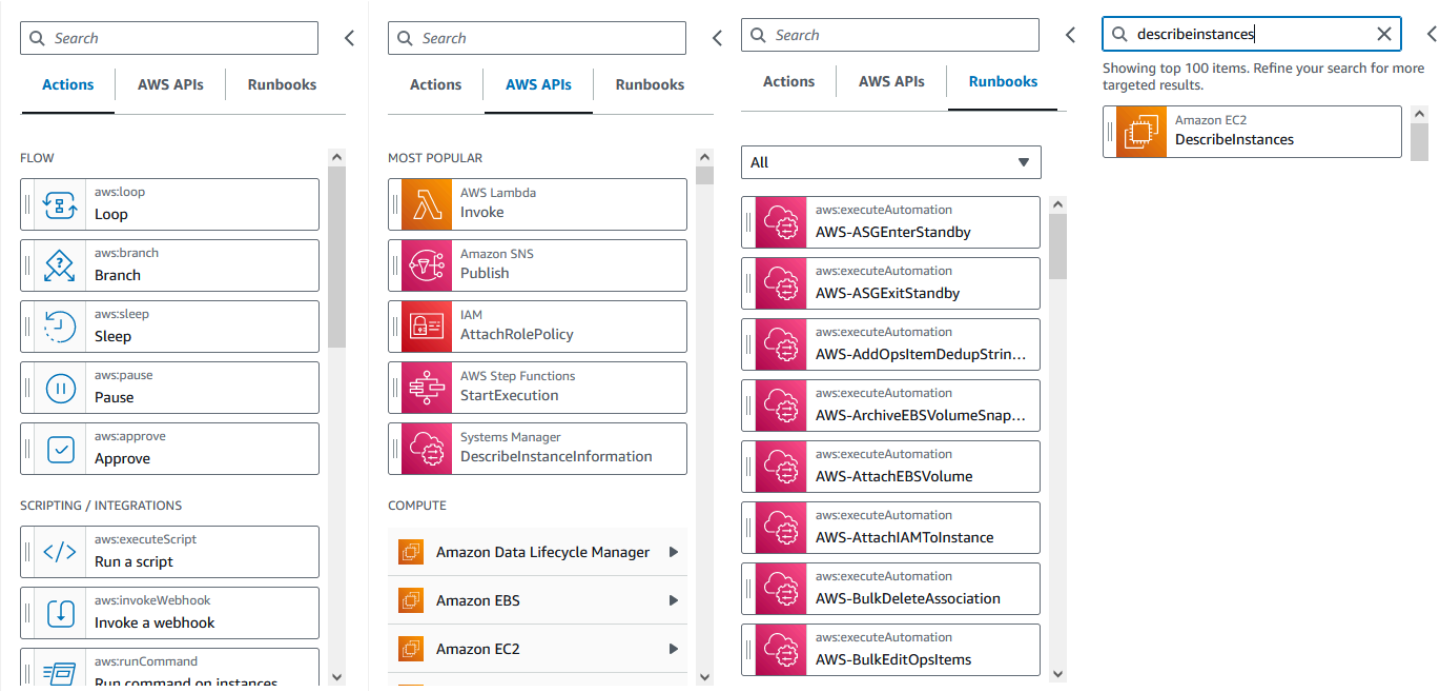
Mit Informationslinks wird ein Fenster mit Kontextinformationen geöffnet, falls Sie Hilfe benötigen. Diese Bereiche enthalten auch Links zu verwandten Themen in der Systems-Manager-Automation-Dokumentation.

Aktionsbrowser

Im Aktionsbrowser können Sie Aktionen auswählen, die Sie per Drag-and-Drop in Ihr Workflow-Diagramm ziehen möchten. Mit dem Suchfeld oben im Aktionsbrowser können Sie nach allen Aktionen suchen. Der Aktionsbrowser enthält die folgenden Registerkarten:

- Die Registerkarte Aktionen enthält eine Liste von Automatisierungs-Aktionen, die Sie per Drag-and-Drop in das Workflow-Diagramm Ihres Runbooks auf dem Workflow ziehen können.
- Die Registerkarte AWS-APIs enthält eine Liste von AWS-APIs, die Sie per Drag-and-Drop in das Workflow-Diagramm Ihres Runbooks auf der Arbeitsfläche ziehen können.

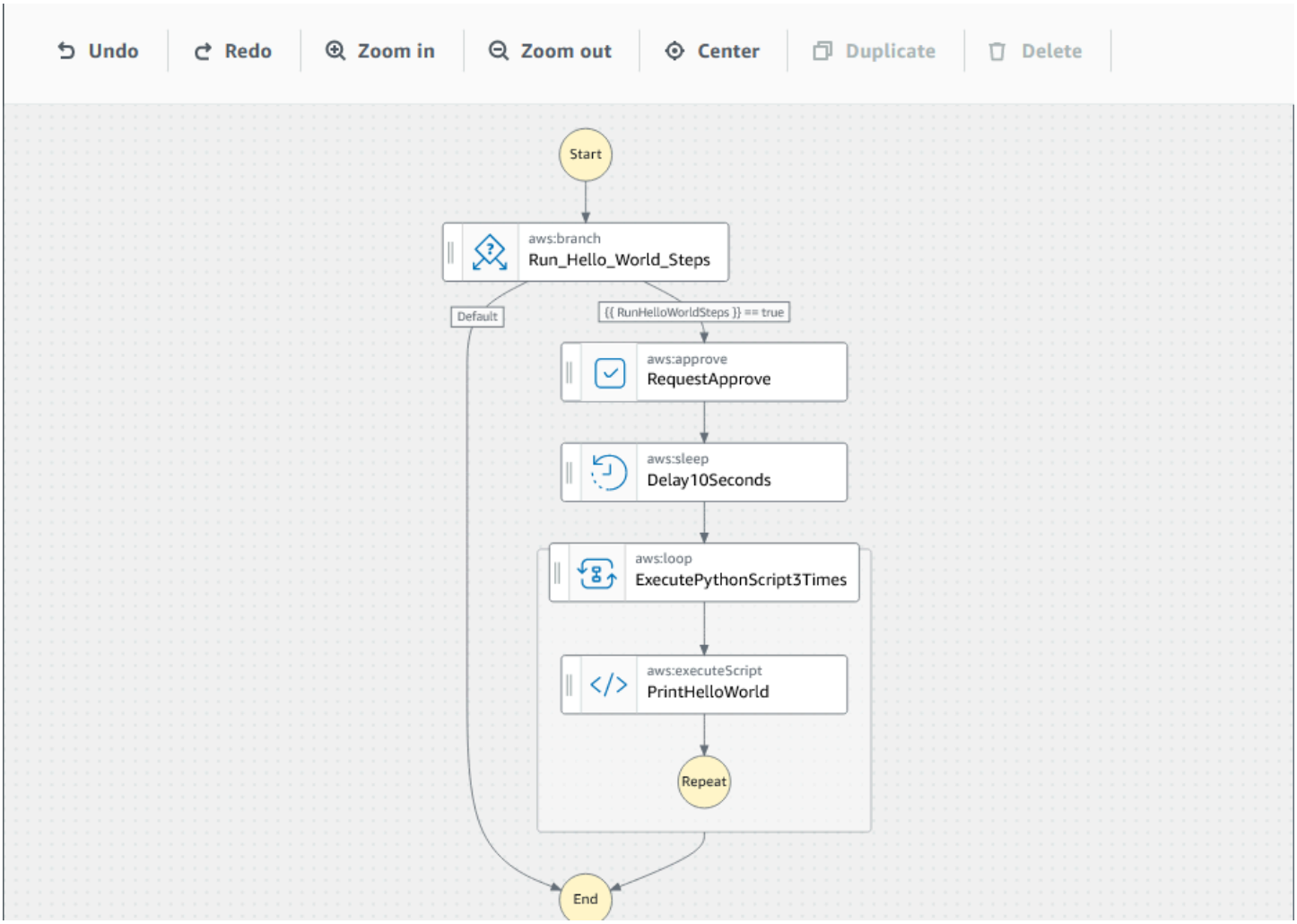
- Die Registerkarte Runbooks enthält mehrere gebrauchsfertige, wiederverwendbare Runbooks als Bausteine, die Sie für eine Vielzahl von Anwendungsfällen verwenden können. Beispielsweise können Sie Runbooks verwenden, um allgemeine Behebungsaufgaben für Amazon-EC2-Instances in Ihrem Workflow durchzuführen, ohne dieselben Aktionen erneut erstellen zu müssen.



Leinwand

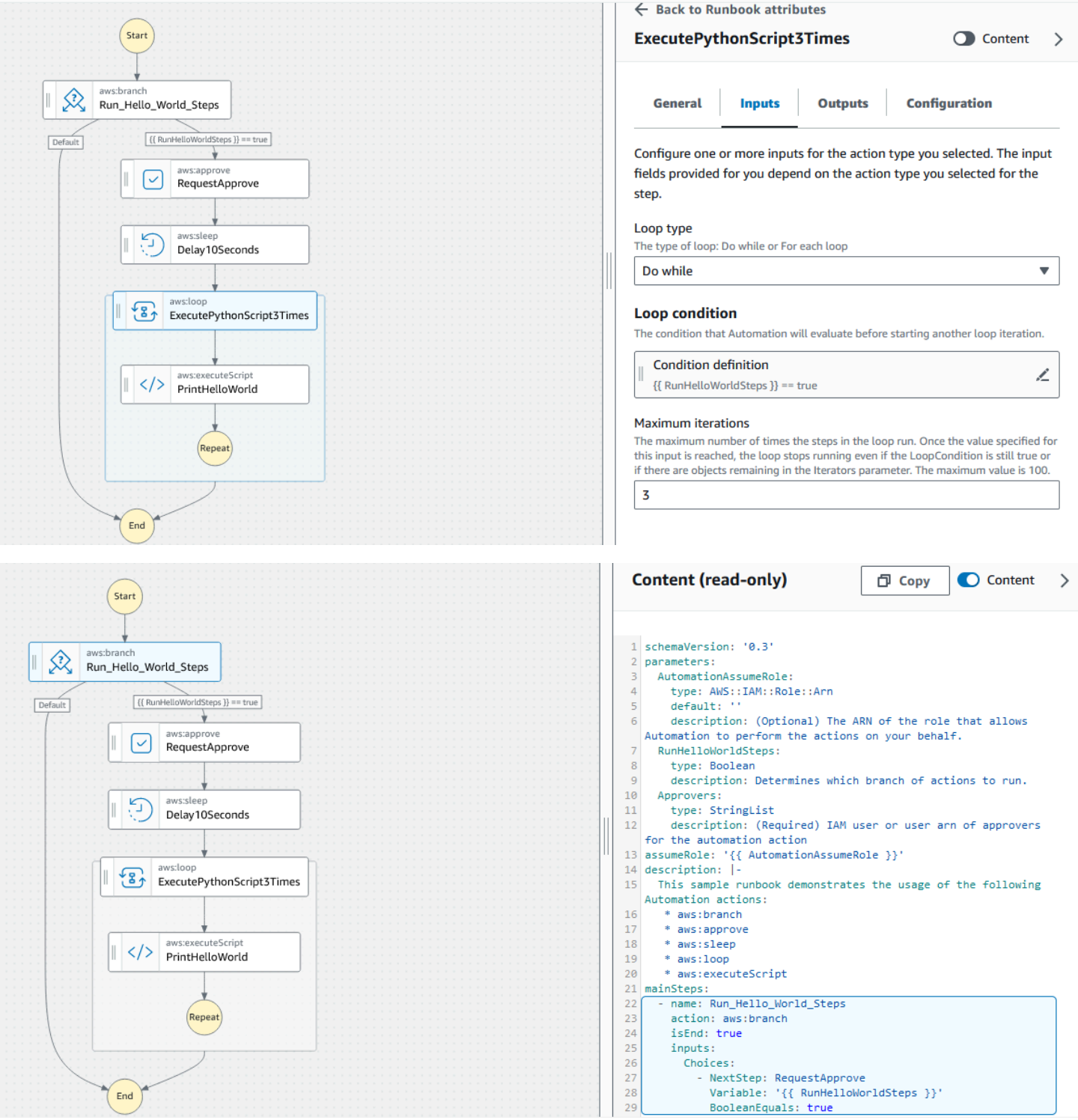
Nachdem Sie eine Aktion ausgewählt haben, die Sie zu Ihrer Automatisierung hinzufügen möchten, ziehen Sie sie auf den Workflow und legen Sie sie in Ihr Workflow-Diagramm ab. Sie können Aktionen auch per Drag-and-Drop an verschiedene Stellen im Workflow Ihres Runbooks verschieben. Wenn Ihr Workflow komplex ist, können Sie ihn möglicherweise nicht vollständig auf der Arbeitsfläche anzeigen. Verwenden Sie die Steuerelemente oben auf der Arbeitsfläche, um die Ansicht zu vergrößern oder zu verkleinern. Um verschiedene Teile eines Workflows anzuzeigen, können Sie das Workflow-Diagramm auf die Arbeitsfläche ziehen.

Ziehen Sie eine Aktion aus dem Browser Aktionen und legen Sie sie in das Workflow-Diagramm Ihres Runbooks ab. Eine Linie zeigt, wo sie in Ihrem Workflow platziert wird. Um die Reihenfolge einer Aktion zu ändern, können Sie sie an eine andere Stelle in Ihrem Workflow ziehen. Die neue Aktion wurde zu Ihrem Workflow hinzugefügt und ihr Code wird automatisch generiert.



Formular

Nachdem Sie Ihrem Runbook-Workflow eine Aktion hinzugefügt haben, können Sie sie so konfigurieren, dass sie Ihrem Anwendungsfall entspricht. Wählen Sie die Aktion aus, die Sie konfigurieren möchten, und die zugehörigen Parameter und Optionen werden im Formular-Bereich angezeigt. Sie können den YAML- oder JSON-Code auch sehen, indem Sie den Schalter Inhalt auswählen. Der Code, der von Ihnen ausgewählten Aktion zugeordnet ist, hervorgehoben.



Tastenkombination

Die visuelle Designerfahrung unterstützt die in der folgenden Tabelle aufgeführten Tastenkombinationen.

Tastaturkürzel	Verwenden Sie Tastaturkürzel, um Aktionen auszuführen.
Strg+Z	Machen Sie den letzten Vorgang rückgängig.
Strg+Y	Wiederholen Sie den letzten Vorgang.
Zentrieren	Zentrieren Sie den Workflow auf der Arbeitsfläche.
Backspace	Löschen Sie alle ausgewählten Zustände.
Entfernen	Entfernen Sie

Tastaturkürzel	alle ausgewählten Zustände.
Strg + D	Sie den ausgewählten Zustand.

Die visuelle Designerfahrung nutzen

Erfahren Sie, wie Sie Runbook-Workflows mithilfe der visuellen Designerfahrung erstellen, bearbeiten und ausführen. Sobald Ihr Workflow fertig ist, können Sie ihn speichern oder exportieren. Sie können die visuelle Designerfahrung auch für Rapid Prototyping nutzen.

Einen Runbook-Workflow erstellen

1. Melden Sie sich bei der Systems-Manager-Automation-Konsole an.
2. Wählen Sie Runbook erstellen.
3. Geben Sie im Feld Name einen Namen für Ihr Runbook ein, z. B. *MyNewRunbook*.
4. Wählen Sie neben der Option Design und Code das Stiftsymbol aus und geben Sie einen Namen für Ihr Runbook ein.

Sie können jetzt einen Workflow für Ihr neues Runbook entwerfen.

Ein Runbook entwerfen

Um einen Runbook-Workflow mithilfe der visuellen Designerfahrung zu entwerfen, ziehen Sie eine Automatisierungs-Aktion aus dem Browser Aktionen auf die Arbeitsfläche und platzieren sie an der gewünschten Stelle im Workflow Ihres Runbooks. Sie können Aktionen in Ihrem Workflow auch neu

anordnen, indem Sie sie an eine andere Position ziehen. Wenn Sie eine Aktion auf die Arbeitsfläche ziehen, wird an der Stelle, an der Sie die Aktion in Ihrem Workflow ablegen können, eine Linie angezeigt. Nachdem eine Aktion auf der Arbeitsfläche abgelegt wurde, wird ihr Code automatisch generiert und dem Inhalt Ihres Runbooks hinzugefügt.

Wenn Sie den Namen der Aktion kennen, die Sie hinzufügen möchten, verwenden Sie das Suchfeld oben im Browser Aktionen, um die Aktion zu finden.

Nachdem Sie eine Aktion auf der Arbeitsfläche abgelegt haben, konfigurieren Sie sie mithilfe des Fensters Formular auf der rechten Seite. Dieser Bereich enthält die Registerkarten Allgemein, Eingaben, Ausgaben und Konfiguration für jede Automatisierungs-Aktion oder API-Aktion, die Sie auf der Arbeitsfläche platzieren. Die Registerkarte Allgemein enthält beispielsweise die folgenden Abschnitte:

- Der Schrittname identifiziert den Schritt. Geben Sie einen eindeutigen Wert für den Schrittnamen an.
- Mithilfe der Beschreibung können Sie beschreiben, was die Aktion im Workflow Ihres Runbooks bewirkt.

Die Registerkarte Eingaben enthält Felder, die je nach Aktion variieren. Die `aws:executeScript`-Automatisierungs-Aktion enthält beispielsweise die folgenden Abschnitte:

- Die Laufzeit ist die Sprache, die zum Ausführen des bereitgestellten Skripts verwendet wird.
- Der Handler ist der Name Ihrer Funktion. Sie müssen sicherstellen, dass die im Handler definierte Funktion über zwei Parameter verfügt: `events` und `context`. Die PowerShell-Laufzeit unterstützt diesen Parameter nicht.
- Das Skript ist ein eingebettetes Skript, das während des Workflows ausgeführt werden soll.
- (Optional) Der Anhang ist für eigenständige Skripts oder ZIP-Dateien vorgesehen, die durch die Aktion aufgerufen werden können. Dieser Parameter muss für JSON-Runbooks angegeben werden.

Auf der Registerkarte Ausgaben können Sie die Werte angeben, die Sie aus einer Aktion ausgeben möchten. Sie können in späteren Aktionen Ihres Workflows auf Ausgabewerte verweisen oder zu Protokollierungszwecken Ausgaben aus Aktionen generieren. Nicht alle Aktionen verfügen über eine Registerkarte Ausgaben, da nicht alle Aktionen Ausgaben unterstützen. Die `aws:pause`-Aktion unterstützt beispielsweise keine Ausgaben. Für Aktionen, die Ausgaben unterstützen, besteht die Registerkarte Ausgaben aus den folgenden Abschnitten:

- Der Name ist der Name, der für den Ausgabewert verwendet werden soll. Sie können in späteren Aktionen Ihres Workflows auf Ausgaben verweisen.
- Der Selector ist eine JSONPath-Ausdruckszeichenfolge, die mit `"$."` beginnt und zum Auswählen einer oder mehrerer Komponenten innerhalb eines JSON-Elements verwendet wird.
- Der Typ ist der Datentyp für den Ausgabewert. Beispielsweise ein Datentyp `String` oder `Integer`.

Die Registerkarte Konfiguration enthält Eigenschaften und Optionen, die von allen Automatisierungs-Aktionen verwendet werden können. Die Aktion besteht aus folgenden Abschnitten:

- Die Eigenschaft Max. Versuche gibt an, wie oft eine Aktion wiederholt wird, wenn sie fehlschlägt.
- Die Eigenschaft Timeout in Sekunden gibt den Timeout-Wert für eine Aktion an.
- Die Eigenschaft Ist kritisch bestimmt, ob der Aktionsfehler die gesamte Automatisierung stoppt.
- Die Eigenschaft Nächster Schritt bestimmt, welche Aktion die Automatisierung als Nächstes im Runbook ausführt.
- Die Eigenschaft Schlägt fehl bestimmt, welche Aktion die Automatisierung im Runbook als Nächstes ausführt, falls die Aktion fehlschlägt.
- Die Eigenschaft Wird abgebrochen bestimmt, welche Aktion die Automatisierung als Nächstes im Runbook ausführt, wenn die Aktion von einem Benutzer abgebrochen wird.

Um eine Aktion zu löschen, können Sie die Rücktaste, die Werkzeugleiste über der Arbeitsfläche, verwenden oder mit der rechten Maustaste klicken und Aktion löschen wählen.

Wenn Ihr Workflow wächst, passt er möglicherweise nicht in die Arbeitsfläche. Um den Workflow an die Arbeitsfläche anzupassen, führen Sie eine der folgenden Optionen aus:

- Verwenden Sie die Steuerelemente an den Seitenbereichen, um die Größe der Bedienfelder zu ändern oder sie zu schließen.
- Verwenden Sie die Werkzeugleiste oben auf der Leinwand, um das Workflow-Diagramm zu vergrößern oder zu verkleinern.

Ihr Runbook aktualisieren

Sie können einen vorhandenen Runbook-Workflow aktualisieren, indem Sie eine neue Version Ihres Runbook erstellen. Aktualisierungen Ihrer Runbooks können mithilfe der visuellen Designerfahrung

oder durch direkte Bearbeitung des Codes vorgenommen werden. Um ein vorhandenes Runbook zu aktualisieren, gehen Sie wie folgt vor:

1. Melden Sie sich bei der Systems-Manager-Automation-Konsole an.
2. Wählen Sie das Runbooks, das Sie aktualisieren möchten.
3. Wählen Sie Create new version (Neue Version erstellen)aus.
4. Die visuelle Designerfahrung besteht aus zwei Bereichen: einem Codebereich und einem visuellen Workflow-Bereich. Wählen Sie im visuellen Workflow-Bereich die Option Design aus, um Ihren Workflow mit der visuellen Designerfahrung zu bearbeiten. Wenn Sie fertig sind, wählen Sie Neue Version erstellen aus, um Ihre Änderungen zu speichern und den Vorgang zu beenden.
5. (Optional) Verwenden Sie den Codebereich, um den Runbook-Inhalt in YAML oder JSON zu bearbeiten.

Ihr Runbook exportieren

Gehen Sie wie folgt vor, um den Workflow-YAML- oder JSON-Code Ihres Runbooks sowie ein Diagramm Ihres Workflows zu exportieren:

1. Wählen Sie Ihr Runbook in der Dokumentenkonsole aus.
2. Wählen Sie Create new version (Neue Version erstellen)aus.
3. Wählen Sie in der Dropdownliste Aktionen aus, ob Sie das Diagramm oder das Runbook exportieren möchten und welches Format Sie bevorzugen.

Konfigurieren von Eingaben und Ausgaben für Ihre Aktionen

Jede Automatisierungs-Aktion reagiert auf der Grundlage von Eingaben, die sie empfängt. In den meisten Fällen geben Sie die Ausgabe dann an die nachfolgenden Aktionen weiter. In der visuellen Designerfahrung können Sie die Eingabe- und Ausgabedaten einer Aktion auf den Registerkarten Eingaben und Ausgaben des Formularfensters konfigurieren.

Weitere Informationen zum Definieren und Verwenden von Ausgaben für Automatisierungs-Aktionen finden Sie unter Verwenden von Aktionsausgaben als Eingaben.

Eingabedaten für eine Aktion angeben

Jede Automatisierungs-Aktion hat eine oder mehrere Eingaben, für die Sie einen Wert angeben müssen. Der Wert, den Sie für die Eingabe einer Aktion angeben, wird durch den Datentyp und

das Format bestimmt, die von der Aktion akzeptiert werden. Für die `aws:sleep`-Aktionen ist beispielsweise ein Zeichenfolgenwert im ISO-8601-Format für die `Duration`-Eingabe erforderlich.

Im Allgemeinen verwenden Sie im Workflow Ihres Runbooks Aktionen, die Ausgaben zurückgeben, die Sie in nachfolgenden Aktionen verwenden möchten. Es ist wichtig, dass Sie sicherstellen, dass Ihre Eingabewerte korrekt sind, um Fehler im Workflow Ihres Runbooks zu vermeiden. Eingabewerte sind auch deshalb wichtig, weil sie bestimmen, ob die Aktion die erwartete Ausgabe zurückgibt. Wenn Sie die `aws:executeAwsApi`-Aktion verwenden, sollten Sie beispielsweise sicherstellen, dass Sie den richtigen Wert für den API-Vorgang angeben.

Die Ausgabedaten für eine Aktion definieren

Einige Automatisierungs-Aktionen geben eine Ausgabe zurück, nachdem sie ihre definierten Operationen ausgeführt haben. Aktionen, die Ausgaben zurückgeben, haben entweder vordefinierte Ausgaben oder ermöglichen es Ihnen, die Ausgaben selbst zu definieren. Die `aws:createImage`-Aktion hat beispielsweise vordefinierte Ausgaben, die `ImageId` und `ImageState` zurückgeben. Im Vergleich dazu können Sie mit der `aws:executeAwsApi`-Aktion die Ausgaben definieren, die Sie von der angegebenen API-Operation erwarten. Daher können Sie einen oder mehrere Werte aus einer einzelnen API-Operation zurückgeben, um sie in nachfolgenden Aktionen zu verwenden.

Um Ihre eigenen Ausgaben für eine Automatisierungs-Aktion zu definieren, müssen Sie einen Namen der Ausgabe, den Datentyp und den Ausgabewert angeben. Um die `aws:executeAwsApi`-Aktion weiterhin als Beispiel zu verwenden, nehmen wir an, Sie rufen den `DescribeInstances`-API-Vorgang von Amazon EC2 aus auf. In diesem Beispiel möchten Sie den `State` einer Amazon-EC2-Instance zurückgeben oder ausgeben und den Workflow Ihres Runbooks auf der Grundlage der Ausgabe verzweigen. Sie geben der Ausgabe **`InstanceState`** einen Namen und verwenden den **`String`**-Datentyp.

Das Verfahren zur Definition des tatsächlichen Werts der Ausgabe unterscheidet sich je nach Aktion. Wenn Sie beispielsweise die `aws:executeScript`-Aktion verwenden, müssen Sie `return`-Anweisungen in Ihren Funktionen verwenden, um Daten für Ihre Ausgaben bereitzustellen. Bei anderen Aktionen wie `aws:executeAwsApi`, `aws:waitForAwsResourceProperty` und `aws:assertAwsResourceProperty` ist `Selector` erforderlich. `Selector` oder `PropertySelector`, wie sich einige Aktionen darauf beziehen, ist eine JSONPath-Zeichenfolge, die verwendet wird, um die JSON-Antwort aus einer API-Operation zu verarbeiten. Es ist wichtig zu verstehen, wie das JSON-Antwortobjekt aus einer API-Operation strukturiert ist, damit Sie den richtigen Wert für Ihre Ausgabe auswählen können. Sehen Sie sich das folgende Beispiel für eine JSON-Antwort an, indem Sie die zuvor erwähnte `DescribeInstances`-API-Operation verwenden:

```
{
  "reservationSet": {
    "item": {
      "reservationId": "r-1234567890abcdef0",
      "ownerId": 123456789012,
      "groupSet": "",
      "instancesSet": {
        "item": {
          "instanceId": "i-1234567890abcdef0",
          "imageId": "ami-bff32ccc",
          "instanceState": {
            "code": 16,
            "name": "running"
          },
          "privateDnsName": "ip-192-168-1-88.eu-west-1.compute.internal",
          "dnsName": "ec2-54-194-252-215.eu-west-1.compute.amazonaws.com",
          "reason": "",
          "keyName": "my_keypair",
          "amiLaunchIndex": 0,
          "productCodes": "",
          "instanceType": "t2.micro",
          "launchTime": "2018-05-08T16:46:19.000Z",
          "placement": {
            "availabilityZone": "eu-west-1c",
            "groupName": "",
            "tenancy": "default"
          },
          "monitoring": {
            "state": "disabled"
          },
          "subnetId": "subnet-56f5f000",
          "vpcId": "vpc-11112222",
          "privateIpAddress": "192.168.1.88",
          "ipAddress": "54.194.252.215",
          "sourceDestCheck": true,
          "groupSet": {
            "item": {
              "groupId": "sg-e4076000",
              "groupName": "SecurityGroup1"
            }
          },
          "architecture": "x86_64",
          "rootDeviceType": "ebs",
```

```
            "rootDeviceName": "/dev/xvda",
            "blockDeviceMapping": {
              "item": {
                "deviceName": "/dev/xvda",
                "ebs": {
                  "volumeId": "vol-1234567890abcdef0",
                  "status": "attached",
                  "attachTime": "2015-12-22T10:44:09.000Z",
                  "deleteOnTermination": true
                }
              }
            },
            "virtualizationType": "hvm",
            "clientToken": "xMcwG14507example",
            "tagSet": {
              "item": {
                "key": "Name",
                "value": "Server_1"
              }
            },
            "hypervisor": "xen",
            "networkInterfaceSet": {
              "item": {
                "networkInterfaceId": "eni-551ba000",
                "subnetId": "subnet-56f5f000",
                "vpcId": "vpc-11112222",
                "description": "Primary network interface",
                "ownerId": 123456789012,
                "status": "in-use",
                "macAddress": "02:dd:2c:5e:01:69",
                "privateIpAddress": "192.168.1.88",
                "privateDnsName": "ip-192-168-1-88.eu-west-1.compute.internal",
                "sourceDestCheck": true,
                "groupSet": {
                  "item": {
                    "groupId": "sg-e4076000",
                    "groupName": "SecurityGroup1"
                  }
                },
                "attachment": {
                  "attachmentId": "eni-attach-39697adc",
                  "deviceIndex": 0,
                  "status": "attached",
                  "attachTime": "2018-05-08T16:46:19.000Z",
```

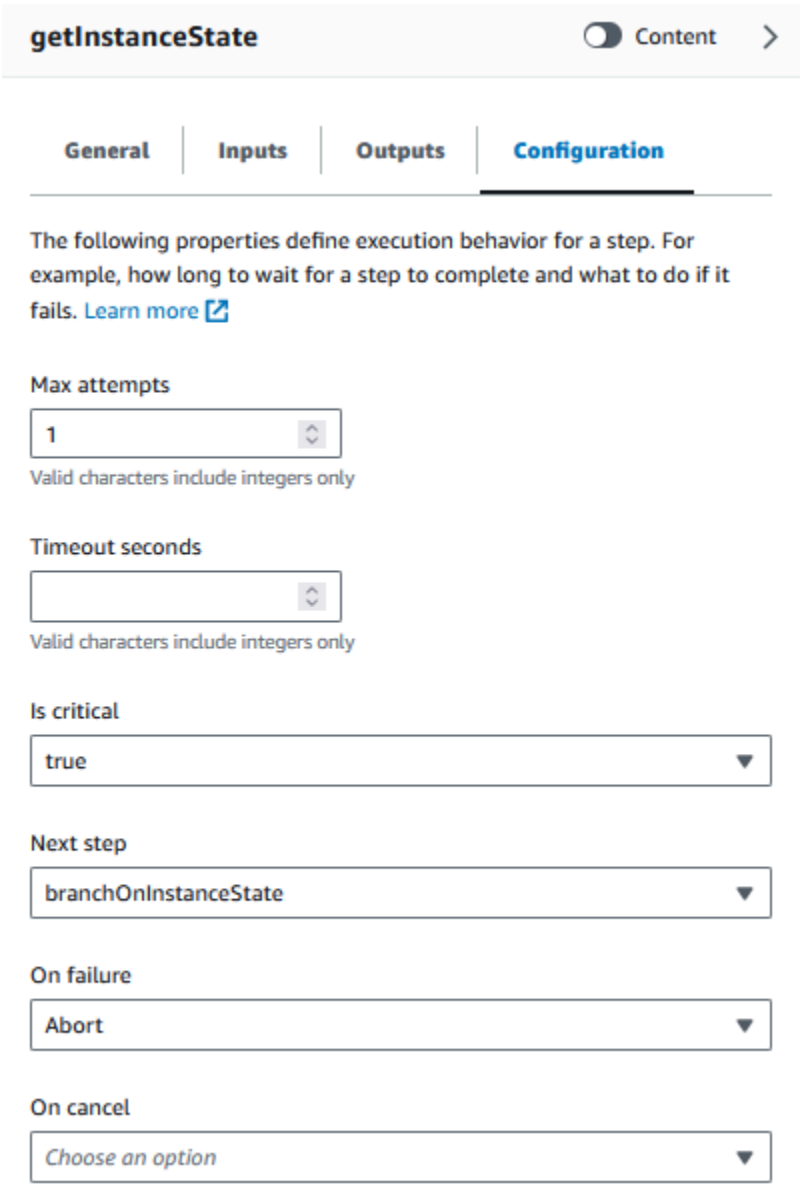
```
                "deleteOnTermination": true
              },
              "association": {
                "publicIp": "54.194.252.215",
                "publicDnsName": "ec2-54-194-252-215.eu-west-1.compute.amazonaws.com",
                "ipOwnerId": "amazon"
              },
              "privateIpAddressesSet": {
                "item": {
                  "privateIpAddress": "192.168.1.88",
                  "privateDnsName": "ip-192-168-1-88.eu-west-1.compute.internal",
                  "primary": true,
                  "association": {
                    "publicIp": "54.194.252.215",
                    "publicDnsName": "ec2-54-194-252-215.eu-
west-1.compute.amazonaws.com",
                    "ipOwnerId": "amazon"
                  }
                }
              },
              "ipv6AddressesSet": {
                "item": {
                  "ipv6Address": "2001:db8:1234:1a2b::123"
                }
              }
            }
          },
          "iamInstanceProfile": {
            "arn": "arn:aws:iam::123456789012:instance-profile/AdminRole",
            "id": "ABCAJEDNCAA64SSD123AB"
          },
          "ebsOptimized": false,
          "cpuOptions": {
            "coreCount": 1,
            "threadsPerCore": 1
          }
        }
      }
    }
  }
}
```

Im JSON-Antwortobjekt ist die Instance `State` in einem `Instances`-Objekt verschachtelt, das im `Reservations`-Objekt verschachtelt ist. Um den Wert der Instance `State` zurückzugeben, verwenden Sie die folgende Zeichenfolge für `Selector`, damit der Wert in unserer Ausgabe verwendet werden kann: **`$.Reservations[0].Instances[0].State.Name`**.

Um in nachfolgenden Aktionen des Workflows Ihres Runbooks auf einen Ausgabewert zu verweisen, wird das folgende Format verwendet: `{{ StepName.NameOfOutput }}`. Beispiel, **`{{ GetInstanceState.InstanceState }}`**. In der visuellen Designerfahrung können Sie mithilfe der Dropdownliste für die Eingabe Ausgabewerte auswählen, die in nachfolgenden Aktionen verwendet werden sollen. Wenn Sie Ausgaben in nachfolgenden Aktionen verwenden, muss der Datentyp der Ausgabe mit dem Datentyp für die Eingabe übereinstimmen. In diesem Beispiel ist die `InstanceState`-Ausgabe `String`. Um den Wert in der Eingabe einer nachfolgenden Aktion zu verwenden, muss die Eingabe daher `String` akzeptieren.

Fehlerbehandlung bei der visuellen Designerfahrung

Wenn eine Aktion einen Fehler meldet, stoppt Automation standardmäßig den Workflow des Runbooks vollständig. Das liegt daran, dass der Standardwert für die `onFailure`-Eigenschaft für alle Aktionen `Abort` ist. Sie können konfigurieren, wie Automation mit Fehlern im Workflow Ihres Runbooks umgeht. Auch wenn Sie die Fehlerbehandlung konfiguriert haben, können einige Fehler dennoch dazu führen, dass eine Automatisierung fehlschlägt. Weitere Informationen finden Sie unter Fehlerbehebung für Systems Manager Automation.. In der visuellen Designerfahrung konfigurieren Sie die Fehlerbehandlung im Bereich Konfiguration.



Bei einem Fehler die Aktion erneut versuchen

Um eine Aktion im Falle eines Fehlers erneut zu versuchen, geben Sie einen Wert für die Eigenschaft Max. Versuche an. Der Standardwert lautet 1. Wenn Sie einen Wert größer als 1 angeben, gilt die Aktion erst dann als fehlgeschlagen, wenn alle Wiederholungsversuche fehlgeschlagen sind.

Timeouts

Sie können ein Timeout für Aktionen konfigurieren, um festzulegen, wie viele Sekunden Ihre Aktion maximal ausgeführt werden kann, bevor sie fehlschlägt. Um ein Timeout zu konfigurieren, geben Sie in der Eigenschaft Timeout-Sekunden die Anzahl der Sekunden ein, die Ihre Aktion warten soll, bis die Aktion fehlschlägt. Wenn das Timeout erreicht ist und die Aktion einen Wert von `Max attempts`

hat, der größer als 1 ist, gilt der Schritt erst dann als Timeout, wenn die Wiederholungsversuche abgeschlossen sind.

Fehlgeschlagene Aktionen

Wenn eine Aktion fehlschlägt, stoppt Automation standardmäßig den Workflow des Runbooks vollständig. Sie können dieses Verhalten ändern, indem Sie einen alternativen Wert für die Eigenschaft Bei einem Ausfall der Aktionen in Ihrem Runbook angeben. Wenn Sie möchten, dass der Workflow mit dem nächsten Schritt im Runbook fortfährt, wählen Sie Weiter aus. Wenn der Workflow zu einem anderen nachfolgenden Schritt im Runbook springen soll, wählen Sie Schritt aus und geben Sie dann den Namen des Schritts ein.

Abgebrochene Aktionen

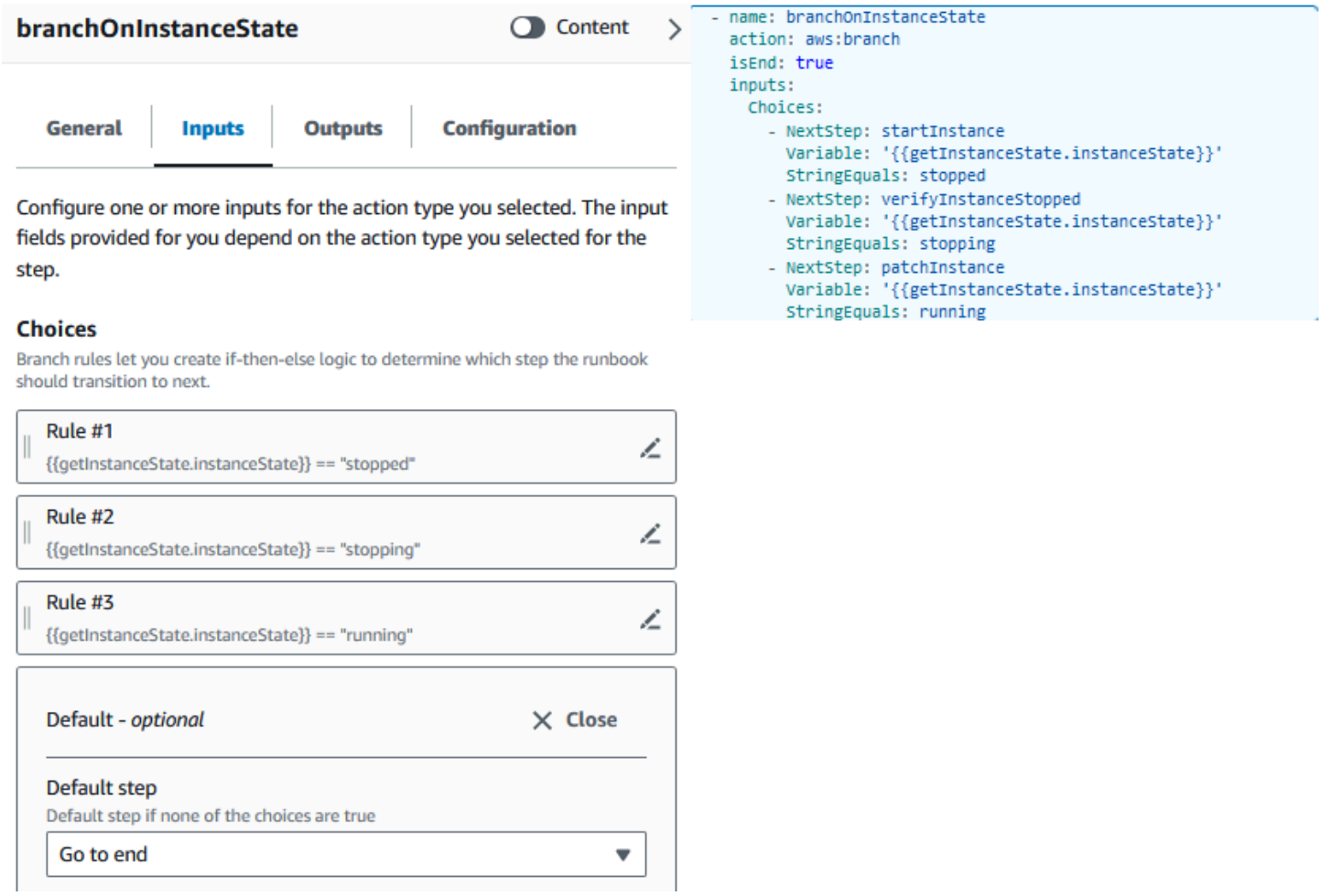
Wenn eine Aktion von einem Benutzer abgebrochen wird, stoppt Automation standardmäßig den Workflow des Runbooks vollständig. Sie können dieses Verhalten ändern, indem Sie einen alternativen Wert für die Eigenschaft Bei Abbruch der Aktionen in Ihrem Runbook angeben. Wenn der Workflow zu einem anderen nachfolgenden Schritt im Runbook springen soll, wählen Sie Schritt aus und geben Sie dann den Namen des Schritts ein.

Kritische Aktionen

Sie können eine Aktion als kritisch kennzeichnen, was bedeutet, dass sie den allgemeinen Berichtsstatus Ihrer Automatisierung bestimmt. Wenn ein Schritt mit dieser Bezeichnung fehlschlägt, meldet Automation den Endstatus als `Failed` unabhängig vom Erfolg anderer Aktionen. Um eine Aktion als kritisch zu konfigurieren, belassen Sie den Standardwert Richtig für die Eigenschaft Ist kritisch.

Aktionen beenden

Die Eigenschaft Ist am Ende stoppt eine Automatisierung am Ende der angegebenen Aktion. Der Standardwert dieser Eigenschaft ist `false`. Wenn Sie diese Eigenschaft für eine Aktion konfigurieren, stoppt die Automatisierung unabhängig davon, ob die Aktion erfolgreich ist oder fehlschlägt. Diese Eigenschaft wird am häufigsten bei `aws:branch`-Aktionen verwendet, um unerwartete oder undefinierte Eingabewerte zu verarbeiten. Das folgende Beispiel zeigt ein Runbook, das einen Instance-Status von entweder `running`, `stopping` oder `stopped` erwartet. Wenn sich eine Instance in einem anderen Status befindet, wird die Automatisierung beendet.



Tutorial: Ein Runbook mithilfe der visuellen Designerfahrung erstellen

In diesem Tutorial lernen Sie die Grundlagen für die Arbeit mit der visuellen Designerfahrung von Systems Manager Automation. In der visuellen Designerfahrung können Sie ein Runbook erstellen, das mehrere Aktionen verwendet. Sie verwenden das Drag-and-Drop-Feature, um Aktionen auf der Arbeitsfläche anzuordnen. Sie suchen auch nach diesen Aktionen, wählen sie aus und konfigurieren sie. Anschließend können Sie den automatisch generierten YAML-Code für den Workflow Ihres Runbooks anzeigen, die visuelle Designerfahrung beenden, das Runbook ausführen und die Ausführungsdetails überprüfen.

In diesem Tutorial erfahren Sie auch, wie Sie das Runbook aktualisieren und die neue Version anzeigen. Am Ende des Tutorials führen Sie einen Bereinigungsschritt durch und löschen Ihr Runbook.

Nachdem Sie dieses Tutorial abgeschlossen haben, wissen Sie, wie Sie mithilfe der visuellen Designerfahrung ein Runbook erstellen können. Sie werden auch wissen, wie Sie Ihr Runbook aktualisieren, ausführen und löschen.

> **Note**
>
> Bevor Sie mit diesem Tutorial beginnen, stellen Sie sicher, dass Sie Einrichten der Automatisierung abschließen.

Themen

- Schritt 1: Zur visuellen Designerfahrung navigieren
- Schritt 2: Einen Workflow erstellen
- Schritt 3: Den automatisch generierten Code überprüfen
- Schritt 4: Ihr neues Runbook ausführen
- Schritt 5: Bereinigen

Schritt 1: Zur visuellen Designerfahrung navigieren

1. Melden Sie sich bei der Systems-Manager-Automation-Konsole an.
2. Wählen Sie Automation-Runbook erstellen.

Schritt 2: Einen Workflow erstellen

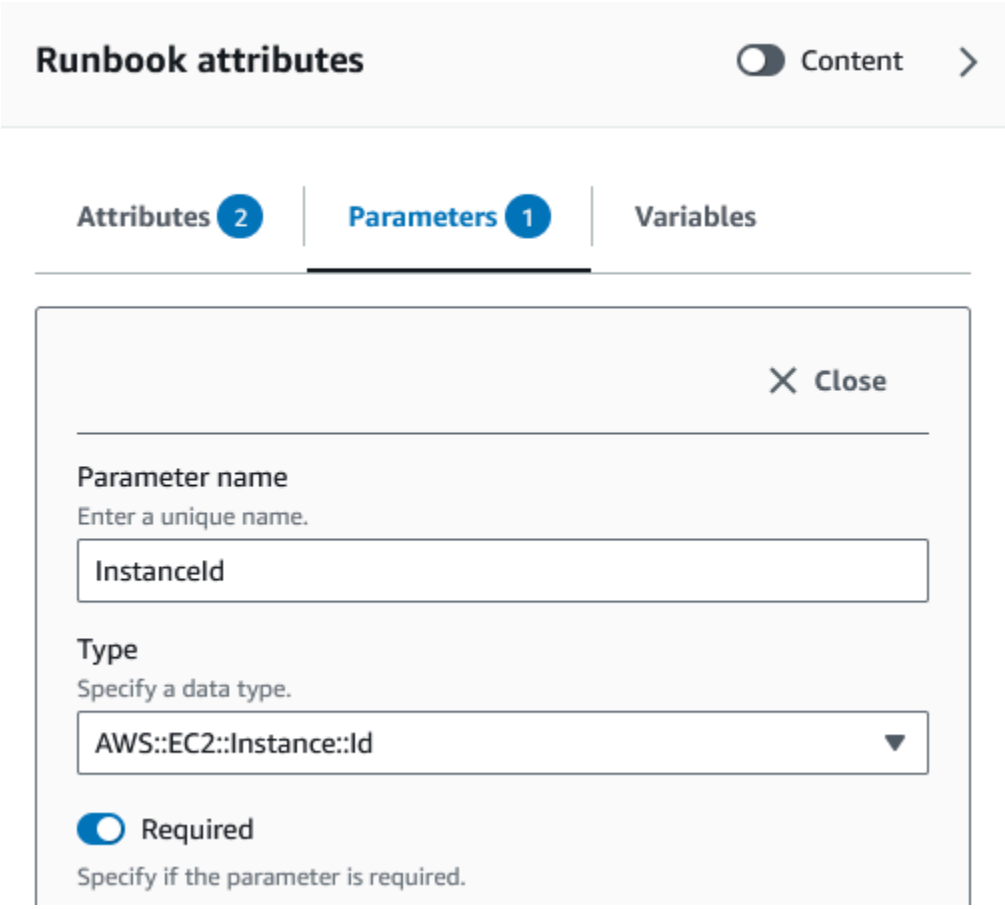
In der visuellen Designerfahrung ist ein Workflow eine grafische Darstellung Ihres Runbooks auf der Arbeitsfläche. Sie können die visuelle Designerfahrung verwenden, um die einzelnen Aktionen Ihres Runbooks zu definieren, zu konfigurieren und zu untersuchen.

So erstellen Sie ein Workflow

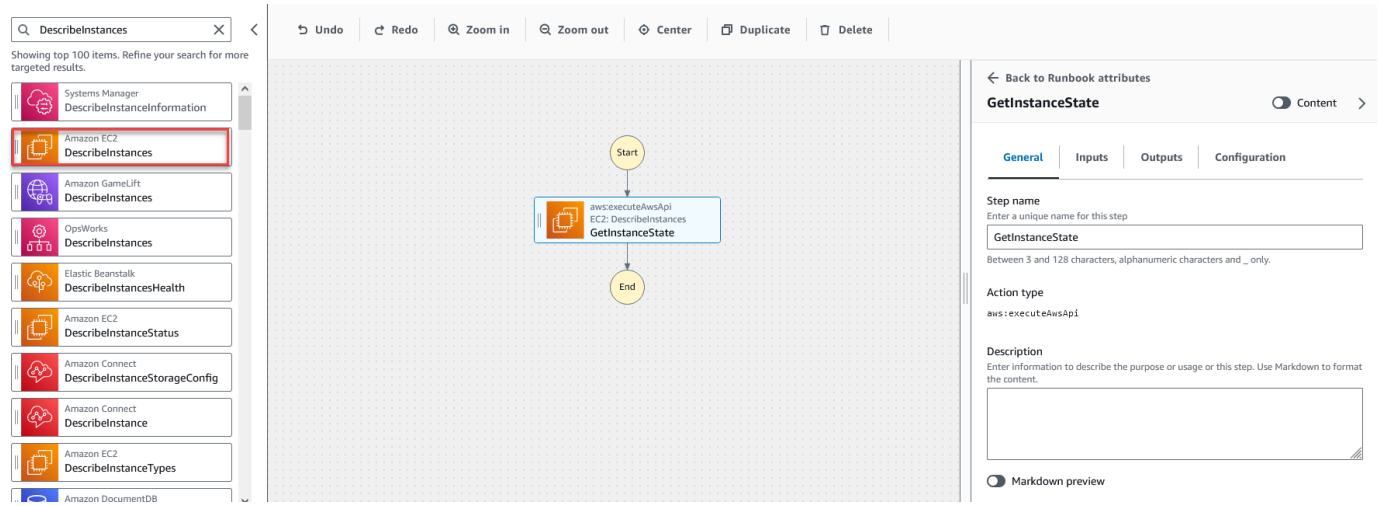
1. Wählen Sie neben der Option Design und Code das Stiftsymbol aus und geben Sie einen Namen für Ihr Runbook ein. Geben Sie für dieses Tutorial **`VisualDesignExperienceTutorial`** ein.

2. Erweitern Sie im Bereich Dokumentattribute des Bedienfelds Formular die Dropdownliste Eingabeparameter und wählen Sie Parameter hinzufügen aus.
   a. Geben Sie im Feld Parametername **`InstanceId`** ein.
   b. Wählen Sie in der Dropdownliste Typ die Option AWS::EC2::Instance.

c. Wählen Sie den Schalter Erforderlich aus.



3. Geben Sie im AWS APIsBrowser **`DescribeInstances`** in die Suchleiste ein.
4. Ziehe eine Amazon EC2 — DescribeInstances Aktion auf die leere Leinwand.
5. Geben Sie für Schrittname einen Wert ein. Verwenden Sie in diesem Tutorial **`GetInstanceState`** als Namen.



a. Erweitern Sie das Dropdown-Menü Zusätzliche Eingaben und geben Sie im Feld Eingabename **`InstanceIds`** ein.

b. Wählen Sie die Registerkarte Eingaben.

c. Wählen Sie im Feld Eingabewert die **`InstanceId`** Dokumenteingabe aus. Dies verweist auf den Wert des Eingabeparameters, den Sie zu Beginn des Verfahrens erstellt haben. Da die InstanceIdsEingabe für die `DescribeInstances` Aktion `StringList` Werte akzeptiert, müssen Sie die InstanceIdEingabe in eckige Klammern setzen. Das YAML für den Eingabewert sollte dem Folgenden entsprechen: **`['{{ InstanceId }}']`**.

d. Wählen Sie auf der Registerkarte Ausgaben die Option Ausgabe hinzufügen aus und geben Sie **`InstanceState`** in das Feld Name ein.

e. Geben Sie **`$.Reservations[0].Instances[0].State.Name`** im Feld Auswahl ein.

f. Wählen Sie in der Dropdownliste Typ die Option Zeichenfolge aus.

6. Ziehen Sie eine Branch-Aktion aus dem Aktionsbrowser und legen Sie sie unter dem **`GetInstanceState`**-Schritt ab.

7. Geben Sie für Schrittname einen Wert ein. Verwenden Sie in diesem Tutorial den Namen **`BranchOnInstanceState`**.

   Um die Branch-Logik zu definieren, führen Sie die folgenden Schritte aus:

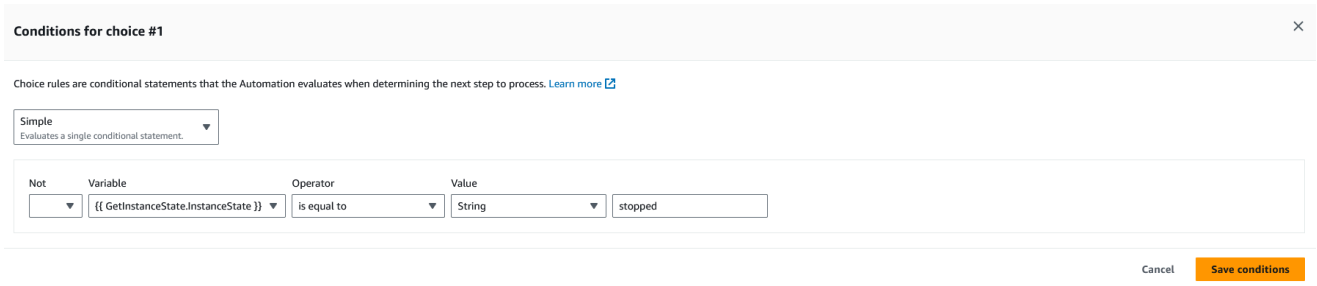
   a. Wählen Sie den **`Branch`**-Status auf der Arbeitsfläche aus. Wählen Sie dann unter Eingaben und Wahlmöglichkeiten das Stiftsymbol aus, um Regel #1 zu bearbeiten.

   b. Wählen Sie Bedingungen hinzufügen.

c. Wählen Sie im Dialogfeld Bedingungen für Regel #1 die **`GetInstanceState.InstanceState`**-Schrittausgabe aus der Dropdownliste Variable aus.

d. Wählen Sie für Operator die Option Ist gleich aus.

e. Wählen Sie als Wert Zeichenfolge aus der Dropdown-Liste aus. Geben Sie **`stopped`** ein.



f. Wählen Sie Bedingungen speichern aus.

g. Wählen Sie Neue Auswahlregel hinzufügen aus.

h. Wählen Sie Bedingungen hinzufügen für Regel #2.

i. Wählen Sie im Dialogfeld Bedingungen für Regel #2 die **`GetInstanceState.InstanceState`**-Schrittausgabe aus der Dropdownliste Variable aus.

j. Wählen Sie für Operator die Option Ist gleich aus.

k. Wählen Sie als Wert Zeichenfolge aus der Dropdown-Liste aus. Geben Sie **`stopping`** ein.

l. Wählen Sie Bedingungen speichern aus.

m. Wählen Sie Neue Auswahlregel hinzufügen aus.

n. Wählen Sie für Regel #3 Bedingungen hinzufügen.

o. Wählen Sie im Dialogfeld Bedingungen für Regel #3 die **`GetInstanceState.InstanceState`**-Schrittausgabe aus der Dropdownliste Variable aus.

p. Wählen Sie für Operator die Option Ist gleich aus.

q. Wählen Sie als Wert Zeichenfolge aus der Dropdown-Liste aus. Geben Sie **`running`** ein.

r. Wählen Sie Bedingungen speichern aus.

s. Wählen Sie in der Standardregel für den Standardschritt die Option Gehe zum Ende aus.

8. Ziehen Sie eine Aktion „Instanzstatus ändern“ in das leere Feld Aktion hierher ziehen unter dem Feld {{ GetInstanceState. InstanceState }} == Zustand „gestoppt“.

a. Geben Sie als Schrittnamen **`StartInstance`** ein.

b. Wählen Sie auf der Registerkarte Eingaben unter Instanz IDs den Eingabewert für das InstanceIdDokument aus der Dropdownliste aus.

c. Geben Sie für den gewünschten Status **`running`** an.

9. Ziehen Sie eine Aktion „Auf AWS Ressource warten“ in das leere Feld Aktion hierher ziehen unter {{ GetInstanceState. InstanceState }} == Zustand „stoppt“.

10. Geben Sie für Schrittname einen Wert ein. Verwenden Sie in diesem Tutorial den Namen **`WaitForInstanceStop`**.

    a. Wählen Sie für das Feld Service Amazon aus EC2.

    b. Wählen Sie für das API-Feld DescribeInstances.

    c. Geben Sie für das Feld Eigenschaftsauswahl den Wert **`$.Reservations[0].Instances[0].State.Name`** ein.

    d. Geben **`["stopped"]`** Sie für den Parameter Gewünschte Werte ein.

    e. Wählen Sie auf der Registerkarte Konfiguration der WaitForInstanceStopAktion StartInstanceaus der Dropdownliste Nächster Schritt aus.

11. Ziehen Sie die Aktion „Befehl auf Instanzen ausführen“ in das leere Feld Aktion hierher ziehen unter {{ GetInstanceState. InstanceState }} == Zustand „läuft“.

12. Geben Sie als Schrittnamen **`SayHello`** ein.

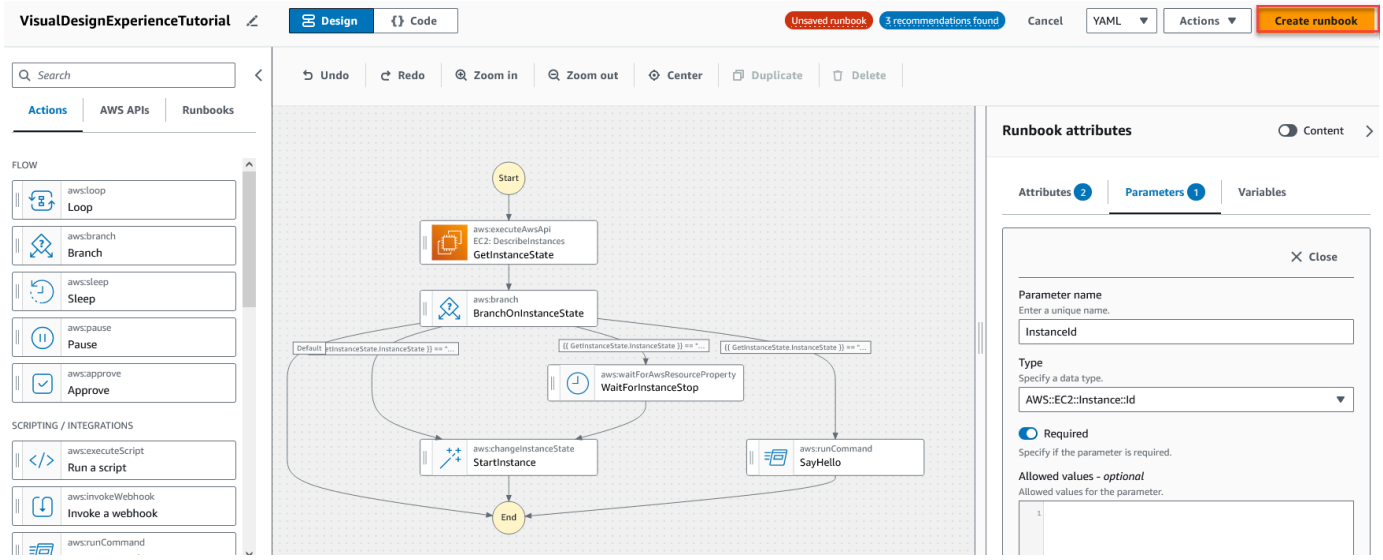
    a. Geben Sie auf der Registerkarte Eingaben den Wert **`AWS-RunShellScript`** für den Parameter Dokumentname ein.

    b. Wählen Sie für InstanceIdsden InstanceIdDokumenteingabewert aus der Dropdownliste aus.

    c. Erweitern Sie das Dropdownmenü Zusätzliche Eingaben und wählen Sie im Dropdownmenü Eingabename die Option Parameter aus.

    d. Geben Sie im Feld Eingabewert **`{"commands": "echo 'Hello World'"}`** ein.

13. Prüfen Sie das fertige Runbook auf der Arbeitsfläche und wählen Sie Runbook erstellen aus, um das Tutorial-Runbook zu speichern.



Schritt 3: Den automatisch generierten Code überprüfen

Wenn Sie Aktionen aus dem Browser Aktion auf die Arbeitsfläche ziehen und dort ablegen, erstellt die visuelle Designerfahrung automatisch den YAML- oder JSON-Inhalt Ihres Runbooks in Echtzeit. Sie können diesen Code anzeigen und bearbeiten. Um den automatisch generierten Code anzuzeigen, wählen Sie Code für die Umschalter Design und Code aus.

Schritt 4: Ihr neues Runbook ausführen

Nachdem Sie Ihr Runbook erstellt haben, können Sie die Automatisierung ausführen.

So führen Sie Ihr neues Automation-Runbook aus

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Automatisierung und Automatisierung ausführen aus.
3. Wählen Sie in der Liste Automation-Dokument ein Runbook. Wählen Sie eine oder mehrere Optionen im Bereich Dokumentkategorien, um SSM-Dokumente nach ihrem Zweck zu filtern. Um ein Runbook anzuzeigen, das Sie besitzen, wählen Sie die Im Besitz von mir-Registerkarte. Um ein Runbook anzuzeigen, das für Ihr Konto freigegeben ist, wählen Sie die Registerkarte Mit mir geteilt. Um alle Runbooks anzuzeigen, wählen Sie die Registerkarte Alle Dokumente.

> **Note**
>
> Sie können Informationen zu einem Runbook einsehen, indem Sie den Runbook-Namen auswählen.

4. Überprüfen Sie im Abschnitt Dokument-Details, ob Dokumentversion auf die Version gesetzt ist, die Sie ausführen möchten. Das System bietet die folgenden Versionsoptionen:
   - Standardversion zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird und eine neue Standardversion zugewiesen ist.
   - Letzte Version zur Laufzeit – Wählen Sie diese Option aus, wenn das Automation-Runbook regelmäßig aktualisiert wird, und Sie die Version auszuführen möchten, die zuletzt aktualisiert wurde.
   - 1 (Standard) – Wählen Sie diese Option zur Ausführung der ersten Version des Dokuments, welches der Standard ist.
5. Wählen Sie Weiter aus.
6. Klicken Sie auf der Seite Automation-Runbook ausführen auf Einfache Ausführung.
7. Geben Sie im Abschnitt Eingabeparameter die erforderlichen Eingaben an. Optional können Sie eine IAM-Dienstrolle aus der AutomationAssumeRoleListe auswählen.
8. (Optional) Wählen Sie einen CloudWatch Amazon-Alarm aus, der auf Ihre Automatisierung zur Überwachung angewendet werden soll. Um Ihrer Automatisierung einen CloudWatch Alarm zuzuweisen, muss der IAM-Principal, der die Automatisierung startet, über die Genehmigung für die `iam:createServiceLinkedRole` Aktion verfügen. Weitere Informationen zu CloudWatch Alarmen finden Sie unter CloudWatchAmazon-Alarme verwenden. Wenn die Automatisierung gestoppt wird, wird Ihr Alarm aktiviert. Wenn Sie AWS CloudTrail verwenden, sehen Sie den API-Aufruf in Ihrem Trail.
9. Wählen Sie Ausführen.

Schritt 5: Bereinigen

So löschen Sie Ihr Runbook

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.

3. Wählen Sie die Registerkarte In meinem Besitz aus.
4. Suchen Sie das VisualDesignExperienceTutorialRunbook.
5. Wählen Sie die Schaltfläche auf der Dokumentkartenseite aus und wählen Sie dann in der Dropdownliste Aktionen die Option Dokument löschen aus.

Erstellen von Automation-Runbooks

Jedes Runbook in Automation, einem Tool in AWS Systems Manager, definiert eine Automatisierung. Automatisierungs-Runbooks definieren die Aktionen, die während einer Automatisierung ausgeführt werden. Im Runbookinhalt definieren Sie die Eingangsparameter, Ausgaben und Aktionen, die Systems Manager auf Ihren verwalteten Instances und AWS-Ressourcen durchführt.

Automatisierung umfasst mehrere vordefinierte Runbooks, die Sie verwenden können, um allgemeine Aufgaben wie das Neustarten einer oder mehrerer Amazon Elastic Compute Cloud (Amazon EC2)-Instances oder das Erstellen eines Amazon Machine Image (AMI) auszuführen. Ihre Anwendungsfälle können jedoch über die Funktionen der vordefinierten Runbooks hinausgehen. In diesem Fall können Sie eigene Runbooks erstellen und an Ihre Bedürfnisse anpassen.

Ein Runbook besteht aus Automatisierungsaktionen, Parametern für diese Aktionen und Eingabeparametern, die Sie angeben. Der Inhalt eines Runbooks wird entweder in YAML oder JSON geschrieben. Wenn Sie mit YAML oder JSON nicht vertraut sind, empfehlen wir die Verwendung des visuellen Designers oder das Erlernen mehr über eine der Auszeichnungssprachen, bevor Sie versuchen, Ihr eigenes Runbook zu erstellen. Weitere Informationen zum visuellen Designer finden Sie unter Visuelle Designerfahrung für Automation-Runbooks.

Die folgenden Abschnitten helfen Ihnen, Ihr erstes Runbook erstellen.

Identifizieren Sie Ihren Anwendungsfall

Der erste Schritt beim Erstellen eines Runbooks besteht darin, Ihren Anwendungsfall zu identifizieren. Beispielsweise haben Sie geplant, das `AWS-CreateImage`-Runbook täglich auf allen Ihren Amazon EC2 Produktions-Instances auszuführen.. Am Ende des Monats entscheiden Sie, dass Sie über mehr Images verfügen, als für Wiederherstellungspunkte erforderlich sind. Künftig möchten Sie automatisch die älteste AMI einer Amazon-EC2-Instance löschen, wenn eine neue AMI erstellt wird. Um dies zu erreichen, erstellen Sie ein neues Runbook, das folgende Funktionen erfüllt:

1. Führt die `aws:createImage`-Aktion aus und gibt die Instance-ID in der Image-Beschreibung an.
2. Führt die `aws:waitForAwsResourceProperty`-Aktion aus, um den Zustand des Images abzufragen, bis es `available` ist.

3. Nachdem der Image-Status `available` ist, führt die`aws:executeScript`-Aktion ein benutzerdefiniertes Python-Skript aus, das die IDs aller Images sammelt, die mit Ihrer Amazon-EC2-Instance verknüpft sind. Das Skript führt diese Filterung aus, indem es die Instance-ID in der Image-Beschreibung verwendet, die Sie bei der Erstellung angegeben haben. Anschließend sortiert das Skript die Liste der Bild-IDs basierend auf dem `creationDate` des Images und gibt die ID der ältesten AMI aus.
4. Zu guter Letzt wird die `aws:deleteImage`-Aktion ausgeführt, um die älteste AMI zu löschen, mithilfe der ID aus der Ausgabe des vorherigen Schritts.

In diesem Szenario haben Sie bereits das `AWS-CreateImage`-Runbook verwendet, haben aber festgestellt, dass Ihr Anwendungsfall eine größere Flexibilität erforderte. Das kommt häufig vor, da es Überschneidungen zwischen Runbooks und Automatisierungsaktionen geben kann. Daher müssen Sie möglicherweise anpassen, welche Runbooks oder Aktionen Sie verwenden, um Ihren Anwendungsfall zu adressieren.

Zum Beispiel ermöglichen die `aws:executeScript`- und die `aws:invokeLambdaFunction`-Aktion es Ihnen, benutzerdefinierte Skripts als Teil Ihrer Automatisierung auszuführen. Sie bevorzugen vielleicht `aws:invokeLambdaFunction` aufgrund der zusätzlichen unterstützten Laufzeitsprachen. Möglicherweise bevorzugen Sie jedoch `aws:executeScript`, da Sie damit Ihre Skriptinhalte direkt in YAML Runbooks erstellen und Skriptinhalte als Anhänge für JSON-Runbooks bereitstellen können. Sie könnten auch `aws:executeScript` als einfacher in Bezug auf AWS Identity and Access Management(IAM)-Einrichtung empfinden. Da es die Berechtigungen verwendet, die in der `AutomationAssumeRole` bereitgestellt sind, benötigt `aws:executeScript` keine zusätzliche AWS Lambda-Funktionsausführungsrolle.

In einem bestimmten Szenario kann eine Aktion mehr Flexibilität oder zusätzliche Funktionalität gegenüber einer anderen bieten. Daher empfiehlt es sich, die verfügbaren Eingabeparameter für das Runbook oder die Aktion zu überprüfen, die Sie verwenden möchten, um zu bestimmen, welche am besten zu Ihrem Anwendungsfall und Ihren Voreinstellungen passt.

Einrichten Ihrer Entwicklungsumgebung

Nachdem Sie Ihren Anwendungsfall und die vordefinierten Runbooks oder Automatisierungsaktionen identifiziert haben, die Sie in Ihrem Runbook verwenden möchten, müssen Sie Ihre Entwicklungsumgebung für den Inhalt Ihres Runbooks einrichten. Um Ihre Runbookinhalte zu entwickeln, empfehlen wir das AWS Toolkit for Visual Studio Code anstelle der Systems Manager Dokumentationskonsole zu verwenden.

Das Toolkit for VS Code ist eine Open-Source-Erweiterung für Visual Studio Code (VS Code), die mehr Funktionen bietet als die Systems Manager Dokumentenkonsole. Zu den hilfreichen Funktionen gehören die Schemavalidierung für YAML und JSON, Snippets für Automatisierungsaktionstypen und die automatische Vervollständigung verschiedener Optionen in YAML und JSON.

Weitere Informationen zum Installieren des Toolkit for VS Code finden Sie unterInstallieren von AWS Toolkit for Visual Studio Code. Weitere Informationen zur Verwendung des Toolkit for VS Code zum Erstellen von Runbooks finden Sie unter Arbeiten mit Systems Manager Automation-Dokumenten im AWS Toolkit for Visual Studio Code-Benutzerhandbuch.

Entwickeln von Runbook-Inhalten

Nachdem Ihr Anwendungsfall identifiziert und die Umgebung eingerichtet ist, können Sie den Inhalt für Ihr Runbook entwickeln. Ihr Anwendungsfall und Ihre Einstellungen bestimmen weitgehend die Automatisierungsaktionen oder Runbooks, die Sie in Ihren Runbook-Inhalten verwenden. Einige Aktionen unterstützen nur eine Teilmenge von Eingabeparametern im Vergleich zu einer anderen Aktion, mit der Sie eine ähnliche Aufgabe ausführen können. Andere Aktionen haben spezifische Ausgaben, wie `aws:createImage`, wo einige Aktionen es Ihnen ermöglichen, eigene Ausgaben zu definieren, z. B. `aws:executeAwsApi`.

Wenn Sie sich nicht sicher sind, wie Sie eine bestimmte Aktion in Ihrem Runbook verwenden, empfehlen wir Ihnen, den entsprechenden Eintrag für die Aktion im Systems Manager Automation Aktionen-Referenz nachzulesen. Wir empfehlen auch, den Inhalt vordefinierter Runbooks zu überprüfen, um Beispiele für die Verwendung dieser Aktionen zu sehen. Weitere Beispiele für Anwendungen von Runbooks in der Praxis finden Sie unter Weitere Runbook-Beispiele.

Um die Unterschiede in Bezug auf Einfachheit und Flexibilität zu demonstrieren, die Runbook-Inhalte bieten, bieten die folgenden Tutorials ein Beispiel, wie Sie Gruppen von Amazon-EC2-Instances stufenweise patchen:

- the section called “Beispiel 1: Erstellen von über- und untergeordneten Runbooks” – In diesem Beispiel werden zwei Runbooks in einer Untergeordnet-Übergeordnet-Beziehung verwendet. Das übergeordnete Runbook initiiert eine Automatisierung der Ratensteuerung des untergeordneten Runbooks.
- the section called “Beispiel 2: Skriptbasiertes Runbook” – Dieses Beispiel zeigt, wie Sie die gleichen Aufgaben von Beispiel 1 ausführen können, indem Sie den Inhalt zu einem einzigen Runbook zusammenfassen und Skripte in Ihrem Runbook verwenden.

Beispiel 1: Erstellen von über- und untergeordneten Runbooks

Das folgende Beispiel zeigt, wie Sie zwei Runbooks erstellen, die markierte Gruppen von Amazon Elastic Compute Cloud (Amazon EC2) -Instances schrittweise patchen. Diese Runbooks werden in einer Untergeordnet–Übergeordnet-Beziehung mit dem übergeordneten Runbook verwendet, das verwendet wird, um eine Kurssteuerungsautomatisierung des untergeordneten Runbooks zu initiieren. Weitere Informationen über die Ratenregelung-Automatisierungen finden Sie unter Automatisierte Abläufe in großem Umfang ausführen. Weitere Informationen zu den hier verwendeten Automation-Aktionen finden Sie unter Systems Manager Automation Aktionen-Referenz.

Erstellen des untergeordneten Runbooks

In diesem Beispiel-Runbook wird das folgende Szenario behandelt. Emily ist Systemingenieurin bei AnyCompany Consultants, LLC. Sie muss das Patching für Gruppen von Amazon Elastic Compute Cloud (Amazon EC2) -Instances konfigurieren, die primäre und sekundäre Datenbanken hosten. Anwendungen greifen 24 Stunden am Tag auf diese Datenbanken zu, sodass eine der Datenbankinstances immer verfügbar sein muss.

Sie entscheidet, dass das Patchen der Instances stufenweise der beste Ansatz ist. Die primäre Gruppe von Datenbankinstances wird zuerst gepatcht, gefolgt von der sekundären Gruppe von Datenbankinstances. Um zusätzliche Kosten zu vermeiden, indem Instances ausgeführt werden, die zuvor gestoppt wurden, möchte Emily außerdem, dass die gepatchten Instances in ihren ursprünglichen Zustand zurückversetzt werden, bevor das Patchen stattgefunden hat.

Emily identifiziert die primären und sekundären Gruppen von Datenbankinstances anhand der Tags, die den Instances zugeordnet sind. Sie beschließt, ein übergeordnetes Runbook zu erstellen, das eine Automatisierung der Ratenkontrolle eines untergeordneten Runbooks startet. Auf diese Weise kann sie die Tags ausrichten, die mit den primären und sekundären Gruppen von Datenbank-Instances verknüpft sind, und die Parallelität der untergeordneten Automatisierungen verwalten. Nachdem sie die verfügbaren Systems Manager (SSM)-Dokumente zum Patchen überprüft hat, wählt sie das `AWS-RunPatchBaseline`-Document. Mithilfe dieses SSM-Dokuments können ihre Kollegen die zugehörigen Patch-Compliance-Informationen überprüfen, nachdem der Patch-Vorgang abgeschlossen ist.

Um mit der Erstellung ihrer Runbook-Inhalte zu beginnen, überprüft Emily die verfügbaren Automatisierungsaktionen und beginnt mit der Erstellung des Inhalts für das untergeordnete Runbook wie folgt:

1. Zunächst stellt sie Werte für das Schema und die Beschreibung des Runbooks bereit und definiert die Eingabeparameter für das untergeordnete Runbook.

Durch die Verwendung des `AutomationAssumeRole`-Parameters können Emily und ihre Kollegen eine vorhandene IAM-Rolle verwenden, die der Automatisierung erlaubt, die Aktionen im Runbook für sie auszuführen. Emily verwendet das `InstanceId`-Parameter, um die Instance zu bestimmen, die gepatcht werden soll. Optional können die `Operation`-,`RebootOption`- , und`SnapshotId`-Parameter verwendet werden, um Werte für Dokumentparameter für `AWS-RunPatchBaseline` bereitzustellen. Um zu verhindern, dass für diese Dokumentparameter ungültige Werte bereitgestellt werden, definiert sie die `allowedValues` nach Bedarf.

YAML

```
schemaVersion: '0.3'
description: 'An example of an Automation runbook that patches groups of Amazon
 EC2 instances in stages.'
assumeRole: '{{AutomationAssumeRole}}'
parameters:
  AutomationAssumeRole:
    type: String
    description: >-
      '(Optional) The Amazon Resource Name (ARN) of the IAM role that allows
 Automation to perform the
      actions on your behalf. If no role is specified, Systems Manager
      Automation uses your IAM permissions to operate this runbook.'
    default: ''
  InstanceId:
    type: String
    description: >-
      '(Required) The instance you want to patch.'
  SnapshotId:
    type: String
    description: '(Optional) The snapshot ID to use to retrieve a patch baseline
 snapshot.'
    default: ''
  RebootOption:
    type: String
    description: '(Optional) Reboot behavior after a patch Install operation. If
 you choose NoReboot and patches are installed, the instance is marked as non-
compliant until a subsequent reboot and scan.'
    allowedValues:
      - NoReboot
      - RebootIfNeeded
    default: RebootIfNeeded
  Operation:
```

```
   type: String
   description: '(Optional) The update or configuration to perform on the
instance. The system checks if patches specified in the patch baseline are
installed on the instance. The install operation installs patches missing from
the baseline.'
   allowedValues:
     - Install
     - Scan
   default: Install
```

JSON

```
{
  "schemaVersion":"0.3",
  "description":"An example of an Automation runbook that patches groups of
Amazon EC2 instances in stages.",
  "assumeRole":"{{AutomationAssumeRole}}",
  "parameters":{
     "AutomationAssumeRole":{
        "type":"String",
        "description":"(Optional) The Amazon Resource Name (ARN) of the IAM role
that allows Automation to perform the actions on your behalf. If no role is
specified, Systems Manager Automation uses your IAM permissions to operate this
runbook.",
        "default":""
     },
     "InstanceId":{
        "type":"String",
        "description":"(Required) The instance you want to patch."
     },
     "SnapshotId":{
        "type":"String",
        "description":"(Optional) The snapshot ID to use to retrieve a patch
baseline snapshot.",
        "default":""
     },
     "RebootOption":{
        "type":"String",
        "description":"(Optional) Reboot behavior after a patch Install
operation. If you choose NoReboot and patches are installed, the instance is
marked as non-compliant until a subsequent reboot and scan.",
        "allowedValues":[
           "NoReboot",
```

```
            "RebootIfNeeded"
         ],
         "default":"RebootIfNeeded"
      },
      "Operation":{
         "type":"String",
         "description":"(Optional) The update or configuration to perform on
 the instance. The system checks if patches specified in the patch baseline are
 installed on the instance. The install operation installs patches missing from
 the baseline.",
         "allowedValues":[
            "Install",
            "Scan"
         ],
         "default":"Install"
      }
   }
},
```

2. Wenn die Elemente der obersten Ebene definiert sind, arbeitet Emily mit der Erstellung der Aktionen, welche die `mainSteps` des Runbooks melden. Der erste Schritt gibt den aktuellen Status der Ziel-Instance aus, die im `InstanceId`-Eingabeparameter mit der `aws:executeAwsApi`-Aktion angegeben ist. Die Ausgabe dieser Aktion wird in späteren Aktionen verwendet.

   YAML

```
mainSteps:
  - name: getInstanceState
    action: 'aws:executeAwsApi'
    onFailure: Abort
    inputs:
      inputs:
      Service: ec2
      Api: DescribeInstances
      InstanceIds:
        - '{{InstanceId}}'
    outputs:
      - Name: instanceState
        Selector: '$.Reservations[0].Instances[0].State.Name'
        Type: String
    nextStep: branchOnInstanceState
```

JSON

```
"mainSteps":[
      {
         "name":"getInstanceState",
         "action":"aws:executeAwsApi",
         "onFailure":"Abort",
         "inputs":{
            "inputs":null,
            "Service":"ec2",
            "Api":"DescribeInstances",
            "InstanceIds":[
               "{{InstanceId}}"
            ]
         },
         "outputs":[
            {
               "Name":"instanceState",
               "Selector":"$.Reservations[0].Instances[0].State.Name",
               "Type":"String"
            }
         ],
         "nextStep":"branchOnInstanceState"
      },
```

3. Anstatt den ursprünglichen Zustand jeder Instance, die gepatcht werden muss, manuell zu starten und zu verfolgen, verwendet Emily die Ausgabe der vorherigen Aktion, um die Automatisierung basierend auf dem Status der Ziel-Instance zu verzweigen. Auf diese Weise kann die Automatisierung verschiedene Schritte ausführen, abhängig von den Bedingungen, die in der `aws:branch`-Aktion angegeben sind und verbessert die Gesamteffizienz der Automatisierung ohne manuellen Eingriff.

   Wenn der Instance-Status bereits `running` ist, schreitet die Automatisierung mit dem Patchen der Instance mit dem `AWS-RunPatchBaseline`-Dokument unter Verwendung der `aws:runCommand`-Aktion fort.

   Wenn der Instancestatus `stopping` ist, fragt die Automatisierung ab, ob die Instance den Status `stopped` mit der Aktion `aws:waitForAwsResourceProperty` erreicht, startet die Instance mit der Aktion `executeAwsApi` und fragt die Instance ab, um den Status `running` zu erreichen, bevor die Instance gepatcht wird.

Wenn der Status der Instance `stopped` ist, startet die Automatisierung die Instance und fragt die Instance ab, einen `running`-Status vor dem Patchen der Instance unter Verwendung der gleichen Aktionen zu erreichen.

YAML

```
- name: branchOnInstanceState
    action: 'aws:branch'
    onFailure: Abort
    inputs:
      Choices:
        - NextStep: startInstance
          Variable: '{{getInstanceState.instanceState}}'
          StringEquals: stopped
         - NextStep: verifyInstanceStopped
          Variable: '{{getInstanceState.instanceState}}'
          StringEquals: stopping
         - NextStep: patchInstance
          Variable: '{{getInstanceState.instanceState}}'
          StringEquals: running
    isEnd: true
  - name: startInstance
    action: 'aws:executeAwsApi'
    onFailure: Abort
    inputs:
      Service: ec2
      Api: StartInstances
      InstanceIds:
        - '{{InstanceId}}'
    nextStep: verifyInstanceRunning
  - name: verifyInstanceRunning
    action: 'aws:waitForAwsResourceProperty'
    timeoutSeconds: 120
    inputs:
      Service: ec2
      Api: DescribeInstances
      InstanceIds:
        - '{{InstanceId}}'
      PropertySelector: '$.Reservations[0].Instances[0].State.Name'
      DesiredValues:
        - running
    nextStep: patchInstance
  - name: verifyInstanceStopped
```

```
    action: 'aws:waitForAwsResourceProperty'
    timeoutSeconds: 120
    inputs:
      Service: ec2
      Api: DescribeInstances
      InstanceIds:
        - '{{InstanceId}}'
      PropertySelector: '$.Reservations[0].Instances[0].State.Name'
      DesiredValues:
        - stopped
      nextStep: startInstance
  - name: patchInstance
    action: 'aws:runCommand'
    onFailure: Abort
    timeoutSeconds: 5400
    inputs:
      DocumentName: 'AWS-RunPatchBaseline'
      InstanceIds:
      - '{{InstanceId}}'
      Parameters:
        SnapshotId: '{{SnapshotId}}'
        RebootOption: '{{RebootOption}}'
        Operation: '{{Operation}}'
```

JSON

```
{
        "name":"branchOnInstanceState",
        "action":"aws:branch",
        "onFailure":"Abort",
        "inputs":{
           "Choices":[
              {
                 "NextStep":"startInstance",
                 "Variable":"{{getInstanceState.instanceState}}",
                 "StringEquals":"stopped"
              },
              {
                 "Or":[
                    {
                       "Variable":"{{getInstanceState.instanceState}}",
                       "StringEquals":"stopping"
                    }
```

```
                ],
                "NextStep":"verifyInstanceStopped"
             },
             {
                "NextStep":"patchInstance",
                "Variable":"{{getInstanceState.instanceState}}",
                "StringEquals":"running"
             }
          ]
       },
       "isEnd":true
    },
    {
       "name":"startInstance",
       "action":"aws:executeAwsApi",
       "onFailure":"Abort",
       "inputs":{
          "Service":"ec2",
          "Api":"StartInstances",
          "InstanceIds":[
             "{{InstanceId}}"
          ]
       },
       "nextStep":"verifyInstanceRunning"
    },
    {
       "name":"verifyInstanceRunning",
       "action":"aws:waitForAwsResourceProperty",
       "timeoutSeconds":120,
       "inputs":{
          "Service":"ec2",
          "Api":"DescribeInstances",
          "InstanceIds":[
             "{{InstanceId}}"
          ],
          "PropertySelector":"$.Reservations[0].Instances[0].State.Name",
          "DesiredValues":[
             "running"
          ]
       },
       "nextStep":"patchInstance"
    },
    {
       "name":"verifyInstanceStopped",
```

```
         "action":"aws:waitForAwsResourceProperty",
         "timeoutSeconds":120,
         "inputs":{
            "Service":"ec2",
            "Api":"DescribeInstances",
            "InstanceIds":[
               "{{InstanceId}}"
            ],
            "PropertySelector":"$.Reservations[0].Instances[0].State.Name",
            "DesiredValues":[
               "stopped"
            ],
            "nextStep":"startInstance"
         }
      },
      {
         "name":"patchInstance",
         "action":"aws:runCommand",
         "onFailure":"Abort",
         "timeoutSeconds":5400,
         "inputs":{
            "DocumentName":"AWS-RunPatchBaseline",
            "InstanceIds":[
               "{{InstanceId}}"
            ],
            "Parameters":{
               "SnapshotId":"{{SnapshotId}}",
               "RebootOption":"{{RebootOption}}",
               "Operation":"{{Operation}}"
            }
         }
      },
```

4. Nach Abschluss des Patching-Vorgangs möchte Emily, dass die Automatisierung die Ziel-Instance in denselben Zustand versetzt, in dem sie sich vor dem Automatisierungsstart befanden. Sie tut dies, indem sie erneut die Ausgabe der ersten Aktion verwendet. Die Automatisierung verzweigt sich basierend auf dem ursprünglichen Zustand der Ziel-Instance unter Verwendung der `aws:branch`-Aktion. Wenn sich die Instance zuvor in einem anderen Zustand als `running` befand, wird die Instance angehalten. Lautet der Status der Instance `running`, stoppt die Automatisierung.

YAML

```
- name: branchOnOriginalInstanceState
    action: 'aws:branch'
    onFailure: Abort
    inputs:
      Choices:
        - NextStep: stopInstance
          Not:
            Variable: '{{getInstanceState.instanceState}}'
            StringEquals: running
    isEnd: true
  - name: stopInstance
    action: 'aws:executeAwsApi'
    onFailure: Abort
    inputs:
      Service: ec2
      Api: StopInstances
      InstanceIds:
        - '{{InstanceId}}'
```

JSON

```
{
        "name":"branchOnOriginalInstanceState",
        "action":"aws:branch",
        "onFailure":"Abort",
        "inputs":{
           "Choices":[
              {
                 "NextStep":"stopInstance",
                 "Not":{
                    "Variable":"{{getInstanceState.instanceState}}",
                    "StringEquals":"running"
                 }
              }
           ]
        },
        "isEnd":true
     },
     {
        "name":"stopInstance",
        "action":"aws:executeAwsApi",
```

```
         "onFailure":"Abort",
         "inputs":{
            "Service":"ec2",
            "Api":"StopInstances",
            "InstanceIds":[
               "{{InstanceId}}"
            ]
         }
      }
   ]
}
```

5. Emily überprüft den vollständigen Inhalt des untergeordneten Runbooks und erstellt das Runbook in derselben AWS-Konto und AWS-Region wie die Ziel-Instances. Jetzt ist sie bereit, mit der Erstellung des übergeordneten Runbooks fortzufahren. Im Folgenden finden Sie den vollständigen untergeordneten Runbook-Inhalt.

   YAML

```
schemaVersion: '0.3'
description: 'An example of an Automation runbook that patches groups of Amazon
 EC2 instances in stages.'
assumeRole: '{{AutomationAssumeRole}}'
parameters:
  AutomationAssumeRole:
    type: String
    description: >-
      '(Optional) The Amazon Resource Name (ARN) of the IAM role that allows
 Automation to perform the
      actions on your behalf. If no role is specified, Systems Manager
      Automation uses your IAM permissions to operate this runbook.'
    default: ''
  InstanceId:
    type: String
    description: >-
      '(Required) The instance you want to patch.'
  SnapshotId:
    type: String
    description: '(Optional) The snapshot ID to use to retrieve a patch baseline
 snapshot.'
    default: ''
  RebootOption:
    type: String
```

```
    description: '(Optional) Reboot behavior after a patch Install operation. If
 you choose NoReboot and patches are installed, the instance is marked as non-
compliant until a subsequent reboot and scan.'
    allowedValues:
      - NoReboot
      - RebootIfNeeded
    default: RebootIfNeeded
  Operation:
    type: String
    description: '(Optional) The update or configuration to perform on the
 instance. The system checks if patches specified in the patch baseline are
 installed on the instance. The install operation installs patches missing from
 the baseline.'
    allowedValues:
      - Install
      - Scan
    default: Install
mainSteps:
  - name: getInstanceState
    action: 'aws:executeAwsApi'
    onFailure: Abort
    inputs:
      inputs:
      Service: ec2
      Api: DescribeInstances
      InstanceIds:
        - '{{InstanceId}}'
    outputs:
      - Name: instanceState
        Selector: '$.Reservations[0].Instances[0].State.Name'
        Type: String
    nextStep: branchOnInstanceState
  - name: branchOnInstanceState
    action: 'aws:branch'
    onFailure: Abort
    inputs:
      Choices:
        - NextStep: startInstance
          Variable: '{{getInstanceState.instanceState}}'
          StringEquals: stopped
        - Or:
            - Variable: '{{getInstanceState.instanceState}}'
              StringEquals: stopping
          NextStep: verifyInstanceStopped
```

```
      - NextStep: patchInstance
        Variable: '{{getInstanceState.instanceState}}'
        StringEquals: running
  isEnd: true
- name: startInstance
  action: 'aws:executeAwsApi'
  onFailure: Abort
  inputs:
    Service: ec2
    Api: StartInstances
    InstanceIds:
      - '{{InstanceId}}'
  nextStep: verifyInstanceRunning
- name: verifyInstanceRunning
  action: 'aws:waitForAwsResourceProperty'
  timeoutSeconds: 120
  inputs:
    Service: ec2
    Api: DescribeInstances
    InstanceIds:
      - '{{InstanceId}}'
    PropertySelector: '$.Reservations[0].Instances[0].State.Name'
    DesiredValues:
      - running
  nextStep: patchInstance
- name: verifyInstanceStopped
  action: 'aws:waitForAwsResourceProperty'
  timeoutSeconds: 120
  inputs:
    Service: ec2
    Api: DescribeInstances
    InstanceIds:
      - '{{InstanceId}}'
    PropertySelector: '$.Reservations[0].Instances[0].State.Name'
    DesiredValues:
      - stopped
    nextStep: startInstance
- name: patchInstance
  action: 'aws:runCommand'
  onFailure: Abort
  timeoutSeconds: 5400
  inputs:
    DocumentName: 'AWS-RunPatchBaseline'
    InstanceIds:
```

```
      - '{{InstanceId}}'
      Parameters:
        SnapshotId: '{{SnapshotId}}'
        RebootOption: '{{RebootOption}}'
        Operation: '{{Operation}}'
  - name: branchOnOriginalInstanceState
    action: 'aws:branch'
    onFailure: Abort
    inputs:
      Choices:
        - NextStep: stopInstance
          Not:
            Variable: '{{getInstanceState.instanceState}}'
            StringEquals: running
    isEnd: true
  - name: stopInstance
    action: 'aws:executeAwsApi'
    onFailure: Abort
    inputs:
      Service: ec2
      Api: StopInstances
      InstanceIds:
        - '{{InstanceId}}'
```

JSON

```
{
   "schemaVersion":"0.3",
   "description":"An example of an Automation runbook that patches groups of
 Amazon EC2 instances in stages.",
   "assumeRole":"{{AutomationAssumeRole}}",
   "parameters":{
      "AutomationAssumeRole":{
         "type":"String",
         "description":"'(Optional) The Amazon Resource Name (ARN) of the IAM
 role that allows Automation to perform the actions on your behalf. If no role is
 specified, Systems Manager Automation uses your IAM permissions to operate this
 runbook.'",
         "default":""
      },
      "InstanceId":{
         "type":"String",
         "description":"'(Required) The instance you want to patch.'"
```

```
      },
      "SnapshotId":{
         "type":"String",
         "description":"(Optional) The snapshot ID to use to retrieve a patch
baseline snapshot.",
         "default":""
      },
      "RebootOption":{
         "type":"String",
         "description":"(Optional) Reboot behavior after a patch Install
operation. If you choose NoReboot and patches are installed, the instance is
marked as non-compliant until a subsequent reboot and scan.",
         "allowedValues":[
            "NoReboot",
            "RebootIfNeeded"
         ],
         "default":"RebootIfNeeded"
      },
      "Operation":{
         "type":"String",
         "description":"(Optional) The update or configuration to perform on
the instance. The system checks if patches specified in the patch baseline are
installed on the instance. The install operation installs patches missing from
the baseline.",
         "allowedValues":[
            "Install",
            "Scan"
         ],
         "default":"Install"
      }
   },
   "mainSteps":[
      {
         "name":"getInstanceState",
         "action":"aws:executeAwsApi",
         "onFailure":"Abort",
         "inputs":{
            "inputs":null,
            "Service":"ec2",
            "Api":"DescribeInstances",
            "InstanceIds":[
               "{{InstanceId}}"
            ]
         },
```

```
         "outputs":[
            {
               "Name":"instanceState",
               "Selector":"$.Reservations[0].Instances[0].State.Name",
               "Type":"String"
            }
         ],
         "nextStep":"branchOnInstanceState"
      },
      {
         "name":"branchOnInstanceState",
         "action":"aws:branch",
         "onFailure":"Abort",
         "inputs":{
            "Choices":[
               {
                  "NextStep":"startInstance",
                  "Variable":"{{getInstanceState.instanceState}}",
                  "StringEquals":"stopped"
               },
               {
                  "Or":[
                     {
                        "Variable":"{{getInstanceState.instanceState}}",
                        "StringEquals":"stopping"
                     }
                  ],
                  "NextStep":"verifyInstanceStopped"
               },
               {
                  "NextStep":"patchInstance",
                  "Variable":"{{getInstanceState.instanceState}}",
                  "StringEquals":"running"
               }
            ]
         },
         "isEnd":true
      },
      {
         "name":"startInstance",
         "action":"aws:executeAwsApi",
         "onFailure":"Abort",
         "inputs":{
            "Service":"ec2",
```

```
            "Api":"StartInstances",
            "InstanceIds":[
               "{{InstanceId}}"
            ]
         },
         "nextStep":"verifyInstanceRunning"
      },
      {
         "name":"verifyInstanceRunning",
         "action":"aws:waitForAwsResourceProperty",
         "timeoutSeconds":120,
         "inputs":{
            "Service":"ec2",
            "Api":"DescribeInstances",
            "InstanceIds":[
               "{{InstanceId}}"
            ],
            "PropertySelector":"$.Reservations[0].Instances[0].State.Name",
            "DesiredValues":[
               "running"
            ]
         },
         "nextStep":"patchInstance"
      },
      {
         "name":"verifyInstanceStopped",
         "action":"aws:waitForAwsResourceProperty",
         "timeoutSeconds":120,
         "inputs":{
            "Service":"ec2",
            "Api":"DescribeInstances",
            "InstanceIds":[
               "{{InstanceId}}"
            ],
            "PropertySelector":"$.Reservations[0].Instances[0].State.Name",
            "DesiredValues":[
               "stopped"
            ],
            "nextStep":"startInstance"
         }
      },
      {
         "name":"patchInstance",
         "action":"aws:runCommand",
```

```
         "onFailure":"Abort",
         "timeoutSeconds":5400,
         "inputs":{
            "DocumentName":"AWS-RunPatchBaseline",
            "InstanceIds":[
               "{{InstanceId}}"
            ],
            "Parameters":{
               "SnapshotId":"{{SnapshotId}}",
               "RebootOption":"{{RebootOption}}",
               "Operation":"{{Operation}}"
            }
         }
      },
      {
         "name":"branchOnOriginalInstanceState",
         "action":"aws:branch",
         "onFailure":"Abort",
         "inputs":{
            "Choices":[
               {
                  "NextStep":"stopInstance",
                  "Not":{
                     "Variable":"{{getInstanceState.instanceState}}",
                     "StringEquals":"running"
                  }
               }
            ]
         },
         "isEnd":true
      },
      {
         "name":"stopInstance",
         "action":"aws:executeAwsApi",
         "onFailure":"Abort",
         "inputs":{
            "Service":"ec2",
            "Api":"StopInstances",
            "InstanceIds":[
               "{{InstanceId}}"
            ]
         }
      }
   ]
```

```
}
```

Weitere Informationen zu den hier verwendeten Automation-Aktionen finden Sie unter Systems Manager Automation Aktionen-Referenz.

Erstellen des übergeordneten Runbooks

In diesem Beispiel-Runbook wird das Szenario fortgesetzt, das im vorherigen Abschnitt beschrieben wird. Nachdem Emily nun das untergeordnete Runbook erstellt hat, beginnt sie mit der Erstellung des Inhalts für das übergeordnete Runbook wie folgt:

1. Zunächst stellt sie Werte für das Schema und die Beschreibung des Runbooks bereit und definiert die Eingabeparameter für das übergeordnete Runbook.

   Durch die Verwendung des `AutomationAssumeRole`-Parameters können Emily und ihre Kollegen eine vorhandene IAM-Rolle verwenden, die der Automatisierung erlaubt, die Aktionen im Runbook für sie auszuführen. Emily verwendet die `PatchGroupPrimaryKey`- und `PatchGroupPrimaryValue`-Parameter, um das Tag anzugeben, das mit der primären Gruppe von Datenbankinstances verknüpft ist, die gepatcht werden sollen. Sie verwendet den `PatchGroupSecondaryKey`- und `PatchGroupSecondaryValue`-Parameter, um das Tag anzugeben, das mit der sekundären Gruppe von Datenbankinstances verknüpft ist, die gepatcht werden sollen.

   YAML

   ```
   description: 'An example of an Automation runbook that patches groups of Amazon
    EC2 instances in stages.'
   schemaVersion: '0.3'
   assumeRole: '{{AutomationAssumeRole}}'
   parameters:
     AutomationAssumeRole:
       type: String
       description: '(Optional) The Amazon Resource Name (ARN) of the IAM role that
    allows Automation to perform the actions on your behalf. If no role is specified,
    Systems Manager Automation uses your IAM permissions to operate this runbook.'
       default: ''
     PatchGroupPrimaryKey:
       type: String
       description: '(Required) The key of the tag for the primary group of instances
    you want to patch.''
     PatchGroupPrimaryValue:
   ```

```
    type: String
    description: '(Required) The value of the tag for the primary group of
 instances you want to patch.'
  PatchGroupSecondaryKey:
    type: String
    description: '(Required) The key of the tag for the secondary group of
 instances you want to patch.'
  PatchGroupSecondaryValue:
    type: String
    description: '(Required) The value of the tag for the secondary group of
 instances you want to patch.'
```

JSON

```
{
   "schemaVersion": "0.3",
   "description": "An example of an Automation runbook that patches groups of
 Amazon EC2 instances in stages.",
   "assumeRole": "{{AutomationAssumeRole}}",
   "parameters": {
      "AutomationAssumeRole": {
         "type": "String",
         "description": "(Optional) The Amazon Resource Name (ARN) of the IAM
 role that allows Automation to perform the actions on your behalf. If no role is
 specified, Systems Manager Automation uses your IAM permissions to operate this
 runbook.",
         "default": ""
      },
      "PatchGroupPrimaryKey": {
         "type": "String",
         "description": "(Required) The key of the tag for the primary group of
 instances you want to patch."
      },
      "PatchGroupPrimaryValue": {
         "type": "String",
         "description": "(Required) The value of the tag for the primary group of
 instances you want to patch."
      },
      "PatchGroupSecondaryKey": {
         "type": "String",
         "description": "(Required) The key of the tag for the secondary group of
 instances you want to patch."
      },
```

```
      "PatchGroupSecondaryValue": {
         "type": "String",
         "description": "(Required) The value of the tag for the secondary group
 of instances you want to patch."
      }
   }
},
```

2. Wenn die Elemente der obersten Ebene definiert sind, arbeitet Emily mit der Erstellung der Aktionen, welche die `mainSteps` des Runbooks melden.

   Bei der ersten Aktion wird eine Ratensteuerungsautomatisierung mit dem soeben erstellten untergeordneten Runbook gestartet, das Instances betrifft, die mit dem Tag verknüpft sind, das in den `PatchGroupPrimaryKey`- und `PatchGroupPrimaryValue`-Eingabeparametern angegeben ist. Sie verwendet die Werte, die den Eingabeparametern zur Verfügung gestellt werden, um den Schlüssel und den Wert des Tags anzugeben, welcher der primären Gruppe von Datenbankinstances zugeordnet ist, die sie patchen möchte.

   Nach der Fertigstellung der ersten Automatisierung, startet die zweite Aktion eine andere Ratensteuerungsautomatisierung unter Verwendung des untergeordneten Runbooks, das Instances betrifft, die mit dem Tag verknüpft sind, das in den `PatchGroupSecondaryKey`- und `PatchGroupSecondaryValue`-Eingabeparametern angegeben ist. Sie verwendet die Werte, die den Eingabeparametern zur Verfügung gestellt werden, um den Schlüssel und den Wert des Tags anzugeben, welcher der sekundären Gruppe von Datenbankinstances zugeordnet ist, die sie patchen möchte.

   YAML

```
mainSteps:
  - name: patchPrimaryTargets
    action: 'aws:executeAutomation'
    onFailure: Abort
    timeoutSeconds: 7200
    inputs:
      DocumentName: RunbookTutorialChildAutomation
      Targets:
        - Key: 'tag:{{PatchGroupPrimaryKey}}'
          Values:
            - '{{PatchGroupPrimaryValue}}'
      TargetParameterName: 'InstanceId'
  - name: patchSecondaryTargets
    action: 'aws:executeAutomation'
```

```
    onFailure: Abort
    timeoutSeconds: 7200
    inputs:
      DocumentName: RunbookTutorialChildAutomation
      Targets:
        - Key: 'tag:{{PatchGroupSecondaryKey}}'
          Values:
            - '{{PatchGroupSecondaryValue}}'
      TargetParameterName: 'InstanceId'
```

JSON

```
"mainSteps":[
      {
         "name":"patchPrimaryTargets",
         "action":"aws:executeAutomation",
         "onFailure":"Abort",
         "timeoutSeconds":7200,
         "inputs":{
            "DocumentName":"RunbookTutorialChildAutomation",
            "Targets":[
               {
                  "Key":"tag:{{PatchGroupPrimaryKey}}",
                  "Values":[
                     "{{PatchGroupPrimaryValue}}"
                  ]
               }
            ],
            "TargetParameterName":"InstanceId"
         }
      },
      {
         "name":"patchSecondaryTargets",
         "action":"aws:executeAutomation",
         "onFailure":"Abort",
         "timeoutSeconds":7200,
         "inputs":{
            "DocumentName":"RunbookTutorialChildAutomation",
            "Targets":[
               {
                  "Key":"tag:{{PatchGroupSecondaryKey}}",
                  "Values":[
                     "{{PatchGroupSecondaryValue}}"
```

```
                ]
             }
          ],
          "TargetParameterName":"InstanceId"
       }
    }
  ]
}
```

3. Emily überprüft den vollständigen Inhalt des übergeordneten Runbooks und erstellt das Runbook in derselben AWS-Konto und in den Ziel-Instances AWS-Region . Jetzt ist sie bereit, ihre Runbooks zu testen, um sicherzustellen, dass die Automatisierung wie gewünscht funktioniert, bevor sie in ihre Produktionsumgebung implementiert werden. Im Folgenden finden Sie den vollständigen übergeordneten Runbook-Inhalt.

   YAML

```
description: An example of an Automation runbook that patches groups of Amazon EC2
 instances in stages.
schemaVersion: '0.3'
assumeRole: '{{AutomationAssumeRole}}'
parameters:
  AutomationAssumeRole:
    type: String
    description: '(Optional) The Amazon Resource Name (ARN) of the IAM role that
 allows Automation to perform the actions on your behalf. If no role is specified,
 Systems Manager Automation uses your IAM permissions to operate this runbook.'
    default: ''
  PatchGroupPrimaryKey:
    type: String
    description: (Required) The key of the tag for the primary group of instances
 you want to patch.
  PatchGroupPrimaryValue:
    type: String
    description: '(Required) The value of the tag for the primary group of
 instances you want to patch. '
  PatchGroupSecondaryKey:
    type: String
    description: (Required) The key of the tag for the secondary group of
 instances you want to patch.
  PatchGroupSecondaryValue:
    type: String
```

```
    description: '(Required) The value of the tag for the secondary group of
 instances you want to patch.  '
mainSteps:
  - name: patchPrimaryTargets
    action: 'aws:executeAutomation'
    onFailure: Abort
    timeoutSeconds: 7200
    inputs:
      DocumentName: RunbookTutorialChildAutomation
      Targets:
        - Key: 'tag:{{PatchGroupPrimaryKey}}'
          Values:
            - '{{PatchGroupPrimaryValue}}'
      TargetParameterName: 'InstanceId'
  - name: patchSecondaryTargets
    action: 'aws:executeAutomation'
    onFailure: Abort
    timeoutSeconds: 7200
    inputs:
      DocumentName: RunbookTutorialChildAutomation
      Targets:
        - Key: 'tag:{{PatchGroupSecondaryKey}}'
          Values:
            - '{{PatchGroupSecondaryValue}}'
      TargetParameterName: 'InstanceId'
```

JSON

```
{
   "description":"An example of an Automation runbook that patches groups of
 Amazon EC2 instances in stages.",
   "schemaVersion":"0.3",
   "assumeRole":"{{AutomationAssumeRole}}",
   "parameters":{
      "AutomationAssumeRole":{
         "type":"String",
         "description":"(Optional) The Amazon Resource Name (ARN) of the IAM role
 that allows Automation to perform the actions on your behalf. If no role is
 specified, Systems Manager Automation uses your IAM permissions to operate this
 runbook.",
         "default":""
      },
      "PatchGroupPrimaryKey":{
```

```
        "type":"String",
        "description":"(Required) The key of the tag for the primary group of
instances you want to patch."
     },
     "PatchGroupPrimaryValue":{
        "type":"String",
        "description":"(Required) The value of the tag for the primary group of
instances you want to patch. "
     },
     "PatchGroupSecondaryKey":{
        "type":"String",
        "description":"(Required) The key of the tag for the secondary group of
instances you want to patch."
     },
     "PatchGroupSecondaryValue":{
        "type":"String",
        "description":"(Required) The value of the tag for the secondary group of
instances you want to patch.  "
     }
  },
  "mainSteps":[
     {
        "name":"patchPrimaryTargets",
        "action":"aws:executeAutomation",
        "onFailure":"Abort",
        "timeoutSeconds":7200,
        "inputs":{
           "DocumentName":"RunbookTutorialChildAutomation",
           "Targets":[
              {
                 "Key":"tag:{{PatchGroupPrimaryKey}}",
                 "Values":[
                    "{{PatchGroupPrimaryValue}}"
                 ]
              }
           ],
           "TargetParameterName":"InstanceId"
        }
     },
     {
        "name":"patchSecondaryTargets",
        "action":"aws:executeAutomation",
        "onFailure":"Abort",
        "timeoutSeconds":7200,
```

```
         "inputs":{
            "DocumentName":"RunbookTutorialChildAutomation",
            "Targets":[
               {
                  "Key":"tag:{{PatchGroupSecondaryKey}}",
                  "Values":[
                     "{{PatchGroupSecondaryValue}}"
                  ]
               }
            ],
            "TargetParameterName":"InstanceId"
         }
      }
   ]
}
```

Weitere Informationen zu den hier verwendeten Automation-Aktionen finden Sie unter Systems Manager Automation Aktionen-Referenz.

Beispiel 2: Skriptbasiertes Runbook

In diesem Beispiel-Runbook wird das folgende Szenario behandelt. Emily ist Systemingenieurin bei AnyCompany Consultants, LLC. Zuvor hat sie zwei Runbooks erstellt, die in einer Eltern-Kind-Beziehung verwendet werden, um Gruppen von Amazon Elastic Compute Cloud (Amazon EC2) -Instances zu patchen, die primäre und sekundäre Datenbanken hosten. Anwendungen greifen 24 Stunden am Tag auf diese Datenbanken zu, sodass eine der Datenbankinstances immer verfügbar sein muss.

Basierend auf dieser Anforderung hat sie eine Lösung entwickelt, welche die Instances stufenweise mit dem `AWS-RunPatchBaseline`-Systems Manager (SSM)-Dokument patcht. Mithilfe dieses SSM-Dokuments können ihre Kollegen die zugehörigen Patch-Compliance-Informationen überprüfen, nachdem der Patch-Vorgang abgeschlossen ist.

Die primäre Gruppe von Datenbankinstances wird zuerst gepatcht, gefolgt von der sekundären Gruppe von Datenbankinstances. Um zusätzliche Kosten zu vermeiden, indem Instances ausgeführt werden, die zuvor gestoppt wurden, hat Emily sichergestellt, dass die Automatisierung die gepatchten Instances in ihren ursprünglichen Zustand zurückversetzte, bevor das Patchen stattgefunden hat. Emily verwendete Tags, die den primären und sekundären Gruppen von Datenbankinstances zugeordnet sind, um zu ermitteln, welche Instances in der gewünschten Reihenfolge gepatcht werden sollen.

Ihre bestehende automatisierte Lösung funktioniert, aber sie will ihre Lösung nach Möglichkeit verbessern. Um bei der Wartung des Runbook-Inhalts zu helfen und die Fehlerbehebung zu erleichtern, möchte sie die Automatisierung zu einem einzigen Runbook zusammenfassen und die Anzahl der Eingabeparameter vereinfachen. Außerdem möchte sie vermeiden, dass mehrere untergeordnete Automatisierungen erstellt werden.

Nachdem Emily die verfügbaren Automatisierungsaktionen überprüft hat, stellt sie fest, dass sie ihre Lösung mithilfe der `aws:executeScript`-Aktion verbessern kann, um ihre benutzerdefinierten Python-Skripte auszuführen. Sie beginnt nun mit der Erstellung des Inhalts für das Runbook wie folgt:

1. Zunächst stellt sie Werte für das Schema und die Beschreibung des Runbooks bereit und definiert die Eingabeparameter für das übergeordnete Runbook.

   Durch die Verwendung des `AutomationAssumeRole`-Parameters können Emily und ihre Kollegen eine vorhandene IAM-Rolle verwenden, die der Automatisierung erlaubt, die Aktionen im Runbook für sie auszuführen. Im Gegensatz zum Beispiel 1 ist der `AutomationAssumeRole`-Parameter ist jetzt erforderlich und nicht optional. Da dieses Runbook `aws:executeScript` Aktionen enthält, ist immer eine AWS Identity and Access Management (IAM-) Servicerolle (oder eine Rolle übernehmen) erforderlich. Diese Anforderung ist notwendig, da einige der Python-Skripte, die für die Aktionen angegeben sind, AWS -API-Operationen aufrufen.

   Emily verwendet die `PrimaryPatchGroupTag`- und `SecondaryPatchGroupTag`-Parameter, um die Tags anzugeben, die mit der primären und sekundären Gruppe von Datenbankinstances verknüpft sind, die gepatcht werden sollen. Um die erforderlichen Eingabeparameter zu vereinfachen, entscheidet sie sich, `StringMap`-Parameter anstatt mehrerer `String`-Parameter zu verwenden, wie sie im Runbook von Beispiel 1 verwendet wurde. Optional können die `Operation`-,`RebootOption`- , und`SnapshotId`-Parameter verwendet werden, um Werte für Dokumentparameter für `AWS-RunPatchBaseline` bereitzustellen. Um zu verhindern, dass für diese Dokumentparameter ungültige Werte bereitgestellt werden, definiert sie die `allowedValues` nach Bedarf.

   YAML

```
description: 'An example of an Automation runbook that patches groups of Amazon
 EC2 instances in stages.'
schemaVersion: '0.3'
assumeRole: '{{AutomationAssumeRole}}'
parameters:
  AutomationAssumeRole:
    type: String
```

```
    description: '(Required) The Amazon Resource Name (ARN) of the IAM role that
 allows Automation to perform the actions on your behalf. If no role is specified,
 Systems Manager Automation uses your IAM permissions to operate this runbook.'
  PrimaryPatchGroupTag:
    type: StringMap
    description: '(Required) The tag for the primary group of instances you want
 to patch. Specify a key-value pair. Example: {"key" : "value"}'
  SecondaryPatchGroupTag:
    type: StringMap
    description: '(Required) The tag for the secondary group of instances you want
 to patch. Specify a key-value pair. Example: {"key" : "value"}'
  SnapshotId:
    type: String
    description: '(Optional) The snapshot ID to use to retrieve a patch baseline
 snapshot.'
    default: ''
  RebootOption:
    type: String
    description: '(Optional) Reboot behavior after a patch Install operation. If
 you choose NoReboot and patches are installed, the instance is marked as non-
compliant until a subsequent reboot and scan.'
    allowedValues:
      - NoReboot
      - RebootIfNeeded
    default: RebootIfNeeded
  Operation:
    type: String
    description: '(Optional) The update or configuration to perform on the
 instance. The system checks if patches specified in the patch baseline are
 installed on the instance. The install operation installs patches missing from
 the baseline.'
    allowedValues:
      - Install
      - Scan
    default: Install
```

JSON

```
{
   "description":"An example of an Automation runbook that patches groups of
 Amazon EC2 instances in stages.",
   "schemaVersion":"0.3",
   "assumeRole":"{{AutomationAssumeRole}}",
```

```
  "parameters":{
     "AutomationAssumeRole":{
        "type":"String",
        "description":"(Required) The Amazon Resource Name (ARN) of the IAM role
that allows Automation to perform the actions on your behalf. If no role is
specified, Systems Manager Automation uses your IAM permissions to operate this
runbook."
     },
     "PrimaryPatchGroupTag":{
        "type":"StringMap",
        "description":"(Required) The tag for the primary group of instances you
want to patch. Specify a key-value pair. Example: {\"key\" : \"value\"}"
     },
     "SecondaryPatchGroupTag":{
        "type":"StringMap",
        "description":"(Required) The tag for the secondary group of instances
you want to patch. Specify a key-value pair. Example: {\"key\" : \"value\"}"
     },
     "SnapshotId":{
        "type":"String",
        "description":"(Optional) The snapshot ID to use to retrieve a patch
baseline snapshot.",
        "default":""
     },
     "RebootOption":{
        "type":"String",
        "description":"(Optional) Reboot behavior after a patch Install
operation. If you choose NoReboot and patches are installed, the instance is
marked as non-compliant until a subsequent reboot and scan.",
        "allowedValues":[
           "NoReboot",
           "RebootIfNeeded"
        ],
        "default":"RebootIfNeeded"
     },
     "Operation":{
        "type":"String",
        "description":"(Optional) The update or configuration to perform on
the instance. The system checks if patches specified in the patch baseline are
installed on the instance. The install operation installs patches missing from
the baseline.",
        "allowedValues":[
           "Install",
           "Scan"
```

```
         ],
         "default":"Install"
      }
   }
},
```

2. Wenn die Elemente der obersten Ebene definiert sind, arbeitet Emily mit der Erstellung der Aktionen, welche die `mainSteps` des Runbooks melden. Im ersten Schritt werden alle Instanzen erfasst, die IDs dem im Parameter angegebenen Tag zugeordnet sind, und es wird ein `PrimaryPatchGroupTag` `StringMap` Parameter ausgegeben, der die Instanz-ID und den aktuellen Status der Instanz enthält. Die Ausgabe dieser Aktion wird in späteren Aktionen verwendet.

   Beachten Sie, dass die `script`-Eingabeparameter für JSON-Runbooks nicht unterstützt werden. JSON-Runbooks müssen Skriptinhalt mithilfe des `attachment`-Eingabeparameters bereitstellen.

   YAML

```
mainSteps:
  - name: getPrimaryInstanceState
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: getInstanceStates
      InputPayload:
        primaryTag: '{{PrimaryPatchGroupTag}}'
      Script: |-
        def getInstanceStates(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          tag = events['primaryTag']
          tagKey, tagValue = list(tag.items())[0]
          instanceQuery = ec2.describe_instances(
          Filters=[
              {
                  "Name": "tag:" + tagKey,
                  "Values": [tagValue]
              }]
          )
```

```
          if not instanceQuery['Reservations']:
              noInstancesForTagString = "No instances found for specified tag."
              return({ 'noInstancesFound' : noInstancesForTagString })
          else:
              queryResponse = instanceQuery['Reservations']
              originalInstanceStates = {}
              for results in queryResponse:
                  instanceSet = results['Instances']
                  for instance in instanceSet:
                      instanceId = instance['InstanceId']
                      originalInstanceStates[instanceId] = instance['State']
['Name']
              return originalInstanceStates
    outputs:
      - Name: originalInstanceStates
        Selector: $.Payload
        Type: StringMap
    nextStep: verifyPrimaryInstancesRunning
```

JSON

```
"mainSteps":[
      {
         "name":"getPrimaryInstanceState",
         "action":"aws:executeScript",
         "timeoutSeconds":120,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.11",
            "Handler":"getInstanceStates",
            "InputPayload":{
               "primaryTag":"{{PrimaryPatchGroupTag}}"
            },
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"originalInstanceStates",
               "Selector":"$.Payload",
               "Type":"StringMap"
            }
         ],
         "nextStep":"verifyPrimaryInstancesRunning"
```

```
      },
```

3. Emily verwendet die Ausgabe der vorherigen Aktion in einer anderen `aws:executeScript`-Aktion, um zu überprüfen, ob alle Instances, die dem im `PrimaryPatchGroupTag`-Parameter angegebenen Tag zugeordnet sind, in einem `running`-Zustand sind.

   Wenn der Instance-Status bereits `running` oder `shutting-down` ist, durchläuft das Skript weiterhin die verbleibenden Instances.

   Wenn der Status der Instance `stopping` ist, fragt das Skript die Instance ab, den `stopped`-Status zu erreichen und startet die Instance.

   Wenn der Status der Instance `stopped` ist, startet das Skript die Instance.

   YAML

```
- name: verifyPrimaryInstancesRunning
    action: 'aws:executeScript'
    timeoutSeconds: 600
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: verifyInstancesRunning
      InputPayload:
        targetInstances: '{{getPrimaryInstanceState.originalInstanceStates}}'
      Script: |-
        def verifyInstancesRunning(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
            if instanceDict[instance] == 'stopped':
                print("The target instance " + instance + " is stopped. The
 instance will now be started.")
                ec2.start_instances(
                    InstanceIds=[instance]
                    )
            elif instanceDict[instance] == 'stopping':
                print("The target instance " + instance + " is stopping. Polling
 for instance to reach stopped state.")
                while instanceDict[instance] != 'stopped':
```

```
                    poll = ec2.get_waiter('instance_stopped')
                    poll.wait(
                        InstanceIds=[instance]
                    )
                ec2.start_instances(
                    InstanceIds=[instance]
                )
            else:
              pass
    nextStep: waitForPrimaryRunningInstances
```

JSON

```
{
         "name":"verifyPrimaryInstancesRunning",
         "action":"aws:executeScript",
         "timeoutSeconds":600,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.11",
            "Handler":"verifyInstancesRunning",
            "InputPayload":{

 "targetInstances":"{{getPrimaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"waitForPrimaryRunningInstances"
      },
```

4. Emily überprüft, ob alle Instances, die dem im `PrimaryPatchGroupTag`-Parameter angegebenen Tag zugeordnet sind, gestartet wurden oder sich bereits in einem `running`-Zustand befinden. Dann verwendet sie ein anderes Skript, um zu überprüfen, ob alle Instances, einschließlich derjenigen, die in der vorherigen Aktion gestartet wurden, den `running`-Zustand erreicht haben.

   YAML

```
- name: waitForPrimaryRunningInstances
    action: 'aws:executeScript'
    timeoutSeconds: 300
    onFailure: Abort
    inputs:
```

```
      Runtime: python3.11
      Handler: waitForRunningInstances
      InputPayload:
        targetInstances: '{{getPrimaryInstanceState.originalInstanceStates}}'
      Script: |-
        def waitForRunningInstances(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
              poll = ec2.get_waiter('instance_running')
              poll.wait(
                  InstanceIds=[instance]
              )
    nextStep: returnPrimaryTagKey
```

JSON

```
{
         "name":"waitForPrimaryRunningInstances",
         "action":"aws:executeScript",
         "timeoutSeconds":300,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.11",
            "Handler":"waitForRunningInstances",
            "InputPayload":{

 "targetInstances":"{{getPrimaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"returnPrimaryTagKey"
      },
```

5. Emily verwendet zwei weitere Skripte, um einzelne `String`-Werte des Schlüssels und des Werts des Tags zurückzugeben, das im `PrimaryPatchGroupTag`-Parameter angegeben ist. Die Werte, die von diesen Aktionen zurückgegeben werden, ermöglichen es ihr, Werte direkt für die `Targets`-Parameter für das `AWS-RunPatchBaseline`-Dokument bereitzustellen. Die

Automatisierung schreitet dann mit dem Patchen der Instance mit dem `AWS-RunPatchBaseline`-Dokument unter Verwendung der `aws:runCommand`-Aktion fort.

YAML

```
- name: returnPrimaryTagKey
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: returnTagValues
      InputPayload:
        primaryTag: '{{PrimaryPatchGroupTag}}'
      Script: |-
        def returnTagValues(events,context):
          tag = events['primaryTag']
          tagKey = list(tag)[0]
          stringKey = "tag:" + tagKey
          return {'tagKey' : stringKey}
    outputs:
      - Name: Payload
        Selector: $.Payload
        Type: StringMap
      - Name: primaryPatchGroupKey
        Selector: $.Payload.tagKey
        Type: String
    nextStep: returnPrimaryTagValue
  - name: returnPrimaryTagValue
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: returnTagValues
      InputPayload:
        primaryTag: '{{PrimaryPatchGroupTag}}'
      Script: |-
        def returnTagValues(events,context):
          tag = events['primaryTag']
          tagKey = list(tag)[0]
          tagValue = tag[tagKey]
          return {'tagValue' : tagValue}
    outputs:
```

```
      - Name: Payload
        Selector: $.Payload
        Type: StringMap
      - Name: primaryPatchGroupValue
        Selector: $.Payload.tagValue
        Type: String
    nextStep: patchPrimaryInstances
  - name: patchPrimaryInstances
    action: 'aws:runCommand'
    onFailure: Abort
    timeoutSeconds: 7200
    inputs:
      DocumentName: AWS-RunPatchBaseline
      Parameters:
        SnapshotId: '{{SnapshotId}}'
        RebootOption: '{{RebootOption}}'
        Operation: '{{Operation}}'
      Targets:
        - Key: '{{returnPrimaryTagKey.primaryPatchGroupKey}}'
          Values:
            - '{{returnPrimaryTagValue.primaryPatchGroupValue}}'
      MaxConcurrency: 10%
      MaxErrors: 10%
    nextStep: returnPrimaryToOriginalState
```

JSON

```
{
        "name":"returnPrimaryTagKey",
        "action":"aws:executeScript",
        "timeoutSeconds":120,
        "onFailure":"Abort",
        "inputs":{
           "Runtime":"python3.11",
           "Handler":"returnTagValues",
           "InputPayload":{
              "primaryTag":"{{PrimaryPatchGroupTag}}"
           },
           "Script":"..."
        },
        "outputs":[
           {
              "Name":"Payload",
```

```
                "Selector":"$.Payload",
                "Type":"StringMap"
            },
            {
                "Name":"primaryPatchGroupKey",
                "Selector":"$.Payload.tagKey",
                "Type":"String"
            }
        ],
        "nextStep":"returnPrimaryTagValue"
    },
    {
        "name":"returnPrimaryTagValue",
        "action":"aws:executeScript",
        "timeoutSeconds":120,
        "onFailure":"Abort",
        "inputs":{
            "Runtime":"python3.11",
            "Handler":"returnTagValues",
            "InputPayload":{
                "primaryTag":"{{PrimaryPatchGroupTag}}"
            },
            "Script":"..."
        },
        "outputs":[
            {
                "Name":"Payload",
                "Selector":"$.Payload",
                "Type":"StringMap"
            },
            {
                "Name":"primaryPatchGroupValue",
                "Selector":"$.Payload.tagValue",
                "Type":"String"
            }
        ],
        "nextStep":"patchPrimaryInstances"
    },
    {
        "name":"patchPrimaryInstances",
        "action":"aws:runCommand",
        "onFailure":"Abort",
        "timeoutSeconds":7200,
        "inputs":{
```

```
            "DocumentName":"AWS-RunPatchBaseline",
            "Parameters":{
               "SnapshotId":"{{SnapshotId}}",
               "RebootOption":"{{RebootOption}}",
               "Operation":"{{Operation}}"
            },
            "Targets":[
               {
                  "Key":"{{returnPrimaryTagKey.primaryPatchGroupKey}}",
                  "Values":[
                     "{{returnPrimaryTagValue.primaryPatchGroupValue}}"
                  ]
               }
            ],
            "MaxConcurrency":"10%",
            "MaxErrors":"10%"
         },
         "nextStep":"returnPrimaryToOriginalState"
      },
```

6. Nach Abschluss des Patching-Voragngs möchte Emily, dass die Automatisierung die Ziel-Instances, die dem im `PrimaryPatchGroupTag`-Parameter angegebenen Tag zugeordnet sind, in den Zustand zurückversetzt, in dem sie sich vor dem Automatisierungsstart befanden. Sie tut dies, indem sie erneut die Ausgabe der ersten Aktion in einem Skript verwendet. Basierend auf dem ursprünglichen Zustand der Ziel-Instance, wenn sich die Instance zuvor in einem anderen Zustand als `running` befand, wird die Instance angehalten. Wenn der Instance-Status bereits `running` ist, durchläuft das Skript weiterhin die verbleibenden Instances.

   YAML

```
- name: returnPrimaryToOriginalState
    action: 'aws:executeScript'
    timeoutSeconds: 600
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: returnToOriginalState
      InputPayload:
        targetInstances: '{{getPrimaryInstanceState.originalInstanceStates}}'
      Script: |-
        def returnToOriginalState(events,context):
          import boto3
```

```
          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
            if instanceDict[instance] == 'stopped' or instanceDict[instance] ==
 'stopping':
                ec2.stop_instances(
                    InstanceIds=[instance]
                    )
            else:
              pass
    nextStep: getSecondaryInstanceState
```

JSON

```
{
         "name":"returnPrimaryToOriginalState",
         "action":"aws:executeScript",
         "timeoutSeconds":600,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.11",
            "Handler":"returnToOriginalState",
            "InputPayload":{

 "targetInstances":"{{getPrimaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"getSecondaryInstanceState"
      },
```

7. Der Patchvorgang wird für die Instances abgeschlossen, die mit dem Tag verknüpft sind, das im `PrimaryPatchGroupTag`-Parameter angegeben ist. Jetzt dupliziert Emily alle vorherigen Aktionen in ihrem Runbook-Inhalt, um auf die Instances abzuzielen, die dem im `SecondaryPatchGroupTag`-Parameter angegebenen Tag zugeordnet sind.

   YAML

```
- name: getSecondaryInstanceState
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
```

```
    inputs:
      Runtime: python3.11
      Handler: getInstanceStates
      InputPayload:
        secondaryTag: '{{SecondaryPatchGroupTag}}'
      Script: |-
        def getInstanceStates(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          tag = events['secondaryTag']
          tagKey, tagValue = list(tag.items())[0]
          instanceQuery = ec2.describe_instances(
          Filters=[
              {
                  "Name": "tag:" + tagKey,
                  "Values": [tagValue]
              }]
          )
          if not instanceQuery['Reservations']:
              noInstancesForTagString = "No instances found for specified tag."
              return({ 'noInstancesFound' : noInstancesForTagString })
          else:
              queryResponse = instanceQuery['Reservations']
              originalInstanceStates = {}
              for results in queryResponse:
                  instanceSet = results['Instances']
                  for instance in instanceSet:
                      instanceId = instance['InstanceId']
                      originalInstanceStates[instanceId] = instance['State']
['Name']
              return originalInstanceStates
    outputs:
      - Name: originalInstanceStates
        Selector: $.Payload
        Type: StringMap
    nextStep: verifySecondaryInstancesRunning
  - name: verifySecondaryInstancesRunning
    action: 'aws:executeScript'
    timeoutSeconds: 600
    onFailure: Abort
    inputs:
      Runtime: python3.11
```

```
      Handler: verifyInstancesRunning
      InputPayload:
        targetInstances: '{{getSecondaryInstanceState.originalInstanceStates}}'
      Script: |-
        def verifyInstancesRunning(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
            if instanceDict[instance] == 'stopped':
                print("The target instance " + instance + " is stopped. The
instance will now be started.")
                ec2.start_instances(
                    InstanceIds=[instance]
                    )
            elif instanceDict[instance] == 'stopping':
                print("The target instance " + instance + " is stopping. Polling
for instance to reach stopped state.")
                while instanceDict[instance] != 'stopped':
                    poll = ec2.get_waiter('instance_stopped')
                    poll.wait(
                        InstanceIds=[instance]
                    )
                ec2.start_instances(
                    InstanceIds=[instance]
                )
            else:
              pass
    nextStep: waitForSecondaryRunningInstances
  - name: waitForSecondaryRunningInstances
    action: 'aws:executeScript'
    timeoutSeconds: 300
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: waitForRunningInstances
      InputPayload:
        targetInstances: '{{getSecondaryInstanceState.originalInstanceStates}}'
      Script: |-
        def waitForRunningInstances(events,context):
          import boto3
```

```
          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
              poll = ec2.get_waiter('instance_running')
              poll.wait(
                  InstanceIds=[instance]
              )
    nextStep: returnSecondaryTagKey
  - name: returnSecondaryTagKey
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: returnTagValues
      InputPayload:
        secondaryTag: '{{SecondaryPatchGroupTag}}'
      Script: |-
        def returnTagValues(events,context):
          tag = events['secondaryTag']
          tagKey = list(tag)[0]
          stringKey = "tag:" + tagKey
          return {'tagKey' : stringKey}
    outputs:
      - Name: Payload
        Selector: $.Payload
        Type: StringMap
      - Name: secondaryPatchGroupKey
        Selector: $.Payload.tagKey
        Type: String
    nextStep: returnSecondaryTagValue
  - name: returnSecondaryTagValue
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: returnTagValues
      InputPayload:
        secondaryTag: '{{SecondaryPatchGroupTag}}'
      Script: |-
        def returnTagValues(events,context):
          tag = events['secondaryTag']
```

```
          tagKey = list(tag)[0]
          tagValue = tag[tagKey]
          return {'tagValue' : tagValue}
    outputs:
      - Name: Payload
        Selector: $.Payload
        Type: StringMap
      - Name: secondaryPatchGroupValue
        Selector: $.Payload.tagValue
        Type: String
    nextStep: patchSecondaryInstances
  - name: patchSecondaryInstances
    action: 'aws:runCommand'
    onFailure: Abort
    timeoutSeconds: 7200
    inputs:
      DocumentName: AWS-RunPatchBaseline
      Parameters:
        SnapshotId: '{{SnapshotId}}'
        RebootOption: '{{RebootOption}}'
        Operation: '{{Operation}}'
      Targets:
        - Key: '{{returnSecondaryTagKey.secondaryPatchGroupKey}}'
          Values:
          - '{{returnSecondaryTagValue.secondaryPatchGroupValue}}'
      MaxConcurrency: 10%
      MaxErrors: 10%
    nextStep: returnSecondaryToOriginalState
  - name: returnSecondaryToOriginalState
    action: 'aws:executeScript'
    timeoutSeconds: 600
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: returnToOriginalState
      InputPayload:
        targetInstances: '{{getSecondaryInstanceState.originalInstanceStates}}'
      Script: |-
        def returnToOriginalState(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
```

```
          for instance in instanceDict:
            if instanceDict[instance] == 'stopped' or instanceDict[instance] ==
 'stopping':
                ec2.stop_instances(
                    InstanceIds=[instance]
                    )
            else:
              pass
```

JSON

```
{
         "name":"getSecondaryInstanceState",
         "action":"aws:executeScript",
         "timeoutSeconds":120,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.11",
            "Handler":"getInstanceStates",
            "InputPayload":{
               "secondaryTag":"{{SecondaryPatchGroupTag}}"
            },
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"originalInstanceStates",
               "Selector":"$.Payload",
               "Type":"StringMap"
            }
         ],
         "nextStep":"verifySecondaryInstancesRunning"
      },
      {
         "name":"verifySecondaryInstancesRunning",
         "action":"aws:executeScript",
         "timeoutSeconds":600,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.11",
            "Handler":"verifyInstancesRunning",
            "InputPayload":{
```

```
"targetInstances":"{{getSecondaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"waitForSecondaryRunningInstances"
      },
      {
         "name":"waitForSecondaryRunningInstances",
         "action":"aws:executeScript",
         "timeoutSeconds":300,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.11",
            "Handler":"waitForRunningInstances",
            "InputPayload":{

"targetInstances":"{{getSecondaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"returnSecondaryTagKey"
      },
      {
         "name":"returnSecondaryTagKey",
         "action":"aws:executeScript",
         "timeoutSeconds":120,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.11",
            "Handler":"returnTagValues",
            "InputPayload":{
               "secondaryTag":"{{SecondaryPatchGroupTag}}"
            },
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"Payload",
               "Selector":"$.Payload",
               "Type":"StringMap"
            },
            {
               "Name":"secondaryPatchGroupKey",
```

```
                "Selector":"$.Payload.tagKey",
                "Type":"String"
            }
         ],
         "nextStep":"returnSecondaryTagValue"
      },
      {
         "name":"returnSecondaryTagValue",
         "action":"aws:executeScript",
         "timeoutSeconds":120,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.11",
            "Handler":"returnTagValues",
            "InputPayload":{
               "secondaryTag":"{{SecondaryPatchGroupTag}}"
            },
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"Payload",
               "Selector":"$.Payload",
               "Type":"StringMap"
            },
            {
               "Name":"secondaryPatchGroupValue",
               "Selector":"$.Payload.tagValue",
               "Type":"String"
            }
         ],
         "nextStep":"patchSecondaryInstances"
      },
      {
         "name":"patchSecondaryInstances",
         "action":"aws:runCommand",
         "onFailure":"Abort",
         "timeoutSeconds":7200,
         "inputs":{
            "DocumentName":"AWS-RunPatchBaseline",
            "Parameters":{
               "SnapshotId":"{{SnapshotId}}",
               "RebootOption":"{{RebootOption}}",
               "Operation":"{{Operation}}"
```

```
            },
            "Targets":[
               {
                  "Key":"{{returnSecondaryTagKey.secondaryPatchGroupKey}}",
                  "Values":[
                     "{{returnSecondaryTagValue.secondaryPatchGroupValue}}"
                  ]
               }
            ],
            "MaxConcurrency":"10%",
            "MaxErrors":"10%"
         },
         "nextStep":"returnSecondaryToOriginalState"
      },
      {
         "name":"returnSecondaryToOriginalState",
         "action":"aws:executeScript",
         "timeoutSeconds":600,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.11",
            "Handler":"returnToOriginalState",
            "InputPayload":{

 "targetInstances":"{{getSecondaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         }
      }
   ]
}
```

8. Emily überprüft den fertigen skriptbasierten Runbook-Inhalt und erstellt das Runbook in derselben AWS-Konto und AWS-Region wie die Ziel-Instances. Jetzt ist sie bereit, ihr Runbook zu testen, um sicherzustellen, dass die Automatisierung wie gewünscht funktioniert, bevor es in ihre Produktionsumgebung implementiert wird. Im Folgenden finden Sie den vollständigen geskripteten Runbook-Inhalt.

   YAML

```
description: An example of an Automation runbook that patches groups of Amazon EC2
 instances in stages.
schemaVersion: '0.3'
```

```
assumeRole: '{{AutomationAssumeRole}}'
parameters:
  AutomationAssumeRole:
    type: String
    description: '(Required) The Amazon Resource Name (ARN) of the IAM role that
 allows Automation to perform the actions on your behalf. If no role is specified,
 Systems Manager Automation uses your IAM permissions to operate this runbook.'
  PrimaryPatchGroupTag:
    type: StringMap
    description: '(Required) The tag for the primary group of instances you want
 to patch. Specify a key-value pair. Example: {"key" : "value"}'
  SecondaryPatchGroupTag:
    type: StringMap
    description: '(Required) The tag for the secondary group of instances you want
 to patch. Specify a key-value pair. Example: {"key" : "value"}'
  SnapshotId:
    type: String
    description: '(Optional) The snapshot ID to use to retrieve a patch baseline
 snapshot.'
    default: ''
  RebootOption:
    type: String
    description: '(Optional) Reboot behavior after a patch Install operation. If
 you choose NoReboot and patches are installed, the instance is marked as non-
compliant until a subsequent reboot and scan.'
    allowedValues:
      - NoReboot
      - RebootIfNeeded
    default: RebootIfNeeded
  Operation:
    type: String
    description: '(Optional) The update or configuration to perform on the
 instance. The system checks if patches specified in the patch baseline are
 installed on the instance. The install operation installs patches missing from
 the baseline.'
    allowedValues:
      - Install
      - Scan
    default: Install
mainSteps:
  - name: getPrimaryInstanceState
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
```

```
    inputs:
      Runtime: python3.11
      Handler: getInstanceStates
      InputPayload:
        primaryTag: '{{PrimaryPatchGroupTag}}'
      Script: |-
        def getInstanceStates(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          tag = events['primaryTag']
          tagKey, tagValue = list(tag.items())[0]
          instanceQuery = ec2.describe_instances(
          Filters=[
              {
                  "Name": "tag:" + tagKey,
                  "Values": [tagValue]
              }]
          )
          if not instanceQuery['Reservations']:
              noInstancesForTagString = "No instances found for specified tag."
              return({ 'noInstancesFound' : noInstancesForTagString })
          else:
              queryResponse = instanceQuery['Reservations']
              originalInstanceStates = {}
              for results in queryResponse:
                  instanceSet = results['Instances']
                  for instance in instanceSet:
                      instanceId = instance['InstanceId']
                      originalInstanceStates[instanceId] = instance['State']
['Name']
              return originalInstanceStates
    outputs:
      - Name: originalInstanceStates
        Selector: $.Payload
        Type: StringMap
    nextStep: verifyPrimaryInstancesRunning
  - name: verifyPrimaryInstancesRunning
    action: 'aws:executeScript'
    timeoutSeconds: 600
    onFailure: Abort
    inputs:
      Runtime: python3.11
```

```
    Handler: verifyInstancesRunning
    InputPayload:
      targetInstances: '{{getPrimaryInstanceState.originalInstanceStates}}'
    Script: |-
      def verifyInstancesRunning(events,context):
        import boto3

        #Initialize client
        ec2 = boto3.client('ec2')
        instanceDict = events['targetInstances']
        for instance in instanceDict:
          if instanceDict[instance] == 'stopped':
              print("The target instance " + instance + " is stopped. The
instance will now be started.")
              ec2.start_instances(
                  InstanceIds=[instance]
                  )
          elif instanceDict[instance] == 'stopping':
              print("The target instance " + instance + " is stopping. Polling
for instance to reach stopped state.")
              while instanceDict[instance] != 'stopped':
                  poll = ec2.get_waiter('instance_stopped')
                  poll.wait(
                      InstanceIds=[instance]
                  )
              ec2.start_instances(
                  InstanceIds=[instance]
              )
          else:
            pass
  nextStep: waitForPrimaryRunningInstances
- name: waitForPrimaryRunningInstances
  action: 'aws:executeScript'
  timeoutSeconds: 300
  onFailure: Abort
  inputs:
    Runtime: python3.11
    Handler: waitForRunningInstances
    InputPayload:
      targetInstances: '{{getPrimaryInstanceState.originalInstanceStates}}'
    Script: |-
      def waitForRunningInstances(events,context):
        import boto3
```

```
          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
              poll = ec2.get_waiter('instance_running')
              poll.wait(
                  InstanceIds=[instance]
              )
    nextStep: returnPrimaryTagKey
  - name: returnPrimaryTagKey
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: returnTagValues
      InputPayload:
        primaryTag: '{{PrimaryPatchGroupTag}}'
      Script: |-
        def returnTagValues(events,context):
          tag = events['primaryTag']
          tagKey = list(tag)[0]
          stringKey = "tag:" + tagKey
          return {'tagKey' : stringKey}
    outputs:
      - Name: Payload
        Selector: $.Payload
        Type: StringMap
      - Name: primaryPatchGroupKey
        Selector: $.Payload.tagKey
        Type: String
    nextStep: returnPrimaryTagValue
  - name: returnPrimaryTagValue
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: returnTagValues
      InputPayload:
        primaryTag: '{{PrimaryPatchGroupTag}}'
      Script: |-
        def returnTagValues(events,context):
          tag = events['primaryTag']
```

```
          tagKey = list(tag)[0]
          tagValue = tag[tagKey]
          return {'tagValue' : tagValue}
    outputs:
      - Name: Payload
        Selector: $.Payload
        Type: StringMap
      - Name: primaryPatchGroupValue
        Selector: $.Payload.tagValue
        Type: String
    nextStep: patchPrimaryInstances
  - name: patchPrimaryInstances
    action: 'aws:runCommand'
    onFailure: Abort
    timeoutSeconds: 7200
    inputs:
      DocumentName: AWS-RunPatchBaseline
      Parameters:
        SnapshotId: '{{SnapshotId}}'
        RebootOption: '{{RebootOption}}'
        Operation: '{{Operation}}'
      Targets:
        - Key: '{{returnPrimaryTagKey.primaryPatchGroupKey}}'
          Values:
            - '{{returnPrimaryTagValue.primaryPatchGroupValue}}'
      MaxConcurrency: 10%
      MaxErrors: 10%
    nextStep: returnPrimaryToOriginalState
  - name: returnPrimaryToOriginalState
    action: 'aws:executeScript'
    timeoutSeconds: 600
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: returnToOriginalState
      InputPayload:
        targetInstances: '{{getPrimaryInstanceState.originalInstanceStates}}'
      Script: |-
        def returnToOriginalState(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
```

```
          for instance in instanceDict:
            if instanceDict[instance] == 'stopped' or instanceDict[instance] ==
'stopping':
                ec2.stop_instances(
                    InstanceIds=[instance]
                    )
            else:
              pass
    nextStep: getSecondaryInstanceState
  - name: getSecondaryInstanceState
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: getInstanceStates
      InputPayload:
        secondaryTag: '{{SecondaryPatchGroupTag}}'
      Script: |-
        def getInstanceStates(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          tag = events['secondaryTag']
          tagKey, tagValue = list(tag.items())[0]
          instanceQuery = ec2.describe_instances(
          Filters=[
              {
                  "Name": "tag:" + tagKey,
                  "Values": [tagValue]
              }]
          )
          if not instanceQuery['Reservations']:
              noInstancesForTagString = "No instances found for specified tag."
              return({ 'noInstancesFound' : noInstancesForTagString })
          else:
              queryResponse = instanceQuery['Reservations']
              originalInstanceStates = {}
              for results in queryResponse:
                  instanceSet = results['Instances']
                  for instance in instanceSet:
                      instanceId = instance['InstanceId']
```

```
                    originalInstanceStates[instanceId] = instance['State']
['Name']
            return originalInstanceStates
    outputs:
      - Name: originalInstanceStates
        Selector: $.Payload
        Type: StringMap
    nextStep: verifySecondaryInstancesRunning
  - name: verifySecondaryInstancesRunning
    action: 'aws:executeScript'
    timeoutSeconds: 600
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: verifyInstancesRunning
      InputPayload:
        targetInstances: '{{getSecondaryInstanceState.originalInstanceStates}}'
      Script: |-
        def verifyInstancesRunning(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
            if instanceDict[instance] == 'stopped':
                print("The target instance " + instance + " is stopped. The
 instance will now be started.")
                ec2.start_instances(
                    InstanceIds=[instance]
                    )
            elif instanceDict[instance] == 'stopping':
                print("The target instance " + instance + " is stopping. Polling
 for instance to reach stopped state.")
                while instanceDict[instance] != 'stopped':
                    poll = ec2.get_waiter('instance_stopped')
                    poll.wait(
                        InstanceIds=[instance]
                    )
                ec2.start_instances(
                    InstanceIds=[instance]
                )
            else:
              pass
```

```
    nextStep: waitForSecondaryRunningInstances
  - name: waitForSecondaryRunningInstances
    action: 'aws:executeScript'
    timeoutSeconds: 300
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: waitForRunningInstances
      InputPayload:
        targetInstances: '{{getSecondaryInstanceState.originalInstanceStates}}'
      Script: |-
        def waitForRunningInstances(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
              poll = ec2.get_waiter('instance_running')
              poll.wait(
                  InstanceIds=[instance]
              )
    nextStep: returnSecondaryTagKey
  - name: returnSecondaryTagKey
    action: 'aws:executeScript'
    timeoutSeconds: 120
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: returnTagValues
      InputPayload:
        secondaryTag: '{{SecondaryPatchGroupTag}}'
      Script: |-
        def returnTagValues(events,context):
          tag = events['secondaryTag']
          tagKey = list(tag)[0]
          stringKey = "tag:" + tagKey
          return {'tagKey' : stringKey}
    outputs:
      - Name: Payload
        Selector: $.Payload
        Type: StringMap
      - Name: secondaryPatchGroupKey
        Selector: $.Payload.tagKey
```

```
      Type: String
  nextStep: returnSecondaryTagValue
- name: returnSecondaryTagValue
  action: 'aws:executeScript'
  timeoutSeconds: 120
  onFailure: Abort
  inputs:
    Runtime: python3.11
    Handler: returnTagValues
    InputPayload:
      secondaryTag: '{{SecondaryPatchGroupTag}}'
    Script: |-
      def returnTagValues(events,context):
        tag = events['secondaryTag']
        tagKey = list(tag)[0]
        tagValue = tag[tagKey]
        return {'tagValue' : tagValue}
  outputs:
    - Name: Payload
      Selector: $.Payload
      Type: StringMap
    - Name: secondaryPatchGroupValue
      Selector: $.Payload.tagValue
      Type: String
  nextStep: patchSecondaryInstances
- name: patchSecondaryInstances
  action: 'aws:runCommand'
  onFailure: Abort
  timeoutSeconds: 7200
  inputs:
    DocumentName: AWS-RunPatchBaseline
    Parameters:
      SnapshotId: '{{SnapshotId}}'
      RebootOption: '{{RebootOption}}'
      Operation: '{{Operation}}'
    Targets:
      - Key: '{{returnSecondaryTagKey.secondaryPatchGroupKey}}'
        Values:
        - '{{returnSecondaryTagValue.secondaryPatchGroupValue}}'
    MaxConcurrency: 10%
    MaxErrors: 10%
  nextStep: returnSecondaryToOriginalState
- name: returnSecondaryToOriginalState
  action: 'aws:executeScript'
```

```
    timeoutSeconds: 600
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: returnToOriginalState
      InputPayload:
        targetInstances: '{{getSecondaryInstanceState.originalInstanceStates}}'
      Script: |-
        def returnToOriginalState(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2')
          instanceDict = events['targetInstances']
          for instance in instanceDict:
            if instanceDict[instance] == 'stopped' or instanceDict[instance] ==
 'stopping':
                ec2.stop_instances(
                    InstanceIds=[instance]
                    )
            else:
              pass
```

JSON

```
{
   "description":"An example of an Automation runbook that patches groups of
 Amazon EC2 instances in stages.",
   "schemaVersion":"0.3",
   "assumeRole":"{{AutomationAssumeRole}}",
   "parameters":{
      "AutomationAssumeRole":{
         "type":"String",
         "description":"(Required) The Amazon Resource Name (ARN) of the IAM role
 that allows Automation to perform the actions on your behalf. If no role is
 specified, Systems Manager Automation uses your IAM permissions to operate this
 runbook."
      },
      "PrimaryPatchGroupTag":{
         "type":"StringMap",
         "description":"(Required) The tag for the primary group of instances you
 want to patch. Specify a key-value pair. Example: {\"key\" : \"value\"}"
      },
```

```
    "SecondaryPatchGroupTag":{
       "type":"StringMap",
       "description":"(Required) The tag for the secondary group of instances
you want to patch. Specify a key-value pair. Example: {\"key\" : \"value\"}"
    },
    "SnapshotId":{
       "type":"String",
       "description":"(Optional) The snapshot ID to use to retrieve a patch
baseline snapshot.",
       "default":""
    },
    "RebootOption":{
       "type":"String",
       "description":"(Optional) Reboot behavior after a patch Install
operation. If you choose NoReboot and patches are installed, the instance is
marked as non-compliant until a subsequent reboot and scan.",
       "allowedValues":[
          "NoReboot",
          "RebootIfNeeded"
       ],
       "default":"RebootIfNeeded"
    },
    "Operation":{
       "type":"String",
       "description":"(Optional) The update or configuration to perform on
the instance. The system checks if patches specified in the patch baseline are
installed on the instance. The install operation installs patches missing from
the baseline.",
       "allowedValues":[
          "Install",
          "Scan"
       ],
       "default":"Install"
    }
  },
  "mainSteps":[
    {
       "name":"getPrimaryInstanceState",
       "action":"aws:executeScript",
       "timeoutSeconds":120,
       "onFailure":"Abort",
       "inputs":{
          "Runtime":"python3.11",
          "Handler":"getInstanceStates",
```

```
            "InputPayload":{
               "primaryTag":"{{PrimaryPatchGroupTag}}"
            },
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"originalInstanceStates",
               "Selector":"$.Payload",
               "Type":"StringMap"
            }
         ],
         "nextStep":"verifyPrimaryInstancesRunning"
      },
      {
         "name":"verifyPrimaryInstancesRunning",
         "action":"aws:executeScript",
         "timeoutSeconds":600,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.11",
            "Handler":"verifyInstancesRunning",
            "InputPayload":{

"targetInstances":"{{getPrimaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"waitForPrimaryRunningInstances"
      },
      {
         "name":"waitForPrimaryRunningInstances",
         "action":"aws:executeScript",
         "timeoutSeconds":300,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.11",
            "Handler":"waitForRunningInstances",
            "InputPayload":{

"targetInstances":"{{getPrimaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
```

```
            "nextStep":"returnPrimaryTagKey"
         },
         {
            "name":"returnPrimaryTagKey",
            "action":"aws:executeScript",
            "timeoutSeconds":120,
            "onFailure":"Abort",
            "inputs":{
               "Runtime":"python3.11",
               "Handler":"returnTagValues",
               "InputPayload":{
                  "primaryTag":"{{PrimaryPatchGroupTag}}"
               },
               "Script":"..."
            },
            "outputs":[
               {
                  "Name":"Payload",
                  "Selector":"$.Payload",
                  "Type":"StringMap"
               },
               {
                  "Name":"primaryPatchGroupKey",
                  "Selector":"$.Payload.tagKey",
                  "Type":"String"
               }
            ],
            "nextStep":"returnPrimaryTagValue"
         },
         {
            "name":"returnPrimaryTagValue",
            "action":"aws:executeScript",
            "timeoutSeconds":120,
            "onFailure":"Abort",
            "inputs":{
               "Runtime":"python3.11",
               "Handler":"returnTagValues",
               "InputPayload":{
                  "primaryTag":"{{PrimaryPatchGroupTag}}"
               },
               "Script":"..."
            },
            "outputs":[
               {
```

```
                "Name":"Payload",
                "Selector":"$.Payload",
                "Type":"StringMap"
            },
            {
                "Name":"primaryPatchGroupValue",
                "Selector":"$.Payload.tagValue",
                "Type":"String"
            }
        ],
        "nextStep":"patchPrimaryInstances"
    },
    {
        "name":"patchPrimaryInstances",
        "action":"aws:runCommand",
        "onFailure":"Abort",
        "timeoutSeconds":7200,
        "inputs":{
            "DocumentName":"AWS-RunPatchBaseline",
            "Parameters":{
                "SnapshotId":"{{SnapshotId}}",
                "RebootOption":"{{RebootOption}}",
                "Operation":"{{Operation}}"
            },
            "Targets":[
                {
                    "Key":"{{returnPrimaryTagKey.primaryPatchGroupKey}}",
                    "Values":[
                        "{{returnPrimaryTagValue.primaryPatchGroupValue}}"
                    ]
                }
            ],
            "MaxConcurrency":"10%",
            "MaxErrors":"10%"
        },
        "nextStep":"returnPrimaryToOriginalState"
    },
    {
        "name":"returnPrimaryToOriginalState",
        "action":"aws:executeScript",
        "timeoutSeconds":600,
        "onFailure":"Abort",
        "inputs":{
            "Runtime":"python3.11",
```

```
            "Handler":"returnToOriginalState",
            "InputPayload":{

"targetInstances":"{{getPrimaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"getSecondaryInstanceState"
      },
      {
         "name":"getSecondaryInstanceState",
         "action":"aws:executeScript",
         "timeoutSeconds":120,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.11",
            "Handler":"getInstanceStates",
            "InputPayload":{
               "secondaryTag":"{{SecondaryPatchGroupTag}}"
            },
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"originalInstanceStates",
               "Selector":"$.Payload",
               "Type":"StringMap"
            }
         ],
         "nextStep":"verifySecondaryInstancesRunning"
      },
      {
         "name":"verifySecondaryInstancesRunning",
         "action":"aws:executeScript",
         "timeoutSeconds":600,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.11",
            "Handler":"verifyInstancesRunning",
            "InputPayload":{

"targetInstances":"{{getSecondaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
```

```
         },
         "nextStep":"waitForSecondaryRunningInstances"
      },
      {
         "name":"waitForSecondaryRunningInstances",
         "action":"aws:executeScript",
         "timeoutSeconds":300,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.11",
            "Handler":"waitForRunningInstances",
            "InputPayload":{

"targetInstances":"{{getSecondaryInstanceState.originalInstanceStates}}"
            },
            "Script":"..."
         },
         "nextStep":"returnSecondaryTagKey"
      },
      {
         "name":"returnSecondaryTagKey",
         "action":"aws:executeScript",
         "timeoutSeconds":120,
         "onFailure":"Abort",
         "inputs":{
            "Runtime":"python3.11",
            "Handler":"returnTagValues",
            "InputPayload":{
               "secondaryTag":"{{SecondaryPatchGroupTag}}"
            },
            "Script":"..."
         },
         "outputs":[
            {
               "Name":"Payload",
               "Selector":"$.Payload",
               "Type":"StringMap"
            },
            {
               "Name":"secondaryPatchGroupKey",
               "Selector":"$.Payload.tagKey",
               "Type":"String"
            }
         ],
```

```
            "nextStep":"returnSecondaryTagValue"
         },
         {
            "name":"returnSecondaryTagValue",
            "action":"aws:executeScript",
            "timeoutSeconds":120,
            "onFailure":"Abort",
            "inputs":{
               "Runtime":"python3.11",
               "Handler":"returnTagValues",
               "InputPayload":{
                  "secondaryTag":"{{SecondaryPatchGroupTag}}"
               },
               "Script":"..."
            },
            "outputs":[
               {
                  "Name":"Payload",
                  "Selector":"$.Payload",
                  "Type":"StringMap"
               },
               {
                  "Name":"secondaryPatchGroupValue",
                  "Selector":"$.Payload.tagValue",
                  "Type":"String"
               }
            ],
            "nextStep":"patchSecondaryInstances"
         },
         {
            "name":"patchSecondaryInstances",
            "action":"aws:runCommand",
            "onFailure":"Abort",
            "timeoutSeconds":7200,
            "inputs":{
               "DocumentName":"AWS-RunPatchBaseline",
               "Parameters":{
                  "SnapshotId":"{{SnapshotId}}",
                  "RebootOption":"{{RebootOption}}",
                  "Operation":"{{Operation}}"
               },
               "Targets":[
                  {
                     "Key":"{{returnSecondaryTagKey.secondaryPatchGroupKey}}",
```

```
                    "Values":[
                       "{{returnSecondaryTagValue.secondaryPatchGroupValue}}"
                    ]
                 }
              ],
              "MaxConcurrency":"10%",
              "MaxErrors":"10%"
           },
           "nextStep":"returnSecondaryToOriginalState"
        },
        {
           "name":"returnSecondaryToOriginalState",
           "action":"aws:executeScript",
           "timeoutSeconds":600,
           "onFailure":"Abort",
           "inputs":{
              "Runtime":"python3.11",
              "Handler":"returnToOriginalState",
              "InputPayload":{

 "targetInstances":"{{getSecondaryInstanceState.originalInstanceStates}}"
              },
              "Script":"..."
           }
        }
     ]
}
```

Weitere Informationen zu den hier verwendeten Automation-Aktionen finden Sie unter Systems Manager Automation Aktionen-Referenz.

Weitere Runbook-Beispiele

Das folgende Beispiel-Runbook veranschaulicht, wie Sie mithilfe von AWS Systems Manager-Automatisierungsaktionen gängige Bereitstellungs-, Problembehandlungs- und Wartungsaufgaben automatisieren können.

> **Note**
>
> Die Beispiel-Runbooks in diesem Abschnitt werden bereitgestellt, um zu veranschaulichen, wie Sie benutzerdefinierte Runbooks erstellen können, um Ihre spezifischen Betriebsanforderungen zu erfüllen. Diese Runbooks sind nicht für den Einsatz in

Produktionsumgebungen vorgesehen. Sie können sie jedoch für Ihren eigenen Gebrauch anpassen.

Beispiele

- Bereitstellung der VPC-Architektur und der Microsoft Active Directory-Domänencontroller
- Wiederherstellen eines Root-Volumes aus dem letzten Snapshot
- Erstellen eines AMI und einer regionenübergreifenden Kopie

Bereitstellung der VPC-Architektur und der Microsoft Active Directory-Domänencontroller

Um die Effizienz zu steigern und allgemeine Aufgaben zu standardisieren, können Sie sich für die Automatisierung von Bereitstellungen entscheiden. Dies ist nützlich, wenn Sie regelmäßig dieselbe Architektur für mehrere Konten bereitstellen und AWS-Regionen. Durch die Automatisierung von Architekturbereitstellungen kann auch das Risiko menschlicher Fehler verringert werden, die bei der manuellen Bereitstellung der Architektur auftreten können. AWS Systems Manager Automatisierungsaktionen können Ihnen dabei helfen, dies zu erreichen. Automation ist ein Tool in AWS Systems Manager.

Das folgende AWS Systems Manager Beispiel-Runbook führt diese Aktionen aus:

- Ruft das neueste Windows Server 2016 Amazon Machine Image (AMI) mithilfe von Systems Manager abParameter Store, das beim Starten der EC2 Instanzen verwendet wird, die als Domänencontroller konfiguriert werden. Parameter Storeist ein Tool in AWS Systems Manager.
- Verwendet die `aws:executeAwsApi` Automatisierungsaktion, um mehrere AWS API-Operationen aufzurufen, um die VPC-Architektur zu erstellen. Die Domänencontroller-Instances werden in privaten Subnetzen gestartet und stellen über ein NAT-Gateway eine Verbindung zum Internet her. Dies ermöglicht dem SSM Agent auf den Instances auf die erforderlichen Systems Manager Endpunkte zuzugreifen.
- Verwendet die `aws:waitForAwsResourceProperty` Automatisierungsaktion, um zu bestätigen, dass die durch die vorherige Aktion gestarteten Instances `Online` für AWS Systems Manager gelten.
- Verwendet die `aws:runCommand` Automatisierungsaktion zur Konfiguration der als Microsoft Active Directory-Domänencontroller gestarteten Instances.

YAML

```
---
description: Custom Automation Deployment Example
schemaVersion: '0.3'
parameters:
  AutomationAssumeRole:
    type: String
    default: ''
    description: >-
      (Optional) The ARN of the role that allows Automation to perform the
      actions on your behalf. If no role is specified, Systems Manager
      Automation uses your IAM permissions to run this runbook.
mainSteps:
  - name: getLatestWindowsAmi
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ssm
      Api: GetParameter
      Name: >-
        /aws/service/ami-windows-latest/Windows_Server-2016-English-Full-Base
    outputs:
      - Name: amiId
        Selector: $.Parameter.Value
        Type: String
    nextStep: createSSMInstanceRole
  - name: createSSMInstanceRole
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: iam
      Api: CreateRole
      AssumeRolePolicyDocument: >-
        {"Version": "2012-10-17",       "Statement":
[{"Effect":"Allow","Principal":{"Service":["ec2.amazonaws.com"]},"Action":
["sts:AssumeRole"]}]}
      RoleName: sampleSSMInstanceRole
    nextStep: attachManagedSSMPolicy
  - name: attachManagedSSMPolicy
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
```

```
          Service: iam
          Api: AttachRolePolicy
          PolicyArn: 'arn:aws:iam::aws:policy/service-role/
AmazonSSMManagedInstanceCore'
          RoleName: sampleSSMInstanceRole
        nextStep: createSSMInstanceProfile
      - name: createSSMInstanceProfile
        action: aws:executeAwsApi
        onFailure: Abort
        inputs:
          Service: iam
          Api: CreateInstanceProfile
          InstanceProfileName: sampleSSMInstanceRole
        outputs:
          - Name: instanceProfileArn
            Selector: $.InstanceProfile.Arn
            Type: String
        nextStep: addSSMInstanceRoleToProfile
      - name: addSSMInstanceRoleToProfile
        action: aws:executeAwsApi
        onFailure: Abort
        inputs:
          Service: iam
          Api: AddRoleToInstanceProfile
          InstanceProfileName: sampleSSMInstanceRole
          RoleName: sampleSSMInstanceRole
        nextStep: createVpc
      - name: createVpc
        action: aws:executeAwsApi
        onFailure: Abort
        inputs:
          Service: ec2
          Api: CreateVpc
          CidrBlock: 10.0.100.0/22
        outputs:
          - Name: vpcId
            Selector: $.Vpc.VpcId
            Type: String
        nextStep: getMainRtb
      - name: getMainRtb
        action: aws:executeAwsApi
        onFailure: Abort
        inputs:
          Service: ec2
```

```
          Api: DescribeRouteTables
          Filters:
            - Name: vpc-id
              Values:
                - '{{ createVpc.vpcId }}'
        outputs:
          - Name: mainRtbId
            Selector: '$.RouteTables[0].RouteTableId'
            Type: String
        nextStep: verifyMainRtb
      - name: verifyMainRtb
        action: aws:assertAwsResourceProperty
        onFailure: Abort
        inputs:
          Service: ec2
          Api: DescribeRouteTables
          RouteTableIds:
            - '{{ getMainRtb.mainRtbId }}'
          PropertySelector: '$.RouteTables[0].Associations[0].Main'
          DesiredValues:
            - 'True'
        nextStep: createPubSubnet
      - name: createPubSubnet
        action: aws:executeAwsApi
        onFailure: Abort
        inputs:
          Service: ec2
          Api: CreateSubnet
          CidrBlock: 10.0.103.0/24
          AvailabilityZone: us-west-2c
          VpcId: '{{ createVpc.vpcId }}'
        outputs:
          - Name: pubSubnetId
            Selector: $.Subnet.SubnetId
            Type: String
        nextStep: createPubRtb
      - name: createPubRtb
        action: aws:executeAwsApi
        onFailure: Abort
        inputs:
          Service: ec2
          Api: CreateRouteTable
          VpcId: '{{ createVpc.vpcId }}'
        outputs:
```

```
        - Name: pubRtbId
          Selector: $.RouteTable.RouteTableId
          Type: String
      nextStep: createIgw
    - name: createIgw
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: CreateInternetGateway
      outputs:
        - Name: igwId
          Selector: $.InternetGateway.InternetGatewayId
          Type: String
      nextStep: attachIgw
    - name: attachIgw
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: AttachInternetGateway
        InternetGatewayId: '{{ createIgw.igwId }}'
        VpcId: '{{ createVpc.vpcId }}'
      nextStep: allocateEip
    - name: allocateEip
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: AllocateAddress
        Domain: vpc
      outputs:
        - Name: eipAllocationId
          Selector: $.AllocationId
          Type: String
      nextStep: createNatGw
    - name: createNatGw
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: CreateNatGateway
        AllocationId: '{{ allocateEip.eipAllocationId }}'
        SubnetId: '{{ createPubSubnet.pubSubnetId }}'
```

```
    outputs:
      - Name: natGwId
        Selector: $.NatGateway.NatGatewayId
        Type: String
    nextStep: verifyNatGwAvailable
  - name: verifyNatGwAvailable
    action: aws:waitForAwsResourceProperty
    timeoutSeconds: 150
    inputs:
      Service: ec2
      Api: DescribeNatGateways
      NatGatewayIds:
        - '{{ createNatGw.natGwId }}'
      PropertySelector: '$.NatGateways[0].State'
      DesiredValues:
        - available
    nextStep: createNatRoute
  - name: createNatRoute
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: CreateRoute
      DestinationCidrBlock: 0.0.0.0/0
      NatGatewayId: '{{ createNatGw.natGwId }}'
      RouteTableId: '{{ getMainRtb.mainRtbId }}'
    nextStep: createPubRoute
  - name: createPubRoute
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: CreateRoute
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: '{{ createIgw.igwId }}'
      RouteTableId: '{{ createPubRtb.pubRtbId }}'
    nextStep: setPubSubAssoc
  - name: setPubSubAssoc
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: AssociateRouteTable
      RouteTableId: '{{ createPubRtb.pubRtbId }}'
```

```
      SubnetId: '{{ createPubSubnet.pubSubnetId }}'
  - name: createDhcpOptions
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: CreateDhcpOptions
      DhcpConfigurations:
        - Key: domain-name-servers
          Values:
            - '10.0.100.50,10.0.101.50'
        - Key: domain-name
          Values:
            - sample.com
    outputs:
      - Name: dhcpOptionsId
        Selector: $.DhcpOptions.DhcpOptionsId
        Type: String
    nextStep: createDCSubnet1
  - name: createDCSubnet1
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: CreateSubnet
      CidrBlock: 10.0.100.0/24
      AvailabilityZone: us-west-2a
      VpcId: '{{ createVpc.vpcId }}'
    outputs:
      - Name: firstSubnetId
        Selector: $.Subnet.SubnetId
        Type: String
    nextStep: createDCSubnet2
  - name: createDCSubnet2
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: CreateSubnet
      CidrBlock: 10.0.101.0/24
      AvailabilityZone: us-west-2b
      VpcId: '{{ createVpc.vpcId }}'
    outputs:
      - Name: secondSubnetId
```

```
        Selector: $.Subnet.SubnetId
        Type: String
    nextStep: createDCSecGroup
  - name: createDCSecGroup
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: CreateSecurityGroup
      GroupName: SampleDCSecGroup
      Description: Security Group for Sample Domain Controllers
      VpcId: '{{ createVpc.vpcId }}'
    outputs:
      - Name: dcSecGroupId
        Selector: $.GroupId
        Type: String
    nextStep: authIngressDCTraffic
  - name: authIngressDCTraffic
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: AuthorizeSecurityGroupIngress
      GroupId: '{{ createDCSecGroup.dcSecGroupId }}'
      IpPermissions:
        - FromPort: -1
          IpProtocol: '-1'
          IpRanges:
            - CidrIp: 0.0.0.0/0
              Description: Allow all traffic between Domain Controllers
    nextStep: verifyInstanceProfile
  - name: verifyInstanceProfile
    action: aws:waitForAwsResourceProperty
    maxAttempts: 5
    onFailure: Abort
    inputs:
      Service: iam
      Api: ListInstanceProfilesForRole
      RoleName: sampleSSMInstanceRole
      PropertySelector: '$.InstanceProfiles[0].Arn'
      DesiredValues:
        - '{{ createSSMInstanceProfile.instanceProfileArn }}'
    nextStep: iamEventualConsistency
  - name: iamEventualConsistency
```

```
    action: aws:sleep
    inputs:
      Duration: PT2M
    nextStep: launchDC1
  - name: launchDC1
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: RunInstances
      BlockDeviceMappings:
        - DeviceName: /dev/sda1
          Ebs:
            DeleteOnTermination: true
            VolumeSize: 50
            VolumeType: gp2
        - DeviceName: xvdf
          Ebs:
            DeleteOnTermination: true
            VolumeSize: 100
            VolumeType: gp2
      IamInstanceProfile:
        Arn: '{{ createSSMInstanceProfile.instanceProfileArn }}'
      ImageId: '{{ getLatestWindowsAmi.amiId }}'
      InstanceType: t2.micro
      MaxCount: 1
      MinCount: 1
      PrivateIpAddress: 10.0.100.50
      SecurityGroupIds:
        - '{{ createDCSecGroup.dcSecGroupId }}'
      SubnetId: '{{ createDCSubnet1.firstSubnetId }}'
      TagSpecifications:
        - ResourceType: instance
          Tags:
            - Key: Name
              Value: SampleDC1
    outputs:
      - Name: pdcInstanceId
        Selector: '$.Instances[0].InstanceId'
        Type: String
    nextStep: launchDC2
  - name: launchDC2
    action: aws:executeAwsApi
    onFailure: Abort
```

```
        inputs:
          Service: ec2
          Api: RunInstances
          BlockDeviceMappings:
            - DeviceName: /dev/sda1
              Ebs:
                DeleteOnTermination: true
                VolumeSize: 50
                VolumeType: gp2
            - DeviceName: xvdf
              Ebs:
                DeleteOnTermination: true
                VolumeSize: 100
                VolumeType: gp2
          IamInstanceProfile:
            Arn: '{{ createSSMInstanceProfile.instanceProfileArn }}'
          ImageId: '{{ getLatestWindowsAmi.amiId }}'
          InstanceType: t2.micro
          MaxCount: 1
          MinCount: 1
          PrivateIpAddress: 10.0.101.50
          SecurityGroupIds:
            - '{{ createDCSecGroup.dcSecGroupId }}'
          SubnetId: '{{ createDCSubnet2.secondSubnetId }}'
          TagSpecifications:
            - ResourceType: instance
              Tags:
                - Key: Name
                  Value: SampleDC2
        outputs:
          - Name: adcInstanceId
            Selector: '$.Instances[0].InstanceId'
            Type: String
        nextStep: verifyDCInstanceState
      - name: verifyDCInstanceState
        action: aws:waitForAwsResourceProperty
        inputs:
          Service: ec2
          Api: DescribeInstanceStatus
          IncludeAllInstances: true
          InstanceIds:
            - '{{ launchDC1.pdcInstanceId }}'
            - '{{ launchDC2.adcInstanceId }}'
          PropertySelector: '$.InstanceStatuses..InstanceState.Name'
```

```
          DesiredValues:
            - running
        nextStep: verifyInstancesOnlineSSM
      - name: verifyInstancesOnlineSSM
        action: aws:waitForAwsResourceProperty
        timeoutSeconds: 600
        inputs:
          Service: ssm
          Api: DescribeInstanceInformation
          InstanceInformationFilterList:
            - key: InstanceIds
              valueSet:
                - '{{ launchDC1.pdcInstanceId }}'
                - '{{ launchDC2.adcInstanceId }}'
          PropertySelector: '$.InstanceInformationList..PingStatus'
          DesiredValues:
            - Online
        nextStep: installADRoles
      - name: installADRoles
        action: aws:runCommand
        inputs:
          DocumentName: AWS-RunPowerShellScript
          InstanceIds:
            - '{{ launchDC1.pdcInstanceId }}'
            - '{{ launchDC2.adcInstanceId }}'
          Parameters:
            commands: |-
              try {
                  Install-WindowsFeature -Name AD-Domain-Services -
IncludeManagementTools
              }
              catch {
                  Write-Error "Failed to install ADDS Role."
              }
        nextStep: setAdminPassword
      - name: setAdminPassword
        action: aws:runCommand
        inputs:
          DocumentName: AWS-RunPowerShellScript
          InstanceIds:
            - '{{ launchDC1.pdcInstanceId }}'
          Parameters:
            commands:
              - net user Administrator "sampleAdminPass123!"
```

```
        nextStep: createForest
      - name: createForest
        action: aws:runCommand
        inputs:
          DocumentName: AWS-RunPowerShellScript
          InstanceIds:
            - '{{ launchDC1.pdcInstanceId }}'
          Parameters:
            commands: |-
              $dsrmPass = 'sample123!' | ConvertTo-SecureString -asPlainText -Force
              try {
                  Install-ADDSForest -DomainName "sample.com" -DomainMode 6
 -ForestMode 6 -InstallDNS -DatabasePath "D:\NTDS" -SysvolPath "D:\SYSVOL" -
SafeModeAdministratorPassword $dsrmPass -Force
              }
              catch {
                  Write-Error $_
              }
              try {
                  Add-DnsServerForwarder -IPAddress "10.0.100.2"
              }
              catch {
                  Write-Error $_
              }
        nextStep: associateDhcpOptions
      - name: associateDhcpOptions
        action: aws:executeAwsApi
        onFailure: Abort
        inputs:
          Service: ec2
          Api: AssociateDhcpOptions
          DhcpOptionsId: '{{ createDhcpOptions.dhcpOptionsId }}'
          VpcId: '{{ createVpc.vpcId }}'
        nextStep: waitForADServices
      - name: waitForADServices
        action: aws:sleep
        inputs:
          Duration: PT1M
        nextStep: promoteADC
      - name: promoteADC
        action: aws:runCommand
        inputs:
          DocumentName: AWS-RunPowerShellScript
          InstanceIds:
```

```
            - '{{ launchDC2.adcInstanceId }}'
          Parameters:
            commands: |-
              ipconfig /renew
              $dsrmPass = 'sample123!' | ConvertTo-SecureString -asPlainText -Force
              $domAdminUser = "sample\Administrator"
              $domAdminPass = "sampleAdminPass123!" | ConvertTo-SecureString -
asPlainText -Force
              $domAdminCred = New-Object
 System.Management.Automation.PSCredential($domAdminUser,$domAdminPass)

              try {
                  Install-ADDSDomainController -DomainName "sample.com" -InstallDNS
 -DatabasePath "D:\NTDS" -SysvolPath "D:\SYSVOL" -SafeModeAdministratorPassword
 $dsrmPass -Credential $domAdminCred -Force
              }
              catch {
                  Write-Error $_
              }
```

JSON

```
{
      "description": "Custom Automation Deployment Example",
      "schemaVersion": "0.3",
      "assumeRole": "{{ AutomationAssumeRole }}",
      "parameters": {
        "AutomationAssumeRole": {
          "type": "String",
          "description": "(Optional) The ARN of the role that allows Automation
 to perform the actions on your behalf. If no role is specified, Systems Manager
 Automation uses your IAM permissions to run this runbook.",
          "default": ""
        }
      },
      "mainSteps": [
        {
          "name": "getLatestWindowsAmi",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ssm",
            "Api": "GetParameter",
```

```
            "Name": "/aws/service/ami-windows-latest/Windows_Server-2016-English-
Full-Base"
          },
          "outputs": [
            {
              "Name": "amiId",
              "Selector": "$.Parameter.Value",
              "Type": "String"
            }
          ],
          "nextStep": "createSSMInstanceRole"
        },
        {
          "name": "createSSMInstanceRole",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "iam",
            "Api": "CreateRole",
            "AssumeRolePolicyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":
[{\"Effect\":\"Allow\",\"Principal\":{\"Service\":[\"ec2.amazonaws.com\"]},\"Action
\":[\"sts:AssumeRole\"]}]}",
            "RoleName": "sampleSSMInstanceRole"
          },
          "nextStep": "attachManagedSSMPolicy"
        },
        {
          "name": "attachManagedSSMPolicy",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "iam",
            "Api": "AttachRolePolicy",
            "PolicyArn": "arn:aws:iam::aws:policy/service-role/
AmazonSSMManagedInstanceCore",
            "RoleName": "sampleSSMInstanceRole"
          },
          "nextStep": "createSSMInstanceProfile"
        },
        {
          "name": "createSSMInstanceProfile",
          "action":"aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
```

```
            "Service": "iam",
            "Api": "CreateInstanceProfile",
            "InstanceProfileName": "sampleSSMInstanceRole"
          },
          "outputs": [
            {
              "Name": "instanceProfileArn",
              "Selector": "$.InstanceProfile.Arn",
              "Type": "String"
            }
          ],
          "nextStep": "addSSMInstanceRoleToProfile"
        },
        {
          "name": "addSSMInstanceRoleToProfile",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "iam",
            "Api": "AddRoleToInstanceProfile",
            "InstanceProfileName": "sampleSSMInstanceRole",
            "RoleName": "sampleSSMInstanceRole"
          },
          "nextStep": "createVpc"
        },
        {
          "name": "createVpc",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "CreateVpc",
            "CidrBlock": "10.0.100.0/22"
          },
          "outputs": [
            {
              "Name": "vpcId",
              "Selector": "$.Vpc.VpcId",
              "Type": "String"
            }
          ],
          "nextStep": "getMainRtb"
        },
        {
```

```
            "name": "getMainRtb",
            "action": "aws:executeAwsApi",
            "onFailure": "Abort",
            "inputs": {
              "Service": "ec2",
              "Api": "DescribeRouteTables",
              "Filters": [
                {
                  "Name": "vpc-id",
                  "Values": ["{{ createVpc.vpcId }}"]
                }
              ]
            },
            "outputs": [
              {
                "Name": "mainRtbId",
                "Selector": "$.RouteTables[0].RouteTableId",
                "Type": "String"
              }
            ],
            "nextStep": "verifyMainRtb"
          },
          {
            "name": "verifyMainRtb",
            "action": "aws:assertAwsResourceProperty",
            "onFailure": "Abort",
            "inputs": {
              "Service": "ec2",
              "Api": "DescribeRouteTables",
              "RouteTableIds": ["{{ getMainRtb.mainRtbId }}"],
              "PropertySelector": "$.RouteTables[0].Associations[0].Main",
              "DesiredValues": ["True"]
            },
            "nextStep": "createPubSubnet"
          },
          {
            "name": "createPubSubnet",
            "action": "aws:executeAwsApi",
            "onFailure": "Abort",
            "inputs": {
              "Service": "ec2",
              "Api": "CreateSubnet",
              "CidrBlock": "10.0.103.0/24",
              "AvailabilityZone": "us-west-2c",
```

```
            "VpcId": "{{ createVpc.vpcId }}"
          },
          "outputs":[
            {
              "Name": "pubSubnetId",
              "Selector": "$.Subnet.SubnetId",
              "Type": "String"
            }
          ],
          "nextStep": "createPubRtb"
        },
        {
          "name": "createPubRtb",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "CreateRouteTable",
            "VpcId": "{{ createVpc.vpcId }}"
          },
          "outputs": [
            {
              "Name": "pubRtbId",
              "Selector": "$.RouteTable.RouteTableId",
              "Type": "String"
            }
          ],
          "nextStep": "createIgw"
        },
        {
          "name": "createIgw",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "CreateInternetGateway"
          },
          "outputs": [
            {
              "Name": "igwId",
              "Selector": "$.InternetGateway.InternetGatewayId",
              "Type": "String"
            }
          ],
```

```
            "nextStep": "attachIgw"
          },
          {
            "name": "attachIgw",
            "action": "aws:executeAwsApi",
            "onFailure": "Abort",
            "inputs": {
              "Service": "ec2",
              "Api": "AttachInternetGateway",
              "InternetGatewayId": "{{ createIgw.igwId }}",
              "VpcId": "{{ createVpc.vpcId }}"
            },
            "nextStep": "allocateEip"
          },
          {
            "name": "allocateEip",
            "action": "aws:executeAwsApi",
            "onFailure": "Abort",
            "inputs": {
              "Service": "ec2",
              "Api": "AllocateAddress",
              "Domain": "vpc"
            },
            "outputs": [
              {
                "Name": "eipAllocationId",
                "Selector": "$.AllocationId",
                "Type": "String"
              }
            ],
            "nextStep": "createNatGw"
          },
          {
            "name": "createNatGw",
            "action": "aws:executeAwsApi",
            "onFailure": "Abort",
            "inputs": {
              "Service": "ec2",
              "Api": "CreateNatGateway",
              "AllocationId": "{{ allocateEip.eipAllocationId }}",
              "SubnetId": "{{ createPubSubnet.pubSubnetId }}"
            },
            "outputs":[
              {
```

```
            "Name": "natGwId",
            "Selector": "$.NatGateway.NatGatewayId",
            "Type": "String"
          }
        ],
        "nextStep": "verifyNatGwAvailable"
      },
      {
        "name": "verifyNatGwAvailable",
        "action": "aws:waitForAwsResourceProperty",
        "timeoutSeconds": 150,
        "inputs": {
          "Service": "ec2",
          "Api": "DescribeNatGateways",
          "NatGatewayIds": [
            "{{ createNatGw.natGwId }}"
          ],
          "PropertySelector": "$.NatGateways[0].State",
          "DesiredValues": [
            "available"
          ]
        },
        "nextStep": "createNatRoute"
      },
      {
        "name": "createNatRoute",
        "action": "aws:executeAwsApi",
        "onFailure": "Abort",
        "inputs": {
          "Service": "ec2",
          "Api": "CreateRoute",
          "DestinationCidrBlock": "0.0.0.0/0",
          "NatGatewayId": "{{ createNatGw.natGwId }}",
          "RouteTableId": "{{ getMainRtb.mainRtbId }}"
        },
        "nextStep": "createPubRoute"
      },
      {
        "name": "createPubRoute",
        "action": "aws:executeAwsApi",
        "onFailure": "Abort",
        "inputs": {
          "Service": "ec2",
          "Api": "CreateRoute",
```

```
            "DestinationCidrBlock": "0.0.0.0/0",
            "GatewayId": "{{ createIgw.igwId }}",
            "RouteTableId": "{{ createPubRtb.pubRtbId }}"
          },
          "nextStep": "setPubSubAssoc"
        },
        {
          "name": "setPubSubAssoc",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "AssociateRouteTable",
            "RouteTableId": "{{ createPubRtb.pubRtbId }}",
            "SubnetId": "{{ createPubSubnet.pubSubnetId }}"
          }
        },
        {
          "name": "createDhcpOptions",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "CreateDhcpOptions",
            "DhcpConfigurations": [
              {
                "Key": "domain-name-servers",
                "Values": ["10.0.100.50,10.0.101.50"]
              },
              {
                "Key": "domain-name",
                "Values": ["sample.com"]
              }
            ]
          },
          "outputs": [
            {
              "Name": "dhcpOptionsId",
              "Selector": "$.DhcpOptions.DhcpOptionsId",
              "Type": "String"
            }
          ],
          "nextStep": "createDCSubnet1"
        },
```

```
        {
          "name": "createDCSubnet1",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "CreateSubnet",
            "CidrBlock": "10.0.100.0/24",
            "AvailabilityZone": "us-west-2a",
            "VpcId": "{{ createVpc.vpcId }}"
          },
          "outputs": [
            {
              "Name": "firstSubnetId",
              "Selector": "$.Subnet.SubnetId",
              "Type": "String"
            }
          ],
          "nextStep": "createDCSubnet2"
        },
        {
          "name": "createDCSubnet2",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "CreateSubnet",
            "CidrBlock": "10.0.101.0/24",
            "AvailabilityZone": "us-west-2b",
            "VpcId": "{{ createVpc.vpcId }}"
          },
          "outputs": [
            {
              "Name": "secondSubnetId",
              "Selector": "$.Subnet.SubnetId",
              "Type": "String"
            }
          ],
          "nextStep": "createDCSecGroup"
        },
        {
          "name": "createDCSecGroup",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
```

```
      "inputs": {
        "Service": "ec2",
        "Api": "CreateSecurityGroup",
        "GroupName": "SampleDCSecGroup",
        "Description": "Security Group for Example Domain Controllers",
        "VpcId": "{{ createVpc.vpcId }}"
      },
      "outputs": [
        {
          "Name": "dcSecGroupId",
          "Selector": "$.GroupId",
          "Type": "String"
        }
      ],
      "nextStep": "authIngressDCTraffic"
    },
    {
      "name": "authIngressDCTraffic",
      "action": "aws:executeAwsApi",
      "onFailure": "Abort",
      "inputs": {
        "Service": "ec2",
        "Api": "AuthorizeSecurityGroupIngress",
        "GroupId": "{{ createDCSecGroup.dcSecGroupId }}",
        "IpPermissions": [
          {
            "FromPort": -1,
            "IpProtocol": "-1",
            "IpRanges": [
              {
                "CidrIp": "0.0.0.0/0",
                "Description": "Allow all traffic between Domain Controllers"
              }
            ]
          }
        ]
      },
      "nextStep": "verifyInstanceProfile"
    },
    {
      "name": "verifyInstanceProfile",
      "action": "aws:waitForAwsResourceProperty",
      "maxAttempts": 5,
      "onFailure": "Abort",
```

```
          "inputs": {
            "Service": "iam",
            "Api": "ListInstanceProfilesForRole",
            "RoleName": "sampleSSMInstanceRole",
            "PropertySelector": "$.InstanceProfiles[0].Arn",
            "DesiredValues": [
              "{{ createSSMInstanceProfile.instanceProfileArn }}"
            ]
          },
          "nextStep": "iamEventualConsistency"
        },
        {
          "name": "iamEventualConsistency",
          "action": "aws:sleep",
          "inputs": {
            "Duration": "PT2M"
          },
          "nextStep": "launchDC1"
        },
        {
          "name": "launchDC1",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "RunInstances",
            "BlockDeviceMappings": [
              {
                "DeviceName": "/dev/sda1",
                "Ebs": {
                  "DeleteOnTermination": true,
                  "VolumeSize": 50,
                  "VolumeType": "gp2"
                }
              },
              {
                "DeviceName": "xvdf",
                "Ebs": {
                  "DeleteOnTermination": true,
                  "VolumeSize": 100,
                  "VolumeType": "gp2"
                }
              }
            ],
```

```
            "IamInstanceProfile": {
              "Arn": "{{ createSSMInstanceProfile.instanceProfileArn }}"
            },
            "ImageId": "{{ getLatestWindowsAmi.amiId }}",
            "InstanceType": "t2.micro",
            "MaxCount": 1,
            "MinCount": 1,
            "PrivateIpAddress": "10.0.100.50",
            "SecurityGroupIds": [
              "{{ createDCSecGroup.dcSecGroupId }}"
            ],
            "SubnetId": "{{ createDCSubnet1.firstSubnetId }}",
            "TagSpecifications": [
              {
                "ResourceType": "instance",
                "Tags": [
                  {
                    "Key": "Name",
                    "Value": "SampleDC1"
                  }
                ]
              }
            ]
          },
          "outputs": [
            {
              "Name": "pdcInstanceId",
              "Selector": "$.Instances[0].InstanceId",
              "Type": "String"
            }
          ],
          "nextStep": "launchDC2"
        },
        {
          "name": "launchDC2",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "RunInstances",
            "BlockDeviceMappings": [
              {
                "DeviceName": "/dev/sda1",
                "Ebs": {
```

```
                "DeleteOnTermination": true,
                "VolumeSize": 50,
                "VolumeType": "gp2"
              }
            },
            {
              "DeviceName": "xvdf",
              "Ebs": {
                "DeleteOnTermination": true,
                "VolumeSize": 100,
                "VolumeType": "gp2"
              }
            }
          ],
          "IamInstanceProfile": {
            "Arn": "{{ createSSMInstanceProfile.instanceProfileArn }}"
          },
          "ImageId": "{{ getLatestWindowsAmi.amiId }}",
          "InstanceType": "t2.micro",
          "MaxCount": 1,
          "MinCount": 1,
          "PrivateIpAddress": "10.0.101.50",
          "SecurityGroupIds": [
            "{{ createDCSecGroup.dcSecGroupId }}"
          ],
          "SubnetId": "{{ createDCSubnet2.secondSubnetId }}",
          "TagSpecifications": [
            {
              "ResourceType": "instance",
              "Tags": [
                {
                  "Key": "Name",
                  "Value": "SampleDC2"
                }
              ]
            }
          ]
        },
        "outputs": [
          {
            "Name": "adcInstanceId",
            "Selector": "$.Instances[0].InstanceId",
            "Type": "String"
          }
```

```
      ],
      "nextStep": "verifyDCInstanceState"
    },
    {
      "name": "verifyDCInstanceState",
      "action": "aws:waitForAwsResourceProperty",
      "inputs": {
        "Service": "ec2",
        "Api": "DescribeInstanceStatus",
        "IncludeAllInstances": true,
        "InstanceIds": [
          "{{ launchDC1.pdcInstanceId }}",
          "{{ launchDC2.adcInstanceId }}"
        ],
        "PropertySelector": "$.InstanceStatuses[0].InstanceState.Name",
        "DesiredValues": [
          "running"
        ]
      },
      "nextStep": "verifyInstancesOnlineSSM"
    },
    {
      "name": "verifyInstancesOnlineSSM",
      "action": "aws:waitForAwsResourceProperty",
      "timeoutSeconds": 600,
      "inputs": {
        "Service": "ssm",
        "Api": "DescribeInstanceInformation",
        "InstanceInformationFilterList": [
          {
            "key": "InstanceIds",
            "valueSet": [
              "{{ launchDC1.pdcInstanceId }}",
              "{{ launchDC2.adcInstanceId }}"
            ]
          }
        ],
        "PropertySelector": "$.InstanceInformationList[0].PingStatus",
        "DesiredValues": [
          "Online"
        ]
      },
      "nextStep": "installADRoles"
    },
```

```
        {
          "name": "installADRoles",
          "action": "aws:runCommand",
          "inputs": {
            "DocumentName": "AWS-RunPowerShellScript",
            "InstanceIds": [
              "{{ launchDC1.pdcInstanceId }}",
              "{{ launchDC2.adcInstanceId }}"
            ],
            "Parameters": {
              "commands": [
                "try {",
                "  Install-WindowsFeature -Name AD-Domain-Services -
IncludeManagementTools",
                "}",
                "catch {",
                "  Write-Error \"Failed to install ADDS Role.\"",
                "}"
              ]
            }
          },
          "nextStep": "setAdminPassword"
        },
        {
          "name": "setAdminPassword",
          "action": "aws:runCommand",
          "inputs": {
            "DocumentName": "AWS-RunPowerShellScript",
            "InstanceIds": [
              "{{ launchDC1.pdcInstanceId }}"
            ],
            "Parameters": {
              "commands": [
                "net user Administrator \"sampleAdminPass123!\""
              ]
            }
          },
          "nextStep": "createForest"
        },
        {
          "name": "createForest",
          "action": "aws:runCommand",
          "inputs": {
            "DocumentName": "AWS-RunPowerShellScript",
```

```
            "InstanceIds": [
              "{{ launchDC1.pdcInstanceId }}"
            ],
            "Parameters": {
              "commands": [
                "$dsrmPass = 'sample123!' | ConvertTo-SecureString -asPlainText -
Force",
                "try {",
                "   Install-ADDSForest -DomainName \"sample.com\" -DomainMode 6 -
ForestMode 6 -InstallDNS -DatabasePath \"D:\\NTDS\" -SysvolPath \"D:\\SYSVOL\" -
SafeModeAdministratorPassword $dsrmPass -Force",
                "}",
                "catch {",
                "   Write-Error $_",
                "}",
                "try {",
                "   Add-DnsServerForwarder -IPAddress \"10.0.100.2\"",
                "}",
                "catch {",
                "   Write-Error $_",
                "}"
              ]
            }
          },
          "nextStep": "associateDhcpOptions"
        },
        {
          "name": "associateDhcpOptions",
          "action": "aws:executeAwsApi",
          "onFailure": "Abort",
          "inputs": {
            "Service": "ec2",
            "Api": "AssociateDhcpOptions",
            "DhcpOptionsId": "{{ createDhcpOptions.dhcpOptionsId }}",
            "VpcId": "{{ createVpc.vpcId }}"
          },
          "nextStep": "waitForADServices"
        },
        {
          "name": "waitForADServices",
          "action": "aws:sleep",
          "inputs": {
            "Duration": "PT1M"
          },
```

```
          "nextStep": "promoteADC"
        },
        {
          "name": "promoteADC",
          "action": "aws:runCommand",
          "inputs": {
            "DocumentName": "AWS-RunPowerShellScript",
            "InstanceIds": [
              "{{ launchDC2.adcInstanceId }}"
            ],
            "Parameters": {
              "commands": [
                "ipconfig /renew",
                "$dsrmPass = 'sample123!' | ConvertTo-SecureString -asPlainText -
Force",
                "$domAdminUser = \"sample\\Administrator\"",
                "$domAdminPass = \"sampleAdminPass123!\" | ConvertTo-SecureString -
asPlainText -Force",
                "$domAdminCred = New-Object
 System.Management.Automation.PSCredential($domAdminUser,$domAdminPass)",
                "try {",
                "   Install-ADDSDomainController -DomainName \"sample.com
\" -InstallDNS -DatabasePath \"D:\\NTDS\" -SysvolPath \"D:\\SYSVOL\" -
SafeModeAdministratorPassword $dsrmPass -Credential $domAdminCred -Force",
                "}",
                "catch {",
                "   Write-Error $_",
                "}"
              ]
            }
          }
        }
      ]
    }
```

Wiederherstellen eines Root-Volumes aus dem letzten Snapshot

Das Betriebssystem auf einem Root-Volume kann aus verschiedenen Gründen beschädigt werden. Beispielsweise können Instances nach einem Patchvorgang aufgrund eines beschädigten Kernels oder einer beschädigten Registrierung nicht mehr erfolgreich gestartet werden. Durch die Automatisierung gängiger Problembehandlungsaufgaben, wie z. B. das Wiederherstellen eines Root-Volumes aus dem letzten Snapshot, der vor dem Patch-Vorgang erstellt wurde, können Ausfallzeiten

reduziert und Ihre Problembehebungsmaßnahmen beschleunigt werden. AWS Systems Manager Automatisierungsmaßnahmen können Ihnen dabei helfen, dies zu erreichen. Automation ist ein Tool in AWS Systems Manager.

Das folgende AWS Systems Manager Beispiel-Runbook führt diese Aktionen aus:

- Verwendet die `aws:executeAwsApi` Automatisierungsaktion zum Abrufen von Details aus dem Root-Volumen der Instance.
- Verwendet die `aws:executeScript` Automatisierungsaktion zum Abrufen des neuesten Snapshots für das Root-Volume.
- Verwendet die `aws:branch` Automatisierungsaktion, um die Automatisierung fortzusetzen, wenn ein Snapshot für das Root-Volume gefunden wird.

YAML

```
   ---
   description: Custom Automation Troubleshooting Example
   schemaVersion: '0.3'
   assumeRole: "{{ AutomationAssumeRole }}"
   parameters:
     AutomationAssumeRole:
       type: String
       description: "(Required) The ARN of the role that allows Automation to
perform
         the actions on your behalf. If no role is specified, Systems Manager
Automation
         uses your IAM permissions to use this runbook."
       default: ''
     InstanceId:
         type: String
         description: "(Required) The Instance Id whose root EBS volume you want to
restore the latest Snapshot."
         default: ''
   mainSteps:
   - name: getInstanceDetails
     action: aws:executeAwsApi
     onFailure: Abort
     inputs:
       Service: ec2
       Api: DescribeInstances
```

```
        InstanceIds:
        - "{{ InstanceId }}"
      outputs:
        - Name: availabilityZone
          Selector: "$.Reservations[0].Instances[0].Placement.AvailabilityZone"
          Type: String
        - Name: rootDeviceName
          Selector: "$.Reservations[0].Instances[0].RootDeviceName"
          Type: String
      nextStep: getRootVolumeId
    - name: getRootVolumeId
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: DescribeVolumes
        Filters:
        -  Name: attachment.device
           Values: ["{{ getInstanceDetails.rootDeviceName }}"]
        -  Name: attachment.instance-id
           Values: ["{{ InstanceId }}"]
      outputs:
        - Name: rootVolumeId
          Selector: "$.Volumes[0].VolumeId"
          Type: String
      nextStep: getSnapshotsByStartTime
    - name: getSnapshotsByStartTime
      action: aws:executeScript
      timeoutSeconds: 45
      onFailure: Abort
      inputs:
        Runtime: python3.11
        Handler: getSnapshotsByStartTime
        InputPayload:
          rootVolumeId : "{{ getRootVolumeId.rootVolumeId }}"
        Script: |-
          def getSnapshotsByStartTime(events,context):
            import boto3

            #Initialize client
            ec2 = boto3.client('ec2')
            rootVolumeId = events['rootVolumeId']
            snapshotsQuery = ec2.describe_snapshots(
              Filters=[
```

```
                {
                  "Name": "volume-id",
                  "Values": [rootVolumeId]
                }
              ]
            )
            if not snapshotsQuery['Snapshots']:
              noSnapshotFoundString = "NoSnapshotFound"
              return { 'noSnapshotFound' : noSnapshotFoundString }
            else:
              jsonSnapshots = snapshotsQuery['Snapshots']
              sortedSnapshots = sorted(jsonSnapshots, key=lambda k: k['StartTime'],
reverse=True)
              latestSortedSnapshotId = sortedSnapshots[0]['SnapshotId']
              return { 'latestSnapshotId' : latestSortedSnapshotId }
      outputs:
      - Name: Payload
        Selector: $.Payload
        Type: StringMap
      - Name: latestSnapshotId
        Selector: $.Payload.latestSnapshotId
        Type: String
      - Name: noSnapshotFound
        Selector: $.Payload.noSnapshotFound
        Type: String
      nextStep: branchFromResults
    - name: branchFromResults
      action: aws:branch
      onFailure: Abort
      inputs:
        Choices:
        - NextStep: createNewRootVolumeFromSnapshot
          Not:
            Variable: "{{ getSnapshotsByStartTime.noSnapshotFound }}"
            StringEquals: "NoSnapshotFound"
      isEnd: true
    - name: createNewRootVolumeFromSnapshot
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: CreateVolume
        AvailabilityZone: "{{ getInstanceDetails.availabilityZone }}"
        SnapshotId: "{{ getSnapshotsByStartTime.latestSnapshotId }}"
```

```
      outputs:
        - Name: newRootVolumeId
          Selector: "$.VolumeId"
          Type: String
      nextStep: stopInstance
    - name: stopInstance
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: StopInstances
        InstanceIds:
        - "{{ InstanceId }}"
      nextStep: verifyVolumeAvailability
    - name: verifyVolumeAvailability
      action: aws:waitForAwsResourceProperty
      timeoutSeconds: 120
      inputs:
        Service: ec2
        Api: DescribeVolumes
        VolumeIds:
        - "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
        PropertySelector: "$.Volumes[0].State"
        DesiredValues:
        - "available"
      nextStep: verifyInstanceStopped
    - name: verifyInstanceStopped
      action: aws:waitForAwsResourceProperty
      timeoutSeconds: 120
      inputs:
        Service: ec2
        Api: DescribeInstances
        InstanceIds:
        - "{{ InstanceId }}"
        PropertySelector: "$.Reservations[0].Instances[0].State.Name"
        DesiredValues:
        - "stopped"
      nextStep: detachRootVolume
    - name: detachRootVolume
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: DetachVolume
```

```
        VolumeId: "{{ getRootVolumeId.rootVolumeId }}"
      nextStep: verifyRootVolumeDetached
    - name: verifyRootVolumeDetached
      action: aws:waitForAwsResourceProperty
      timeoutSeconds: 30
      inputs:
        Service: ec2
        Api: DescribeVolumes
        VolumeIds:
        - "{{ getRootVolumeId.rootVolumeId }}"
        PropertySelector: "$.Volumes[0].State"
        DesiredValues:
        - "available"
      nextStep: attachNewRootVolume
    - name: attachNewRootVolume
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: AttachVolume
        Device: "{{ getInstanceDetails.rootDeviceName }}"
        InstanceId: "{{ InstanceId }}"
        VolumeId: "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
      nextStep: verifyNewRootVolumeAttached
    - name: verifyNewRootVolumeAttached
      action: aws:waitForAwsResourceProperty
      timeoutSeconds: 30
      inputs:
        Service: ec2
        Api: DescribeVolumes
        VolumeIds:
        - "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
        PropertySelector: "$.Volumes[0].Attachments[0].State"
        DesiredValues:
        - "attached"
      nextStep: startInstance
    - name: startInstance
      action: aws:executeAwsApi
      onFailure: Abort
      inputs:
        Service: ec2
        Api: StartInstances
        InstanceIds:
```

```
        - "{{ InstanceId }}"
```

JSON

```
   {
      "description": "Custom Automation Troubleshooting Example",
      "schemaVersion": "0.3",
      "assumeRole": "{{ AutomationAssumeRole }}",
      "parameters": {
         "AutomationAssumeRole": {
            "type": "String",
            "description": "(Required) The ARN of the role that allows Automation
to perform the actions on your behalf. If no role is specified, Systems Manager
Automation uses your IAM permissions to run this runbook.",
            "default": ""
         },
         "InstanceId": {
            "type": "String",
            "description": "(Required) The Instance Id whose root EBS volume you
want to restore the latest Snapshot.",
            "default": ""
         }
      },
      "mainSteps": [
         {
            "name": "getInstanceDetails",
            "action": "aws:executeAwsApi",
            "onFailure": "Abort",
            "inputs": {
               "Service": "ec2",
               "Api": "DescribeInstances",
               "InstanceIds": [
                  "{{ InstanceId }}"
               ]
            },
            "outputs": [
               {
                  "Name": "availabilityZone",
                  "Selector":
"$.Reservations[0].Instances[0].Placement.AvailabilityZone",
                  "Type": "String"
               },
```

```
            {
               "Name": "rootDeviceName",
               "Selector": "$.Reservations[0].Instances[0].RootDeviceName",
               "Type": "String"
            }
         ],
         "nextStep": "getRootVolumeId"
      },
      {
         "name": "getRootVolumeId",
         "action": "aws:executeAwsApi",
         "onFailure": "Abort",
         "inputs": {
            "Service": "ec2",
            "Api": "DescribeVolumes",
            "Filters": [
               {
                  "Name": "attachment.device",
                  "Values": [
                     "{{ getInstanceDetails.rootDeviceName }}"
                  ]
               },
               {
                  "Name": "attachment.instance-id",
                  "Values": [
                     "{{ InstanceId }}"
                  ]
               }
            ]
         },
         "outputs": [
            {
               "Name": "rootVolumeId",
               "Selector": "$.Volumes[0].VolumeId",
               "Type": "String"
            }
         ],
         "nextStep": "getSnapshotsByStartTime"
      },
      {
         "name": "getSnapshotsByStartTime",
         "action": "aws:executeScript",
         "timeoutSeconds": 45,
         "onFailure": "Continue",
```

```
            "inputs": {
               "Runtime": "python3.11",
               "Handler": "getSnapshotsByStartTime",
               "InputPayload": {
                  "rootVolumeId": "{{ getRootVolumeId.rootVolumeId }}"
               },
               "Attachment": "getSnapshotsByStartTime.py"
            },
            "outputs": [
               {
                  "Name": "Payload",
                  "Selector": "$.Payload",
                  "Type": "StringMap"
               },
               {
                  "Name": "latestSnapshotId",
                  "Selector": "$.Payload.latestSnapshotId",
                  "Type": "String"
               },
               {
                  "Name": "noSnapshotFound",
                  "Selector": "$.Payload.noSnapshotFound",
                  "Type": "String"
               }
            ],
            "nextStep": "branchFromResults"
         },
         {
            "name": "branchFromResults",
            "action": "aws:branch",
            "onFailure": "Abort",
            "inputs": {
               "Choices": [
                  {
                     "NextStep": "createNewRootVolumeFromSnapshot",
                     "Not": {
                        "Variable":
"{{ getSnapshotsByStartTime.noSnapshotFound }}",
                        "StringEquals": "NoSnapshotFound"
                     }
                  }
               ]
            },
            "isEnd": true
```

```
        },
        {
           "name": "createNewRootVolumeFromSnapshot",
           "action": "aws:executeAwsApi",
           "onFailure": "Abort",
           "inputs": {
              "Service": "ec2",
              "Api": "CreateVolume",
              "AvailabilityZone": "{{ getInstanceDetails.availabilityZone }}",
              "SnapshotId": "{{ getSnapshotsByStartTime.latestSnapshotId }}"
           },
           "outputs": [
              {
                 "Name": "newRootVolumeId",
                 "Selector": "$.VolumeId",
                 "Type": "String"
              }
           ],
           "nextStep": "stopInstance"
        },
        {
           "name": "stopInstance",
           "action": "aws:executeAwsApi",
           "onFailure": "Abort",
           "inputs": {
              "Service": "ec2",
              "Api": "StopInstances",
              "InstanceIds": [
                 "{{ InstanceId }}"
              ]
           },
           "nextStep": "verifyVolumeAvailability"
        },
        {
           "name": "verifyVolumeAvailability",
           "action": "aws:waitForAwsResourceProperty",
           "timeoutSeconds": 120,
           "inputs": {
              "Service": "ec2",
              "Api": "DescribeVolumes",
              "VolumeIds": [
                 "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
              ],
              "PropertySelector": "$.Volumes[0].State",
```

```
                "DesiredValues": [
                   "available"
                ]
             },
             "nextStep": "verifyInstanceStopped"
          },
          {
             "name": "verifyInstanceStopped",
             "action": "aws:waitForAwsResourceProperty",
             "timeoutSeconds": 120,
             "inputs": {
                "Service": "ec2",
                "Api": "DescribeInstances",
                "InstanceIds": [
                   "{{ InstanceId }}"
                ],
                "PropertySelector": "$.Reservations[0].Instances[0].State.Name",
                "DesiredValues": [
                   "stopped"
                ]
             },
             "nextStep": "detachRootVolume"
          },
          {
             "name": "detachRootVolume",
             "action": "aws:executeAwsApi",
             "onFailure": "Abort",
             "inputs": {
                "Service": "ec2",
                "Api": "DetachVolume",
                "VolumeId": "{{ getRootVolumeId.rootVolumeId }}"
             },
             "nextStep": "verifyRootVolumeDetached"
          },
          {
             "name": "verifyRootVolumeDetached",
             "action": "aws:waitForAwsResourceProperty",
             "timeoutSeconds": 30,
             "inputs": {
                "Service": "ec2",
                "Api": "DescribeVolumes",
                "VolumeIds": [
                   "{{ getRootVolumeId.rootVolumeId }}"
                ],
```

```
                "PropertySelector": "$.Volumes[0].State",
                "DesiredValues": [
                   "available"
                ]
             },
             "nextStep": "attachNewRootVolume"
          },
          {
             "name": "attachNewRootVolume",
             "action": "aws:executeAwsApi",
             "onFailure": "Abort",
             "inputs": {
                "Service": "ec2",
                "Api": "AttachVolume",
                "Device": "{{ getInstanceDetails.rootDeviceName }}",
                "InstanceId": "{{ InstanceId }}",
                "VolumeId": "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
             },
             "nextStep": "verifyNewRootVolumeAttached"
          },
          {
             "name": "verifyNewRootVolumeAttached",
             "action": "aws:waitForAwsResourceProperty",
             "timeoutSeconds": 30,
             "inputs": {
                "Service": "ec2",
                "Api": "DescribeVolumes",
                "VolumeIds": [
                   "{{ createNewRootVolumeFromSnapshot.newRootVolumeId }}"
                ],
                "PropertySelector": "$.Volumes[0].Attachments[0].State",
                "DesiredValues": [
                   "attached"
                ]
             },
             "nextStep": "startInstance"
          },
          {
             "name": "startInstance",
             "action": "aws:executeAwsApi",
             "onFailure": "Abort",
             "inputs": {
                "Service": "ec2",
                "Api": "StartInstances",
```

```
                "InstanceIds": [
                   "{{ InstanceId }}"
                ]
             }
          }
       ],
       "files": {
            "getSnapshotsByStartTime.py": {
                "checksums": {
                    "sha256": "sampleETagValue"
                }
            }
        }
    }
```

Erstellen eines AMI und einer regionenübergreifenden Kopie

Das Erstellen eines Amazon Machine Image (AMI) einer Instance ist ein üblicher Prozess, der bei Backup und Wiedherherstellung verwendet wird. Sie können sich auch dafür entscheiden, ein AMI als Teil einer Notfallwiederherstellungsarchitektur in eine andere AWS-Region zu kopieren. Durch die Automatisierung gängiger Wartungsaufgaben kann die Ausfallzeit reduziert werden, wenn ein Problem ein Failover erfordert. AWS Systems Manager Automatisierungsaktionen können Ihnen dabei helfen. Automation ist ein Tool in AWS Systems Manager.

Das folgende AWS Systems Manager Beispiel-Runbook führt diese Aktionen aus:

- Verwendet die `aws:executeAwsApi` Automatisierungsaktion zur Erstellung eines AMI.
- Verwendet die `aws:waitForAwsResourceProperty` Automatisierungsaktion zur Bestätigung der Verfügbarkeit des AMI.
- Verwendet die `aws:executeScript` Automatisierungsaktion zum Kopieren des AMI in die Zielregion.

YAML

```
---
description: Custom Automation Backup and Recovery Example
schemaVersion: '0.3'
assumeRole: "{{ AutomationAssumeRole }}"
parameters:
```

```
    AutomationAssumeRole:
      type: String
      description: "(Required) The ARN of the role that allows Automation to
perform
        the actions on your behalf. If no role is specified, Systems Manager
Automation
        uses your IAM permissions to use this runbook."
      default: ''
    InstanceId:
      type: String
      description: "(Required) The ID of the EC2 instance."
      default: ''
  mainSteps:
  - name: createImage
    action: aws:executeAwsApi
    onFailure: Abort
    inputs:
      Service: ec2
      Api: CreateImage
      InstanceId: "{{ InstanceId }}"
      Name: "Automation Image for {{ InstanceId }}"
      NoReboot: false
    outputs:
      - Name: newImageId
        Selector: "$.ImageId"
        Type: String
    nextStep: verifyImageAvailability
  - name: verifyImageAvailability
    action: aws:waitForAwsResourceProperty
    timeoutSeconds: 600
    inputs:
      Service: ec2
      Api: DescribeImages
      ImageIds:
      - "{{ createImage.newImageId }}"
      PropertySelector: "$.Images[0].State"
      DesiredValues:
      - available
    nextStep: copyImage
  - name: copyImage
    action: aws:executeScript
    timeoutSeconds: 45
    onFailure: Abort
    inputs:
```

```
      Runtime: python3.11
      Handler: crossRegionImageCopy
      InputPayload:
        newImageId : "{{ createImage.newImageId }}"
      Script: |-
        def crossRegionImageCopy(events,context):
          import boto3

          #Initialize client
          ec2 = boto3.client('ec2', region_name='us-east-1')
          newImageId = events['newImageId']

          ec2.copy_image(
            Name='DR Copy for ' + newImageId,
            SourceImageId=newImageId,
            SourceRegion='us-west-2'
          )
```

JSON

```
   {
      "description": "Custom Automation Backup and Recovery Example",
      "schemaVersion": "0.3",
      "assumeRole": "{{ AutomationAssumeRole }}",
      "parameters": {
         "AutomationAssumeRole": {
            "type": "String",
            "description": "(Required) The ARN of the role that allows Automation
to perform\nthe actions on your behalf. If no role is specified, Systems Manager
Automation\nuses your IAM permissions to run this runbook.",
            "default": ""
         },
         "InstanceId": {
            "type": "String",
            "description": "(Required) The ID of the EC2 instance.",
            "default": ""
         }
      },
      "mainSteps": [
         {
            "name": "createImage",
            "action": "aws:executeAwsApi",
```

```
            "onFailure": "Abort",
            "inputs": {
               "Service": "ec2",
               "Api": "CreateImage",
               "InstanceId": "{{ InstanceId }}",
               "Name": "Automation Image for {{ InstanceId }}",
               "NoReboot": false
            },
            "outputs": [
               {
                  "Name": "newImageId",
                  "Selector": "$.ImageId",
                  "Type": "String"
               }
            ],
            "nextStep": "verifyImageAvailability"
         },
         {
            "name": "verifyImageAvailability",
            "action": "aws:waitForAwsResourceProperty",
            "timeoutSeconds": 600,
            "inputs": {
               "Service": "ec2",
               "Api": "DescribeImages",
               "ImageIds": [
                  "{{ createImage.newImageId }}"
               ],
               "PropertySelector": "$.Images[0].State",
               "DesiredValues": [
                  "available"
               ]
            },
            "nextStep": "copyImage"
         },
         {
            "name": "copyImage",
            "action": "aws:executeScript",
            "timeoutSeconds": 45,
            "onFailure": "Abort",
            "inputs": {
               "Runtime": "python3.11",
               "Handler": "crossRegionImageCopy",
               "InputPayload": {
                  "newImageId": "{{ createImage.newImageId }}"
```

```
                },
                "Attachment": "crossRegionImageCopy.py"
            }
        }
    ],
    "files": {
         "crossRegionImageCopy.py": {
            "checksums": {
                "sha256": "sampleETagValue"
            }
        }
    }
}
```

Erstellen von Eingabeparametern, die AWS-Ressourcen ausfüllen

Automation, ein Tool im Systems Manager, füllt AWS-Ressourcen in der AWS-Managementkonsole aus, die mit dem Ressourcentyp übereinstimmen, den Sie für einen Eingabeparameter definieren. Ressourcen in Ihrem AWS-Konto, die mit dem Ressourcentyp übereinstimmen, werden in einer Dropdown-Liste angezeigt, die Sie auswählen können. Sie können Eingabeparameter-Typen für Amazon Elastic Compute Cloud (Amazon EC2)-Instances, Amazon Simple Storage Service (Amazon S3)-Buckets und AWS Identity and Access Management (IAM)-Rollen definieren. Die unterstützten Typdefinitionen und die regulären Ausdrücke, die zum Suchen übereinstimmender Ressourcen verwendet werden, lauten wie folgt:

- `AWS::EC2::Instance::Id` - `^m?i-([a-z0-9]{8}|[a-z0-9]{17})$`
- `List<AWS::EC2::Instance::Id>` - `^m?i-([a-z0-9]{8}|[a-z0-9]{17})$`
- `AWS::S3::Bucket::Name` - `^[0-9a-z][a-z0-9\\-\\.]{3,63}$`
- `List<AWS::S3::Bucket::Name>` - `^[0-9a-z][a-z0-9\\-\\.]{3,63}$`
- `AWS::IAM::Role::Arn` - `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):iam::[0-9]{12}:role/.*$`
- `List<AWS::IAM::Role::Arn>` - `^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):iam::[0-9]{12}:role/.*$`

Es folgt ein Beispiel für Eingabeparameter-Typen, die im Runbook-Inhalt definiert sind.

YAML

```
description: Enables encryption on an Amazon S3 bucket
schemaVersion: '0.3'
assumeRole: '{{ AutomationAssumeRole }}'
parameters:
  BucketName:
    type: 'AWS::S3::Bucket::Name'
    description: (Required) The name of the Amazon S3 bucket you want to encrypt.
  SSEAlgorithm:
    type: String
    description: (Optional) The server-side encryption algorithm to use for the
 default encryption.
    default: AES256
  AutomationAssumeRole:
    type: 'AWS::IAM::Role::Arn'
    description: (Optional) The Amazon Resource Name (ARN) of the role that allows
 Automation to perform the actions on your behalf.
    default: ''
mainSteps:
  - name: enableBucketEncryption
    action: 'aws:executeAwsApi'
    inputs:
      Service: s3
      Api: PutBucketEncryption
      Bucket: '{{BucketName}}'
      ServerSideEncryptionConfiguration:
        Rules:
          - ApplyServerSideEncryptionByDefault:
              SSEAlgorithm: '{{SSEAlgorithm}}'
    isEnd: true
```

JSON

```
{
   "description": "Enables encryption on an Amazon S3 bucket",
   "schemaVersion": "0.3",
   "assumeRole": "{{ AutomationAssumeRole }}",
   "parameters": {
      "BucketName": {
         "type": "AWS::S3::Bucket::Name",
         "description": "(Required) The name of the Amazon S3 bucket you want to
 encrypt."
```

```
      },
      "SSEAlgorithm": {
         "type": "String",
         "description": "(Optional) The server-side encryption algorithm to use for
 the default encryption.",
         "default": "AES256"
      },
      "AutomationAssumeRole": {
         "type": "AWS::IAM::Role::Arn",
         "description": "(Optional) The Amazon Resource Name (ARN) of the role that
 allows Automation to perform the actions on your behalf.",
         "default": ""
      }
   },
   "mainSteps": [
      {
         "name": "enableBucketEncryption",
         "action": "aws:executeAwsApi",
         "inputs": {
            "Service": "s3",
            "Api": "PutBucketEncryption",
            "Bucket": "{{BucketName}}",
            "ServerSideEncryptionConfiguration": {
               "Rules": [
                  {
                     "ApplyServerSideEncryptionByDefault": {
                        "SSEAlgorithm": "{{SSEAlgorithm}}"
                     }
                  }
               ]
            }
         },
         "isEnd": true
      }
   ]
}
```

Verwenden von Document Builder zur Erstellung von Runbooks

Wenn die AWS Systems Manager öffentlichen Runbooks nicht alle Aktionen unterstützen, die Sie mit Ihren AWS Ressourcen ausführen möchten, können Sie Ihre eigenen Runbooks erstellen. Um ein benutzerdefiniertes Runbook zu erstellen, können Sie manuell eine lokale Datei im YAML-

oder JSON-Format mit den entsprechenden Automatisierungsaktionen erstellen. Alternativ können Sie Document Builder in der Systems-Manager-Automation-Konsole verwenden, um ein benutzerdefiniertes Runbook zu erstellen.

Mit Document Builder können Sie Ihrem benutzerdefinierten Runbook Automatisierungsaktionen hinzufügen und die erforderlichen Parameter bereitstellen, ohne die JSON- oder YAML-Syntax verwenden zu müssen. Nachdem Sie Schritte hinzugefügt und das Runbook erstellt haben, konvertiert das System die von Ihnen hinzugefügten Aktionen in das YAML-Format, das von Systems Manager zum Ausführen von Automation verwendet werden kann.

Runbooks unterstützen die Verwendung von Markdown, einer Markup-Sprache, mit der Sie Wiki-Beschreibungen zu Runbooks und einzelnen Schritten innerhalb des Runbooks hinzufügen können. Weitere Informationen zur Verwendung von Markdown finden Sie unter Verwenden von Markdown in AWS.

Erstellen eines Runbooks mithilfe von Document Builder

Bevor Sie beginnen

Wir empfehlen Ihnen, sich über die verschiedenen Aktionen zu informieren, die Sie in einem Runbook verwenden können. Weitere Informationen finden Sie unter Systems Manager Automation Aktionen-Referenz.

So erstellen Sie ein Runbook mit Document Builder

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie Create automation (Automation erstellen).
4. Geben Sie unter Name einen aussagekräftigen Namen für das Runbook ein.
5. Geben Sie für Document description (Dokumentbeschreibung) die Beschreibung des Markdown-Stils für das Runbook an. Sie können Anweisungen für die Verwendung des Runbooks, nummerierte Schritte oder jede andere Art von Informationen zur Beschreibung des Runbooks bereitstellen. Informationen zum Formatieren von Inhalten finden Sie im Standardtext.

> **Tip**
>
> Wechseln Sie zwischen Hide preview (Vorschau ausblenden) und Show preview (Vorschau anzeigen), um zu sehen, wie der Beschreibungsinhalt während der Erstellung aussieht.

6. (Optional) Geben Sie unter Assume role (Rolle übernehmen) den Namen oder den ARN einer Servicerolle ein, um Aktionen in Ihrem Auftrag auszuführen. Wenn Sie keine Rolle angeben, verwendet Automation die Zugriffsberechtigungen des Benutzers, der die Automatisierung durchführt.

   > **Important**
   >
   > Für Runbooks, die sich nicht im Besitz von Amazon befinden und die die `aws:executeScript`-Aktion verwenden, muss eine Rolle angegeben werden. Weitere Informationen finden Sie unter Berechtigungen für die Verwendung von Runbooks.

7. (Optional) Geben Sie unter Outputs (Ausgänge) alle Ausgaben für die Automatisierung dieses Runbooks ein, um sie für andere Prozesse verfügbar zu machen.

   Wenn Ihr Runbook beispielsweise ein neues AMI erstellt, können Sie [“ CreateImage angeben. ImageId„] und verwenden Sie dann diese Ausgabe, um in einer nachfolgenden Automatisierung neue Instances zu erstellen.

8. (Optional) Erweitern Sie den Abschnitt Input parameters (Eingabeparameter) und führen Sie die folgenden Schritte aus.

   1. Geben Sie unter Parameter name (Parametername) einen beschreibenden Namen für den Runbookparameter ein, den Sie erstellen.
   2. Wählen Sie unter Type (Typ) einen Typ für den Parameter, z. B. `String` oder `MapList`.
   3. Führen Sie unter Required (Erforderlich) eine der folgenden Aktionen aus:
      - Wählen Sie Yes (Ja), wenn zur Laufzeit ein Wert für diesen Runbookparameter angegeben werden muss.
      - Wählen Sie No (Nein), wenn der Parameter nicht erforderlich ist, und geben Sie (optional) unter Default value (Standardwert) einen Standardparameterwert ein.
   4. Geben Sie unter Description (Beschreibung) eine Beschreibung für den Runbookparameter ein.

> **Note**
>
> Um weitere Runbookparameter hinzuzufügen, wählen Sie Add a parameter (Parameter hinzufügen). Um einen Runbookparameter zu entfernen, klicken Sie auf die Schaltfläche X (Entfernen).

9. (Optional) Erweitern Sie den Abschnitt Target type (Zieltyp) und wählen Sie einen Zieltyp, um die Arten der Ressourcen zu definieren, auf denen die Automatisierung ausgeführt werden kann. Um beispielsweise ein Runbook für EC2 Instanzen zu verwenden, wählen Sie`/AWS::EC2::Instance`.

   > **Note**
   >
   > Wenn Sie den Wert '/' angeben, kann das Runbook auf allen Arten von Ressourcen ausgeführt werden. Eine Liste gültiger Ressourcentypen finden Sie unter AWS - Ressourcentypen – Referenz im AWS CloudFormation Benutzerhandbuch.

10. (Optional) Erweitern Sie den Abschnitt Document tags (Dokument-Tags) und geben Sie ein oder mehrere Tag-Schlüssel-Wert-Paare ein, die auf das Runbook angewendet werden sollen. Tags erleichtern die Identifizierung, Organisation und Suche nach Ressourcen.
11. Geben Sie im Abschnitt Step 1 (Schritt 1) die folgenden Informationen an.

    - Geben Sie unter Step name (Schrittname) einen beschreibenden Namen für den ersten Schritt der Automatisierung ein.
    - Wählen Sie unter Action type (Aktionstyp) den Aktionstyp aus, der für diesen Schritt verwendet werden soll.

      Eine Liste und Informationen zu den verfügbaren Aktionstypen finden Sie unter Systems Manager Automation Aktionen-Referenz.
    - Geben Sie unter Description (Beschreibung) eine Beschreibung für den Automatisierungsschritt ein. Sie können Markdown verwenden, um Ihren Text zu formatieren.
    - Je nach ausgewähltem Action type (Aktionstyp) geben Sie im Abschnitt Step inputs (Schritteingaben) die erforderlichen Eingaben für den Aktionstyp ein. Wenn Sie beispielsweise die Aktion `aws:approve` ausgewählt haben, müssen Sie einen Wert für die `Approvers`-Eigenschaft angeben.

Informationen zu den Schritteingabefeldern finden Sie im Eintrag Systems Manager Automation Aktionen-Referenz für den ausgewählten Aktionstyp. Beispiel: `aws:executeStateMachine`— Führen Sie eine AWS Step Functions Zustandsmaschine aus.

- (Optional) Geben Sie für Additional inputs (Zusätzliche Eingaben) alle zusätzlichen Eingabewerte an, die für das Runbook erforderlich sind. Die verfügbaren Eingabetypen hängen vom Aktionstyp ab, den Sie für den Schritt ausgewählt haben. (Beachten Sie, dass einige Aktionstypen Eingabewerte erfordern.)

  > **Note**
  >
  > Um weitere Eingaben hinzuzufügen, wählen Sie Add optional input (Optionale Eingabe hinzufügen). Um eine Eingabe zu entfernen, wählen Sie die Schaltfläche X (Entfernen).

- (Optional) Geben Sie unter Outputs (Ausgänge) alle Ausgaben für diesen Schritts ein, um sie für andere Prozesse verfügbar zu machen.

  > **Note**
  >
  > Outputs (Ausgaben) sind nicht für alle Aktionstypen verfügbar.

- (Optional) Erweitern Sie den Abschnitt Common properties (Allgemeine Eigenschaften) und geben Sie Eigenschaften für die Aktionen an, die allen Automation-Aktionen gemeinsam sind. Beispielsweise können Sie für Timeout seconds (Timeout in Sekunden) anhand eines Werts in Sekunden angeben, wie lange der Schritt ausgeführt werden kann, bevor er beendet wird.

  Weitere Informationen finden Sie unter Von allen Aktionen gemeinsam genutzte Eigenschaften.

> **Note**
>
> Um weitere Schritte hinzuzufügen, wählen Sie Add step (Schritt hinzufügen) aus und wiederholen Sie das Verfahren zum Erstellen eines Schritts. Um einen Schritt zu entfernen, wählen Sie Remove step (Schritt entfernen).

12. Wählen Sie Create automation (Automation erstellen), um das Runbook zu speichern.

Erstellen eines Runbooks, das Skripte ausführt

Das folgende Verfahren zeigt, wie Sie mit Document Builder in der AWS Systems Manager - Automation-Konsole ein benutzerdefiniertes Runbook erstellen, das ein Skript ausführt.

Im ersten Schritt des von Ihnen erstellten Runbooks wird ein Skript ausgeführt, um eine Amazon Elastic Compute Cloud (Amazon EC2) -Instance zu starten. Im zweiten Schritt wird ein weiteres Skript ausgeführt, um zu überwachen, ob die Instance-Zustandsprüfung auf `ok` geändert werden soll. Anschließend wird für die Automatisierung ein Gesamtzustand von `Success` gemeldet.

Bevor Sie beginnen

Stellen Sie sicher, dass Sie die folgenden Schritte ausgeführt haben:

- Stellen Sie sicher, dass Sie über Administratorrechte verfügen oder dass Ihnen die entsprechenden Berechtigungen für den Zugriff auf Systems Manager in AWS Identity and Access Management (IAM) erteilt wurden.

  Weitere Informationen finden Sie unter Überprüfen des Benutzerzugriffs für Runbooks.
- Stellen Sie sicher, dass Sie in Ihrem AWS-Kontoüber eine IAM-Service-Rolle für Automation (auch als Rolle übernehmen bezeichnet) verfügen. Die Rolle ist erforderlich, da in dieser Anleitung die Aktion `aws:executeScript` verwendet wird.

  Weitere Informationen zum Erstellen dieser Rolle finden Sie unter Konfigurieren eines Service-Rollenzugriffs (Rolle übernehmen) für Automatisierungen.

  Hinweise zur IAM-Service-Rollenanforderung zum Ausführen von `aws:executeScript`, finden Sie unter Berechtigungen für die Verwendung von Runbooks.
- Stellen Sie sicher, dass Sie berechtigt sind, EC2 Instances zu starten.

  Weitere Informationen finden Sie unter IAM und Amazon EC2 im EC2 Amazon-Benutzerhandbuch.

So erstellen Sie ein benutzerdefiniertes Runbook, das Skripts mit Document Builder ausführt

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie Create automation (Automation erstellen).

4. Geben Sie unter Name diesen beschreibenden Namen für das Runbook ein: **`LaunchInstanceAndCheckStatus`**.

5. (Optional) Ersetzen Sie bei Document description (Dokumentbeschreibung) den Standardtext durch eine Beschreibung für dieses Runbooks, indem Sie Markdown verwenden. Im Folgenden wird ein -Beispiel gezeigt.

```
##Title: LaunchInstanceAndCheckState
    -----
    **Purpose**: This runbook first launches an EC2 instance using the AMI
 ID provided in the parameter ```imageId```. The second step of this runbook
 continuously checks the instance status check value for the launched instance
 until the status ```ok``` is returned.

    ##Parameters:
    -----
    Name | Type | Description | Default Value
    ------------- | ------------- | ------------- | -------------
    assumeRole | String | (Optional) The ARN of the role that allows Automation to
 perform the actions on your behalf. | -
    imageId  | String | (Optional) The AMI ID to use for launching the instance.
 The default value uses the latest Amazon Linux 2023 AMI ID available. | {{ ssm:/
aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-arm64 }}
```

6. Geben Sie unter Assume role (Rolle übernehmen) den ARN der IAM-Service-Rolle für Automation (Rolle übernehmen) für die Automatisierung im Format **`arn:aws:iam::111122223333:role/AutomationServiceRole`** ein. Ersetzen Sie 111122223333 durch Ihre AWS-Konto ID.

   Die von Ihnen angegebene Rolle wird verwendet, um die Berechtigungen bereitzustellen, die zum Starten der Automatisierung erforderlich sind.

   > **Important**
   >
   > Für Runbooks, die sich nicht im Besitz von Amazon befinden und die die `aws:executeScript`-Aktion verwenden, muss eine Rolle angegeben werden. Weitere Informationen finden Sie unter Berechtigungen für die Verwendung von Runbooks.

7. Erweitern Sie Input parameters (Eingabeparameter) und gehen Sie folgendermaßen vor.

   1. Geben Sie unter Parameter name (Parametername) **`imageId`** ein.

2. Wählen Sie für Type (Typ) die Option **`String`** aus.
3. Wählen Sie unter Required (Erforderlich) die Option No.
4. Geben Sie unter Default value (Standardwert) Folgendes ein.

```
{{ ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-arm64 }}
```

> **Note**
>
> Dieser Wert startet eine EC2 Amazon-Instance mit der neuesten Amazon Linux 2023 Amazon Machine Image (AMI) -ID. Wenn Sie ein anderes AMI verwenden möchten, ersetzen Sie den Wert durch Ihre AMI-ID.

5. Geben Sie unter Description (Beschreibung) Folgendes ein.

```
(Optional) The AMI ID to use for launching the instance. The default value uses
 the latest released Amazon Linux 2023 AMI ID.
```

8. Wählen Sie Add a parameter (Parameter hinzufügen), um den zweiten Parameter **`tagValue`** zu erstellen, und geben Sie Folgendes ein.

1. Geben Sie unter Parameter name (Parametername) **`tagValue`** ein.
2. Wählen Sie für Type (Typ) die Option **`String`** aus.
3. Wählen Sie unter Required (Erforderlich) die Option No.
4. Für Default value (Standardwert) geben Sie **`LaunchedBySsmAutomation`** ein. Dadurch wird der Instance der Schlüsselpaarwert des Tags `Name:LaunchedBySsmAutomation` hinzugefügt.
5. Geben Sie unter Description (Beschreibung) Folgendes ein.

```
(Optional) The tag value to add to the instance. The default value is
 LaunchedBySsmAutomation.
```

9. Wählen Sie Add a parameter (Parameter hinzufügen), um den dritten Parameter **`instanceType`** zu erstellen, und geben Sie folgende Informationen ein.

1. Geben Sie unter Parameter name (Parametername) **`instanceType`** ein.
2. Wählen Sie für Type (Typ) die Option **`String`** aus.
3. Wählen Sie unter Required (Erforderlich) die Option No.

4. Für Default value (Standardwert) geben Sie **`t2.micro`** ein.
5. Geben Sie unter Parameter description (Parameterbeschreibung) Folgendes ein.

```
(Optional) The instance type to use for the instance. The default value is
 t2.micro.
```

10. Erweitern Sie Target type (Zieltyp) und wählen Sie **`"/"`**.
11. (Optional) Erweitern Sie Document tags (Dokument-Tags), um Ressourcen-Tags auf Ihr Runbook anzuwenden. Geben Sie für Tag key (Tag-Schlüssel **`Purpose`** und für Tag value (Tag-Wert) **`LaunchInstanceAndCheckState`** ein.
12. Führen Sie im Abschnitt Step 1 (Schritt 1) die folgenden Schritte aus.

    1. Geben Sie unter Step name (Schrittname) diesen beschreibenden Schrittnamen für den ersten Schritt der Automatisierung ein: **`LaunchEc2Instance`**.
    2. Wählen Sie unter Action type (Aktionstyp) die Option Run a script (Skript ausführen) (**`aws:executeScript`**).
    3. Geben Sie unter Description (Beschreibung) eine Beschreibung für den Automation-Schritt ein, wie etwa folgende.

```
**About This Step**

   This step first launches an EC2 instance using the ```aws:executeScript```
 action and the provided script.
```

    4. Erweitern Sie Inputs (Eingaben).
    5. Wählen Sie für Runtime (Laufzeit) die Laufzeitsprache aus, die zum Ausführen des bereitgestellten Skripts verwendet werden soll.
    6. Geben Sie unter Handler **`launch_instance`** ein. Dies ist der Funktionsname, der im folgenden Skript deklariert wird.

> **Note**
>
> Dies ist nicht erforderlich für PowerShell.

    7. Ersetzen Sie für Script (Skript) den Standardinhalt durch Folgendes. Stellen Sie sicher, dass das Skript dem entsprechenden Laufzeitwert entspricht.

Python

```
def launch_instance(events, context):
      import boto3
      ec2 = boto3.client('ec2')

      image_id = events['image_id']
      tag_value = events['tag_value']
      instance_type = events['instance_type']

      tag_config = {'ResourceType': 'instance', 'Tags': [{'Key':'Name',
 'Value':tag_value}]}

      res = ec2.run_instances(ImageId=image_id, InstanceType=instance_type,
 MaxCount=1, MinCount=1, TagSpecifications=[tag_config])

      instance_id = res['Instances'][0]['InstanceId']

      print('[INFO] 1 EC2 instance is successfully launched', instance_id)

      return { 'InstanceId' : instance_id }
```

PowerShell

```
Install-Module AWS.Tools.EC2 -Force
    Import-Module AWS.Tools.EC2

    $payload = $env:InputPayload | ConvertFrom-Json

    $imageid = $payload.image_id

    $tagvalue = $payload.tag_value

    $instanceType = $payload.instance_type

    $type = New-Object Amazon.EC2.InstanceType -ArgumentList $instanceType

    $resource = New-Object Amazon.EC2.ResourceType -ArgumentList 'instance'

    $tag = @{Key='Name';Value=$tagValue}

    $tagSpecs = New-Object Amazon.EC2.Model.TagSpecification
```

```
    $tagSpecs.ResourceType = $resource

    $tagSpecs.Tags.Add($tag)

    $res = New-EC2Instance -ImageId $imageId -MinCount 1 -MaxCount 1 -
InstanceType $type -TagSpecification $tagSpecs

    return @{'InstanceId'=$res.Instances.InstanceId}
```

8. Erweitern Sie Additional inputs (Zusätzliche Eingaben).
9. Wählen Sie für Eingabename die Option InputPayload. Geben Sie unter Input value (Eingabewert) die folgenden YAML-Daten ein.

```
image_id: "{{ imageId }}"
    tag_value: "{{ tagValue }}"
    instance_type: "{{ instanceType }}"
```

13. Erweitern Sie Outputs (Ausgänge) und gehen Sie folgendermaßen vor:
    - Geben Sie unter Name **`payload`** ein.
    - Geben Sie für Selector (Selektor) **`$.Payload`** ein.
    - Wählen Sie für Type (Typ) die Option `StringMap` aus.
14. Klicken Sie auf Schritt hinzufügen, um dem Runbook einen zweiten Schritt hinzuzufügen. Der zweite Schritt fragt den Status der in Schritt 1 gestarteten Instance ab und wartet, bis der zurückgegebene Status `ok` ist.
15. Gehen Sie im Abschnitt Step 2 (Schritt 2) folgendermaßen vor.
    1. Geben Sie unter Step name (Schrittname) diesen beschreibenden Namen für den zweiten Schritt der Automatisierung ein: **`WaitForInstanceStatusOk`**.
    2. Wählen Sie unter Action type (Aktionstyp) die Option Run a script (Skript ausführen) (**`aws:executeScript`**).
    3. Geben Sie unter Description (Beschreibung) eine Beschreibung für den Automation-Schritt ein, wie etwa folgende.

```
**About This Step**

    The script continuously polls the instance status check value for the
 instance launched in Step 1 until the ```ok``` status is returned.
```

4. Bei Runtime (Laufzeit) wählen Sie die Laufzeitsprache für die Ausführung des bereitgestellten Skripts verwendet werden soll.
5. Geben Sie unter Handler **`poll_instance`** ein. Dies ist der Funktionsname, der im folgenden Skript deklariert wird.

> **Note**
>
> Dies ist nicht erforderlich für PowerShell.

6. Ersetzen Sie für Script (Skript) den Standardinhalt durch Folgendes. Stellen Sie sicher, dass das Skript dem entsprechenden Laufzeitwert entspricht.

   Python

```
def poll_instance(events, context):
      import boto3
      import time

      ec2 = boto3.client('ec2')

      instance_id = events['InstanceId']

      print('[INFO] Waiting for instance status check to report ok',
 instance_id)

      instance_status = "null"

      while True:
        res = ec2.describe_instance_status(InstanceIds=[instance_id])

        if len(res['InstanceStatuses']) == 0:
          print("Instance status information is not available yet")
          time.sleep(5)
          continue

        instance_status = res['InstanceStatuses'][0]['InstanceStatus']
['Status']

        print('[INFO] Polling to get status of the instance', instance_status)

        if instance_status == 'ok':
          break
```

```
        time.sleep(10)

    return {'Status': instance_status, 'InstanceId': instance_id}
```

PowerShell

```
Install-Module AWS.Tools.EC2 -Force

$inputPayload = $env:InputPayload | ConvertFrom-Json

$instanceId = $inputPayload.payload.InstanceId

$status = Get-EC2InstanceStatus -InstanceId $instanceId

while ($status.Status.Status -ne 'ok'){
   Write-Host 'Polling get status of the instance', $instanceId

   Start-Sleep -Seconds 5

   $status = Get-EC2InstanceStatus -InstanceId $instanceId
}

return @{Status = $status.Status.Status; InstanceId = $instanceId}
```

7. Erweitern Sie Additional inputs (Zusätzliche Eingaben).
8. Wählen Sie für Eingabename die Option InputPayload. Geben Sie unter Input value (Eingabewert) Folgendes ein:

```
{{ LaunchEc2Instance.payload }}
```

16. Wählen Sie Create automation (Automation erstellen), um das Runbook zu speichern.

Verwenden von Skripten in Runbooks

Automation-Runbooks unterstützen das Ausführen von Skripts im Rahmen der Automatisierung. Automation ist ein Tool in AWS Systems Manager. Mithilfe von Runbooks können Sie Skripts direkt in AWS ausführen, ohne eine separate Datenverarbeitungsumgebung zum Ausführen Ihrer Skripts zu erstellen. Da Runbooks Skriptschritte neben anderen Automation-Schritttypen wie Genehmigungen ausführen können, haben Sie in kritischen oder unklaren Situationen die Möglichkeit, manuell

einzugreifen. Sie können die Ausgabe von `aws:executeScript`-Aktionen in Ihren Runbooks zu Amazon CloudWatch Logs verwenden. Weitere Informationen finden Sie unter Protokollierung der Automation-Aktionsausgabe mit CloudWatch Logs.

Berechtigungen für die Verwendung von Runbooks

Um ein Runbook zu verwenden, muss Systems Manager die Berechtigungen einer AWS Identity and Access Management(IAM)-Rolle verwenden. Die Methode, die Automation verwendet, um zu bestimmen, von welcher Rolle die Berechtigungen verwendet werden, hängt von einigen Faktoren und davon ab, ob ein Schritt die `aws:executeScript`-Aktion verwendet.

Für Runbooks, die `aws:executeScript` nicht verwenden, verwendet Automation eine von zwei Berechtigungsquellen:

- Die Berechtigungen einer IAM-Service-Rolle oder einer Assume-Rolle, die im Runbook angegeben oder als Parameter übergeben wird.
- Wenn keine IAM-Servicerolle angegeben ist, werden die Berechtigungen des IAM-Benutzers verwendet, der die Automatisierung gestartet hat.

Wenn ein Schritt in einem Runbook die `aws:executeScript`-Aktion enthält, ist jedoch immer eine IAM-Service-Rolle (Rolle übernehmen) erforderlich, wenn das für die Aktion angegebene Python- oder PowerShell-Skript jegliche AWS-API-Operationen aufruft. Automation prüft diese Rolle in der folgenden Reihenfolge:

- Die Berechtigungen einer IAM-Service-Rolle oder einer Assume-Rolle, die im Runbook angegeben oder als Parameter übergeben wird.
- Wenn keine Rolle gefunden wird, versucht Automation, das für `aws:executeScript` angegebene Python- oder PowerShell-Skript ohne Berechtigungen auszuführen. Wenn das Skript einen AWS-API-Vorgang (z. B. die Amazon EC2 `CreateImage`-Operation) aufruft oder versucht, eine Aktion an einer AWS-Ressource (z. B. eine EC2-Instance) durchzuführen, schlägt der Schritt, der das Skript enthält, fehl und Systems Manager gibt eine Fehlermeldung zurück, die den Fehler meldet.

Hinzufügen von Skripts zu Runbooks

Sie können Skripts zu Runbooks hinzufügen, indem Sie das Skript inline als Teil eines Schritts in das Runbook einfügen. Sie können Skripts auch an das Runbook anhängen, indem Sie die Skripts von Ihrem lokalen Computer hochladen oder einen Amazon Simple Storage Service (Amazon S3)-Bucket angeben, in dem sich die Skripts befinden. Nachdem ein Schritt abgeschlossen ist, in dem

ein Skript ausgeführt wird, steht die Ausgabe des Skripts als JSON-Objekt zur Verfügung, das Sie dann als Eingabe für nachfolgende Schritte im Runbook verwenden können. Weitere Informationen zur `aws:executeScript`-Aktion und zur Verwendung von Anlagen für Skripts finden Sie unter `aws:executeScript` - Führen Sie ein Skript aus.

Skripteinschränkungen für Runbooks

Runbooks erzwingen ein Limit von fünf Dateianhängen. Skripts können entweder in Form eines Python-Skripts (.py), eines PowerShell Core-Skripts (.ps1) oder als Inhalt einer ZIP-Datei angehängt werden.

Verwendung bedingter Anweisungen in Runbooks

Standardmäßig werden die Schritte, die Sie im Abschnitt `mainSteps` eines Runbooks definieren, nacheinander ausgeführt. Wenn eine Aktion abgeschlossen ist, beginnt die nächste im Abschnitt `mainSteps` angegebene Aktion. Wenn eine Aktion nicht erfolgreich ausgeführt wird, schlägt (standardmäßig) die gesamte Automatisierung fehl. Sie können die Automation-Aktion `aws:branch` und die in diesem Abschnitt beschriebenen Optionen für das Runbook zum Erstellen von Automatisierungen verwenden, die bedingte Verzweigungen durchführen. Dies bedeutet, dass Sie Automatisierungen erstellen können, die zu einem anderen Schritt springen, nachdem verschiedene Optionen bewertet wurden oder dynamisch auf Änderungen beim Abschluss eines Schrittes reagieren. Hier finden Sie eine Liste der Optionen, die Sie verwenden können, um dynamische Automatisierungen zu erstellen.

- **`aws:branch`**: Diese Automatisierungsaktion erlaubt das Erstellen einer dynamischen Automatisierung, die mehrere Auswahlmöglichkeiten in einem einzigen Schritt evaluiert und dann auf der Grundlage dieser Evaluierung zu einem anderen Schritt in dem Runbook springt.
- **`nextStep`**: Diese Option gibt an, welcher Schritt in einer Automatisierung nach dem erfolgreichem Abschluss eines Schritts als nächster auszuführen ist.
- **`isEnd`**: Diese Option stoppt eine Automatisierung am Ende eines bestimmten Schrittes. Der Standardwert für diese Option ist "false".
- **`isCritical`**: Diese Option bezeichnet einen Schritt als kritisch für den erfolgreichen Abschluss der Automatisierung. Wenn ein Schritt mit dieser Bezeichnung fehlschlägt, meldet Automation den Endstatus der Automatisierung als `Failed`. Der Standardwert für diese Option ist `true`.
- **`onFailure`**: Diese Option gibt an, ob die Automatisierung bei einem Fehler abgebrochen, fortgesetzt oder bis zu einem bestimmten Schritt übersprungen werden soll. Der Standardwert für diese Option ist "abort".

Der folgende Abschnitt beschreibt die Automation-Aktion `aws:branch`. Weitere Informationen über die Optionen `nextStep`, `isEnd`, `isCritical` und `onFailure` finden Sie unter Beispiel `aws:branch`-Runbooks.

Arbeiten mit der **`aws:branch`**-Aktion

Die Aktion `aws:branch` bietet die dynamischsten Optionen für bedingte Verzweigungen für Automatisierungen. Wie bereits erwähnt, erlaubt diese Aktion, dass Ihre Automatisierung mehrere Bedingungen in einem einzigen Schritt evaluiert und dann auf der Grundlage der Ergebnisse dieser Bewertung zu einem neuen Schritt springt. Die Aktion `aws:branch` funktioniert wie eine `IF-ELIF-ELSE`-Anweisung beim Programmieren.

Hier ist ein YAML-Beispiel für einen `aws:branch`-Schritt:

```
- name: ChooseOSforCommands
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runPowerShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Windows
    - NextStep: runShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Linux
    Default:
      PostProcessing
```

Wenn Sie die Aktion `aws:branch` für einen Schritt angeben, geben Sie die `Choices` an, die die Automatisierung evaluieren muss. Die Automatisierung kann `Choices` auf der Grundlage des Parameters evaluieren, den Sie im Abschnitt `Parameters` des Runbooks angegeben haben. Die Automatisierung kann `Choices` auch auf der Grundlage der Ausgabe eines vorherigen Schritts evaluieren.

Die Automatisierung evaluiert jede Auswahl mithilfe eines booleschen Ausdrucks. Wenn die Evaluierung zu dem Schluss kommt, dass die erste Auswahl `true` ist, springt die Automatisierung zum nächsten Schritt für diese Auswahl. Wenn die Auswertung zu dem Schluss kommt, dass die erste Auswahl `false` ist, evaluiert die Automatisierung die nächste Auswahl. Wenn Ihr Schritt drei oder mehr `Choices` beinhaltet, evaluiert die Automatisierung die Auswahlen nacheinander, bis eine Auswahl als `true` evaluiert wird. Die Automatisierung springt dann zu dem für die als `true` evaluierte Auswahl angegebenen Schritt.

Wenn keine `Choices` als `true` evaluiert werden, prüft die Automatisierung, ob der Schritt einen `Default`-Wert enthält. Ein `Default`-Wert definiert einen Schritt, zu dem die Automatisierung springen soll, wenn keine der Auswahlmöglichkeiten als `true` evaluiert wird. Wenn kein `Default`-Wert für den Schritt definiert ist, verarbeitet die Automatisierung den nächsten Schritt in dem Runbook.

Hier ist ein `aws:branch`-Schritt in YAML mit dem Namen chooseOSfromParameter. Der Schritt beinhaltet zwei `Choices`: (`NextStep: runWindowsCommand`) und (`NextStep: runLinuxCommand`). Die Automatisierung evaluiert diese `Choices`, um zu bestimmen, welcher Befehl für das entsprechende Betriebssystem ausgeführt werden soll. Die `Variable` für jede Auswahl verwendet `{{OSName}}`. Dabei handelt es sich um einen Parameter, den der Autor des Runbooks im Abschnitt `Parameters` des Runbooks festgelegt hat.

```
mainSteps:
- name: chooseOSfromParameter
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runWindowsCommand
      Variable: "{{OSName}}"
      StringEquals: Windows
    - NextStep: runLinuxCommand
      Variable: "{{OSName}}"
      StringEquals: Linux
```

Hier ist ein `aws:branch`-Schritt in YAML mit dem Namen chooseOSfromOutput. Der Schritt beinhaltet zwei `Choices`: (`NextStep: runPowerShellCommand`) und (`NextStep: runShellCommand`). Die Automatisierung evaluiert diese `Choices`, um zu bestimmen, welcher Befehl für das entsprechende Betriebssystem ausgeführt werden soll. Die `Variable` für jede Auswahl verwendet `{{GetInstance.platform}}`. Dies ist die Ausgabe aus einem früheren Schritt in dem Runbook. Dieses Beispiel enthält auch eine Option mit dem Namen `Default`. Wenn die Automatisierung beide `Choices` evaluiert und keine davon `true` ist, springt die Automatisierung zu einem Schritt mit dem Namen `PostProcessing`.

```
mainSteps:
- name: chooseOSfromOutput
  action: aws:branch
  inputs:
    Choices:
```

```
    - NextStep: runPowerShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Windows
    - NextStep: runShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Linux
    Default:
      PostProcessing
```

Erstellen eines **aws:branch**-Schritts in einem Runbook

Wenn Sie einen `aws:branch`-Schritt in einem Runbook erstellen, definieren Sie die `Choices`, die die Automatisierung evaluieren soll, um festzustellen, zu welchem Schritt die Automatisierung dann springen soll. Wie bereits erwähnt, werden `Choices` mit einem booleschen Ausdruck evaluiert. Jede Auswahl muss die folgenden Optionen definieren:

- NextStep: Der nächste Schritt in dem Runbook, der zu verarbeiten ist, wenn die betreffende Auswahl `true` ist.
- Variable: Geben Sie entweder den Namen eines Parameters an, der im Abschnitt `Parameters` des Runbooks definiert ist, einer Variable, die im Abschnitt `Variables` definiert ist, oder geben Sie ein Ausgabeobjekt aus einem vorherigen Schritt in dem Runbook an.

  Geben Sie Variablenwerte mithilfe des folgenden Formulars an.

  `Variable: "{{variable name}}"`

  Geben Sie Parameterwerte mithilfe ders folgenden Formulars an.

  `Variable: "{{parameter name}}"`

  Geben Sie Ausgabeobjektvariablen in der folgenden Form an.

  `Variable: "{{previousStepName.outputName}}"`

  > **Note**
  >
  > Das Erstellen der Ausgabevariable wird im nächsten Abschnitt ausführlicher beschrieben: Informationen zum Erstellen der Ausgabevariable.

- Operation: Die Kriterien für die Evaluierung der Auswahl, etwa `StringEquals: Linux`. Die Aktion `aws:branch` unterstützt die folgenden Operationen:

Zeichenfolgenoperationen

- StringEquals
- EqualsIgnoreCase
- StartsWith
- EndsWith
- Enthält

Numerische Operationen

- NumericEquals
- NumericGreater
- NumericLesser
- NumericGreaterOrEquals
- NumericLesser
- NumericLesserOrEquals

Boolesche Operation

- BooleanEquals

> **Important**
>
> Wenn Sie ein Runbook erstellen, validiert das System alle Operationen im Runbook. Wenn eine Operation nicht unterstützt wird, gibt das System einen Fehler aus, wenn Sie versuchen, das Runbook zu erstellen.

- Default: Geben Sie einen Rückfallschritt an, zu dem die Automatisierung springen soll, wenn keine der `Choices` `true` ist.

> **Note**
>
> Wenn Sie keinen `Default`-Wert angeben möchten, können Sie die `isEnd`-Option angeben. Wenn keine der `Choices` `true` ist und kein `Default`-Wert angegeben ist, wird die Automatisierung am Ende des Schrittes angehalten.

Verwenden Sie die folgenden Vorlagen für die Konstruktion des Schrittes `aws:branch` in Ihrem Runbook. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

YAML

```
mainSteps:
- name: step name
  action: aws:branch
  inputs:
    Choices:
    - NextStep: step to jump to if evaluation for this choice is true
      Variable: "{{parameter name or output from previous step}}"
      Operation type: Operation value
    - NextStep: step to jump to if evaluation for this choice is true
      Variable: "{{parameter name or output from previous step}}"
      Operation type: Operation value
    Default:
      step to jump to if all choices are false
```

JSON

```
{
   "mainSteps":[
      {
         "name":"a name for the step",
         "action":"aws:branch",
         "inputs":{
            "Choices":[
               {
                  "NextStep":"step to jump to if evaluation for this choice is
 true",
                  "Variable":"{{parameter name or output from previous step}}",
                  "Operation type":"Operation value"
               },
               {
                  "NextStep":"step to jump to if evaluation for this choice is
 true",
                  "Variable":"{{parameter name or output from previous step}}",
                  "Operation type":"Operation value"
               }
            ],
            "Default":"step to jump to if all choices are false"
```

```
        }
      }
    ]
}
```

Informationen zum Erstellen der Ausgabevariable

Um eine `aws:branch`-Auswahl zu erstellen, die auf die Ausgabe eines vorherigen Schrittes verweist, müssen Sie den Namen des vorherigen Schrittes und den des Ausgabefeldes angeben. Anschließend kombinieren Sie die Namen des Schrittes und des Feldes im folgenden Format.

`Variable: "{{previousStepName.outputName}}"`

Beispielsweise hat der erste Schritt im folgenden Beispiel den Namen `GetInstance`. Dann gibt es unter `outputs` ein Feld mit dem Namen `platform`. Im zweiten Schritt (`ChooseOSforCommands`) möchte der Autor auf die Ausgabe des Plattform-Feldes als Variable verweisen. Um die Variable zu erstellen, kombinieren Sie einfach den Namen des Schritts (GetInstance) und den Namen des Ausgabefeldes (platform) zu `Variable: "{{GetInstance.platform}}"`.

```
mainSteps:
- Name: GetInstance
  action: aws:executeAwsApi
  inputs:
    Service: ssm
    Api: DescribeInstanceInformation
    Filters:
    - Key: InstanceIds
      Values: ["{{ InstanceId }}"]
  outputs:
  - Name: myInstance
    Selector: "$.InstanceInformationList[0].InstanceId"
    Type: String
  - Name: platform
    Selector: "$.InstanceInformationList[0].PlatformType"
    Type: String
- name: ChooseOSforCommands
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runPowerShellCommand
      Variable: "{{GetInstance.platform}}"
```

```
      StringEquals: Windows
    - NextStep: runShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Linux
    Default:
      Sleep
```

Hier ist ein Beispiel, das zeigt, wie *„Variable“: „{{ descriptionInstance.Platform }}“* aus dem vorherigen Schritt und der Ausgabe erstellt wird.

```
- name: describeInstance
  action: aws:executeAwsApi
  onFailure: Abort
  inputs:
    Service: ec2
    Api: DescribeInstances
    InstanceIds:
    - "{{ InstanceId }}"
  outputs:
  - Name: Platform
    Selector: "$.Reservations[0].Instances[0].Platform"
    Type: String
  nextStep: branchOnInstancePlatform
- name: branchOnInstancePlatform
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runEC2RescueForWindows
      Variable: "{{ describeInstance.Platform }}"
      StringEquals: windows
    Default: runEC2RescueForLinux
```

Beispiel **`aws:branch`**-Runbooks

Hier sind einige Beispiele für Runbooks, die `aws:branch` verwenden.

Beispiel 1: Verwendung von **`aws:branch`** mit einer Ausgabevariablen zur Ausführung von Befehlen auf der Grundlage des Betriebssystemtyps

Im ersten Schritt dieses Beispiels (`GetInstance`) verwendet der Runbook-Autor die `aws:executeAwsApi`-Aktion zum Aufrufen der `ssm` `DescribeInstanceInformation`-API-Operation. Der Autor verwendet diese Aktion, um den Typ des von einer Instance zu verwendenden

Betriebssystems zu bestimmen. Die Aktion `aws:executeAwsApi` gibt die Instance-ID und den Plattformtyp aus.

Im zweiten Schritt (`ChooseOSforCommands`) verwendet der Autor die Aktion `aws:branch` mit zwei `Choices` (`NextStep: runPowerShellCommand`) und (`NextStep: runShellCommand`). Die Automatisierung evaluiert das Betriebssystem der Instance anhand der Ausgabe des vorherigen Schritts (`Variable: "{{GetInstance.platform}}"`). Die Automatisierung springt zu einem Schritt für das angegebene Betriebssystem.

```
---
schemaVersion: '0.3'
assumeRole: "{{AutomationAssumeRole}}"
parameters:
  AutomationAssumeRole:
    default: ""
    type: String
mainSteps:
- name: GetInstance
  action: aws:executeAwsApi
  inputs:
    Service: ssm
    Api: DescribeInstanceInformation
  outputs:
  - Name: myInstance
    Selector: "$.InstanceInformationList[0].InstanceId"
    Type: String
  - Name: platform
    Selector: "$.InstanceInformationList[0].PlatformType"
    Type: String
- name: ChooseOSforCommands
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runPowerShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Windows
    - NextStep: runShellCommand
      Variable: "{{GetInstance.platform}}"
      StringEquals: Linux
    Default:
      Sleep
- name: runShellCommand
  action: aws:runCommand
```

```
  inputs:
    DocumentName: AWS-RunShellScript
    InstanceIds:
    - "{{GetInstance.myInstance}}"
    Parameters:
      commands:
      - ls
  isEnd: true
- name: runPowerShellCommand
  action: aws:runCommand
  inputs:
    DocumentName: AWS-RunPowerShellScript
    InstanceIds:
    - "{{GetInstance.myInstance}}"
    Parameters:
      commands:
      - ls
  isEnd: true
- name: Sleep
  action: aws:sleep
  inputs:
    Duration: PT3S
```

Beispiel 2: Verwendung von **`aws:branch`** mit einer Parametervariablen zur Ausführung von Befehlen auf der Grundlage des Betriebssystemtyps

Der Autor des Runbooks definiert verschiedene Parameteroptionen am Anfang des Runbooks im Abschnitt `parameters`. Ein Parameter hat den Namen `OperatingSystemName`. Im ersten Schritt (`ChooseOS`) verwendet der Autor die Aktion `aws:branch` mit zwei `Choices` (`NextStep: runWindowsCommand`) und (`NextStep: runLinuxCommand`). Die Variable für diese `Choices` verweist auf die im Parameter-Abschnitt angegebene Parameteroption (`Variable: "{{OperatingSystemName}}"`). Wenn der Benutzer dieses Runbook ausführt, gibt er zur Laufzeit einen Wert für `OperatingSystemName` an. Die Automatisierung verwendet den Laufzeitparameter während der Evaluierung der `Choices`. Die Automatisierung springt zu einem Schritt für das angegebene Betriebssystem auf der Grundlage des für `OperatingSystemName` angegebenen Laufzeitparameters.

```
---
schemaVersion: '0.3'
assumeRole: "{{AutomationAssumeRole}}"
parameters:
```

```
  AutomationAssumeRole:
    default: ""
    type: String
  OperatingSystemName:
    type: String
  LinuxInstanceId:
    type: String
  WindowsInstanceId:
    type: String
mainSteps:
- name: ChooseOS
  action: aws:branch
  inputs:
    Choices:
    - NextStep: runWindowsCommand
      Variable: "{{OperatingSystemName}}"
      StringEquals: windows
    - NextStep: runLinuxCommand
      Variable: "{{OperatingSystemName}}"
      StringEquals: linux
    Default:
      Sleep
- name: runLinuxCommand
  action: aws:runCommand
  inputs:
    DocumentName: "AWS-RunShellScript"
    InstanceIds:
    - "{{LinuxInstanceId}}"
    Parameters:
      commands:
      - ls
  isEnd: true
- name: runWindowsCommand
  action: aws:runCommand
  inputs:
    DocumentName: "AWS-RunPowerShellScript"
    InstanceIds:
    - "{{WindowsInstanceId}}"
    Parameters:
      commands:
      - date
  isEnd: true
- name: Sleep
  action: aws:sleep
```

```
  inputs:
    Duration: PT3S
```

Erstellen komplexer verzweigender Automatisierungen mit Operatoren

Sie können Automatisierungen mit komplexen Verzweigungen erstellen, indem Sie die Operatoren `And`, `Or` und `Not` in Ihren `aws:branch`-Schritten verwenden.

Der „Und“-Operator

Verwenden Sie den `And`-Operator, wenn Sie wünschen, dass mehrere Variablen für eine Auswahl `true` sind. Im folgenden Beispiel wird die erste Wahl darauf evaluiert, ob eine Instance `running` ist und das Betriebssystem `Windows` verwendet. Wenn die Evaluierung beider dieser Variablen „true“ ergibt, springt die Automatisierung zum Schritt `runPowerShellCommand`. Wenn eine oder mehrere der Variablen `false` ist, evaluiert die Automatisierung die Variablen für die zweite Auswahl.

```
mainSteps:
- name: switch2
  action: aws:branch
  inputs:
    Choices:
    - And:
      - Variable: "{{GetInstance.pingStatus}}"
        StringEquals: running
      - Variable: "{{GetInstance.platform}}"
        StringEquals: Windows
      NextStep: runPowerShellCommand

    - And:
      - Variable: "{{GetInstance.pingStatus}}"
        StringEquals: running
      - Variable: "{{GetInstance.platform}}"
        StringEquals: Linux
      NextStep: runShellCommand
    Default:
      sleep3
```

Der „Oder“-Operator

Verwenden Sie den `Or`-Operator, wenn Sie wünschen, eine beliebige von mehreren Variablen für eine Auswahl „true“ ist. Im folgenden Beispiel wird die erste Auswahl darauf evaluiert, ob eine

Parameterzeichenfolge `Windows` ist, und ob die Ausgabe eines AWS Lambda-Schrittes „true“ ist. Wenn die Evaluierung feststellt, dass eine dieser Variablen „true“ ist, springt die Automatisierung zum Schritt `RunPowerShellCommand`. Wenn beide Variablen „false“ sind, evaluiert die Automatisierung die Variablen für die zweite Auswahl.

```
- Or:
  - Variable: "{{parameter1}}"
    StringEquals: Windows
  - Variable: "{{BooleanParam1}}"
    BooleanEquals: true
  NextStep: RunPowershellCommand
- Or:
  - Variable: "{{parameter2}}"
    StringEquals: Linux
  - Variable: "{{BooleanParam2}}"
    BooleanEquals: true
  NextStep: RunShellScript
```

Der „Nicht“-Operator

Verwenden Sie den `Not`-Operator, wenn zu einem Schritt gesprungen werden soll, wenn eine Variable nicht „true“ ist. Im folgenden Beispiel wird die erste Auswahl danach evaluiert, ob eine Parameterzeichenfolge `Not Linux` ist. Wenn die Evaluierung feststellt, dass die Variable nicht „Linux“ ist, springt die Automatisierung zum Schritt `sleep2`. Wenn die Evaluierung der ersten Auswahl feststellt, dass sie Linux ist, evaluiert die Automatisierung die nächste Auswahl.

```
mainSteps:
- name: switch
  action: aws:branch
  inputs:
    Choices:
    - NextStep: sleep2
      Not:
        Variable: "{{testParam}}"
        StringEquals: Linux
    - NextStep: sleep1
      Variable: "{{testParam}}"
      StringEquals: Windows
    Default:
      sleep3
```

Beispiele für die Verwendung von bedingten Optionen

Dieser Abschnitt enthält verschiedene Beispiele für die Verwendung dynamischer Optionen in einem Runbook. Jedes Beispiel in diesem Abschnitt erweitert das nachfolgende Runbook. Dieses Runbook verfügt über zwei Aktionen. Die erste Aktion hat den Namen `InstallMsiPackage`. Sie verwendet die Aktion `aws:runCommand` zur Installation einer Anwendung auf einer Windows Server-Instance. Die zweite Aktion hat den Namen `TestInstall`. Sie verwendet die Aktion `aws:invokeLambdaFunction` zum Ausführen eines Tests der installierten Anwendung, sofern die Anwendung erfolgreich installiert wurde. Der erste Schritt gibt `onFailure: Abort` an. Dies bedeutet, dass die Ausführung der Automatisierung vor dem zweiten Schritt gestoppt wird, wenn die Anwendung nicht erfolgreich installiert wird.

Beispiel 1: Runbook mit zwei linearen Aktionen

```
---
schemaVersion: '0.3'
description: Install MSI package and run validation.
assumeRole: "{{automationAssumeRole}}"
parameters:
  automationAssumeRole:
    type: String
    description: "(Required) Assume role."
  packageName:
    type: String
    description: "(Required) MSI package to be installed."
  instanceIds:
    type: String
    description: "(Required) Comma separated list of instances."
mainSteps:
- name: InstallMsiPackage
  action: aws:runCommand
  maxAttempts: 2
  onFailure: Abort
  inputs:
    InstanceIds:
    - "{{instanceIds}}"
    DocumentName: AWS-RunPowerShellScript
    Parameters:
      commands:
      - msiexec /i {{packageName}}
- name: TestInstall
  action: aws:invokeLambdaFunction
```

```
  maxAttempts: 1
  timeoutSeconds: 500
  inputs:
    FunctionName: TestLambdaFunction
...
```

Erstellen einer dynamischen Automatisierung, die anhand der Option **`onFailure`** zu verschiedenen Schritten springt

Im folgenden Beispiel werden die Optionen `onFailure: step:`*`step name`*, `nextStep` und `isEnd` zur Erstellung einer dynamischen Automatisierung verwendet. Wenn in diesem Beispiel die Aktion `InstallMsiPackage` fehlschlägt, springt die Automatisierung zu einer Aktion mit dem Namen PostFailure (`onFailure: step:PostFailure`) zur Ausführung einer AWS Lambda-Funktion, die eine Aktion ausführt, falls die Installation fehlschlägt. Wenn die Installation erfolgreich ist, springt die Automatisierung zur Aktion TestInstall (`nextStep: TestInstall`). Die Schritte `TestInstall` und `PostFailure` verwenden die Option `isEnd` (`isEnd: true`), so dass die Automatisierung abschließt, wenn einer dieser Schritte abgeschlossen ist.

> **Note**
>
> Die Verwendung der Option `isEnd` im letzten Schritt des Abschnitts `mainSteps` ist optional. Wenn der letzte Schritt nicht zu anderen Schritten springt, stoppt die Automatisierung nach der Ausführung der Aktion im letzten Schritt.

Beispiel 2: Eine dynamische Automatisierung, die zu verschiedenen Schritten springt

```
mainSteps
- name: InstallMsiPackage
  action: aws:runCommand
  onFailure: step:PostFailure
  maxAttempts: 2
  inputs:
    InstanceIds:
    - "{{instanceIds}}"
    DocumentName: AWS-RunPowerShellScript
    Parameters:
      commands:
      - msiexec /i {{packageName}}
  nextStep: TestInstall
- name: TestInstall
```

```
  action: aws:invokeLambdaFunction
  maxAttempts: 1
  timeoutSeconds: 500
  inputs:
    FunctionName: TestLambdaFunction
  isEnd: true
- name: PostFailure
  action: aws:invokeLambdaFunction
  maxAttempts: 1
  timeoutSeconds: 500
  inputs:
    FunctionName: PostFailureRecoveryLambdaFunction
  isEnd: true
...
```

> **Note**
>
> Vor der Verarbeitung eines Runbooks überprüft das System, dass das Runbook keine Endlosschleife erstellt. Wenn eine Endlosschleife erkannt wird, gibt Automation einen Fehler und einen Kreis-Trace zurück, aus dem hervorgeht, welche Schritte die Schleife erzeugen.

Erstellen einer dynamischen Automatisierung, die entscheidende Schritte definiert

Sie können angeben, dass ein Schritt für den Erfolg der Automatisierung entscheidend ist. Wenn ein solcher kritischer Schritt fehlschlägt, meldet Automation den Status der Automatisierung als `Failed`. Dies gilt auch dann, wenn ein oder mehrere Schritte erfolgreich ausgeführt wurden. Im folgenden Beispiel identifiziert der Benutzer den Schritt VerifyDependencies, wenn der Schritt InstallMsiPackage fehlschlägt (`onFailure: step:VerifyDependencies`). Der Benutzer gibt an, dass der Schritt `InstallMsiPackage` nicht kritisch ist (`isCritical: false`). In diesem Beispiel gilt: Wenn die Anwendung nicht installiert werden konnten, verarbeitet Automation den Schritt `VerifyDependencies`, um zu bestimmen, ob eine oder mehrere Abhängigkeiten fehlen, was dazu führte, dass die Anwendung nicht installiert werden konnte.

Beispiel 3: Definieren von kritischen Schritten für die Automatisierung

```
---
name: InstallMsiPackage
action: aws:runCommand
onFailure: step:VerifyDependencies
isCritical: false
```

```
maxAttempts: 2
inputs:
  InstanceIds:
  - "{{instanceIds}}"
  DocumentName: AWS-RunPowerShellScript
  Parameters:
    commands:
    - msiexec /i {{packageName}}
nextStep: TestPackage
...
```

Verwenden von Aktionsausgaben als Eingaben

Verschiedene Automatisierungs-Aktionen geben vordefinierte Ausgaben zurück. Sie können diese Ausgaben mithilfe des Formats `{{stepName.outputName}}` als Eingaben an spätere Schritte in Ihrem Runbook übergeben. Sie können benutzerdefinierte Ausgaben für Automatisierungs-Aktionen in Ihren Runbooks definieren. Auf diese Weise können Sie Skripts ausführen oder API-Operationen für andere AWS-Services einmalig aufrufen, sodass Sie die Werte als Eingaben in späteren Aktionen wiederverwenden können. Parametertypen in Runbooks sind statisch. Dies bedeutet, dass der Parametertyp nicht geändert werden kann, nachdem er definiert wurde. Um eine Schrittausgabe zu definieren, geben Sie die folgenden Felder an:

- Name: (Erforderlich) Der Ausgabename, der in späteren Schritten verwendet wird, um auf den Ausgabewert zu verweisen.
- Selektor: (Erforderlich) Der JSONPath Ausdruck, der zur Bestimmung des Ausgabewerts verwendet wird.
- Typ: (Optional) Der Datentyp des Werts, der vom Auswahlfeld zurückgegeben wird. Gültige Typwerte sind `String`, `Integer`, `Boolean`, `StringList`, `StringMap`, `MapList`. Der Standardwert ist `String`.

Wenn der Wert einer Ausgabe nicht dem von Ihnen angegebenen Datentyp entspricht, versucht Automation, den Datentyp zu konvertieren. Wenn der zurückgegebene Wert beispielsweise ein `Integer` ist, der angegebene `Type` jedoch ein `String` ist, ist der endgültige Ausgabewert ein `String`-Wert. Die folgenden Typkonvertierungen werden unterstützt:

- `String`-Werte können in `StringList`, `Integer` und `Boolean` umgewandelt werden.
- `Integer`-Werte können in `String` und `StringList` umgewandelt werden.
- `Boolean`-Werte können in `String` und `StringList` umgewandelt werden.

- `StringList`-, `IntegerList`-, oder `BooleanList`-Werte, die ein Element enthalten, können in `String`, `Integer` oder `Boolean` umgewandelt werden.

Bei der Verwendung von Parametern oder Ausgaben mit Automatisierungs-Aktionen kann der Datentyp nicht dynamisch innerhalb der Eingabe einer Aktion geändert werden.

Hier ist ein Beispiel-Runbook, das veranschaulicht, wie Sie Aktionsausgaben definieren und auf den Wert als Eingabe für eine spätere Aktion verweisen. Die Runbooks tun Folgendes:

- Verwendet die `aws:executeAwsApi` Aktion, um den EC2 DescribeImages Amazon-API-Vorgang aufzurufen, um den Namen eines bestimmten Windows Server 2016 abzurufenAMI. Es gibt die Image-ID als `ImageId` aus.
- Verwendet die `aws:executeAwsApi` Aktion, um den EC2 RunInstances Amazon-API-Vorgang aufzurufen, um eine Instance zu starten, die den `ImageId` aus dem vorherigen Schritt verwendet. Es gibt die Instance-ID als `InstanceId` aus.
- Verwendet die `aws:waitForAwsResourceProperty` Aktion, um den EC2 DescribeInstanceStatus Amazon-API-Vorgang abzufragen, um darauf zu warten, dass die Instance den `running` Status erreicht. Die Aktion endet nach 60 Sekunden durch Timeout. Der Schritt endet durch Timeout, wenn die Instance nach 60 Sekunden Abfrage nicht den Status `running` erreicht.
- Verwendet die `aws:assertAwsResourceProperty` Aktion, um den EC2 `DescribeInstanceStatus` Amazon-API-Vorgang aufzurufen, um zu bestätigen, dass sich die Instance im `running` Status befindet. Der Schritt schlägt fehl, wenn der Status der Instance nicht `running` ist.

```
---
description: Sample runbook using AWS API operations
schemaVersion: '0.3'
assumeRole: "{{ AutomationAssumeRole }}"
parameters:
  AutomationAssumeRole:
    type: String
    description: "(Optional) The ARN of the role that allows Automation to perform the
 actions on your behalf."
    default: ''
  ImageName:
    type: String
    description: "(Optional) Image Name to launch EC2 instance with."
```

```
    default: "Windows_Server-2022-English-Full-Base*"
mainSteps:
- name: getImageId
  action: aws:executeAwsApi
  inputs:
    Service: ec2
    Api: DescribeImages
    Filters:
    - Name: "name"
      Values:
      - "{{ ImageName }}"
  outputs:
  - Name: ImageId
    Selector: "$.Images[0].ImageId"
    Type: "String"
- name: launchOneInstance
  action: aws:executeAwsApi
  inputs:
    Service: ec2
    Api: RunInstances
    ImageId: "{{ getImageId.ImageId }}"
    MaxCount: 1
    MinCount: 1
  outputs:
  - Name: InstanceId
    Selector: "$.Instances[0].InstanceId"
    Type: "String"
- name: waitUntilInstanceStateRunning
  action: aws:waitForAwsResourceProperty
  timeoutSeconds: 60
  inputs:
    Service: ec2
    Api: DescribeInstanceStatus
    InstanceIds:
    - "{{ launchOneInstance.InstanceId }}"
    PropertySelector: "$.InstanceStatuses[0].InstanceState.Name"
    DesiredValues:
    - running
- name: assertInstanceStateRunning
  action: aws:assertAwsResourceProperty
  inputs:
    Service: ec2
    Api: DescribeInstanceStatus
    InstanceIds:
```

```
    - "{{ launchOneInstance.InstanceId }}"
    PropertySelector: "$.InstanceStatuses[0].InstanceState.Name"
    DesiredValues:
    - running
outputs:
- "launchOneInstance.InstanceId"
...
```

Sie können mit jeder der oben beschriebenen Automatisierungsaktionen eine bestimmte API-Operation aufrufen, indem Sie den Service-Namespace, den Namen der API-Operation, die Eingabeparameter und die Ausgabeparameter angeben. Eingaben werden von der ausgewählten API-Operation bestimmt. Sie können die API-Operationen (auch als Methoden bezeichnet) anzeigen, indem Sie einen Service in der linken Navigationsleiste auf der folgenden Service-Referenzen-Seite auswählen. Wählen Sie eine Methode im Abschnitt Client für den Service, den Sie aufrufen möchten. Beispielsweise werden alle API-Vorgänge (Methoden) für Amazon Relational Database Service (Amazon RDS) auf der folgenden Seite aufgelistet: Amazon RDS-Methoden.

Sie können das Schema für jede Automatisierungsaktion an den folgenden Orten anzeigen:

- `aws:assertAwsResourceProperty`— Bestätigt einen AWS Ressourcen- oder Ereignisstatus
- `aws:executeAwsApi`— AWS API-Operationen aufrufen und ausführen
- `aws:waitForAwsResourceProperty`— Warte auf eine AWS Ressourceneigenschaft

Die Schemata umfassen Beschreibungen der erforderlichen Felder für jede Aktion.

Verwendung der Selector/PropertySelector Felder

Jede Automatisierungsaktion erfordert, dass Sie entweder eine Ausgabe `Selector` (für `aws:executeAwsApi`) oder einen `PropertySelector` (für `aws:assertAwsResourceProperty` und `aws:waitForAwsResourceProperty`) enthalten. Diese Felder werden verwendet, um die JSON-Antwort aus einer AWS API-Operation zu verarbeiten. Diese Felder verwenden die JSONPath Syntax.

Hier finden Sie ein Beispiel, das dieses Konzept für die Aktion `aws:executeAwsAPi` erläutert.

```
---
mainSteps:
- name: getImageId
  action: aws:executeAwsApi
  inputs:
```

```
    Service: ec2
    Api: DescribeImages
    Filters:
      - Name: "name"
        Values:
          - "{{ ImageName }}"
  outputs:
    - Name: ImageId
      Selector: "$.Images[0].ImageId"
      Type: "String"
...
```

Im `aws:executeAwsApi`-Schritt`getImageId` ruft die Automatisierung die `DescribeImages`-API-Operation auf und empfängt eine Antwort von `ec2`. Die Automatisierung wendet dann `Selector - "$.Images[0].ImageId"` auf die API-Antwort an und weist der `ImageId`-Ausgabevariablen den ausgewählten Wert zu. Weitere Schritte in dieser Automatisierung können den Wert von `ImageId` verwenden, indem `"{{ getImageId.ImageId }}"` angegeben wird.

Hier finden Sie ein Beispiel, das dieses Konzept für die Aktion `aws:waitForAwsResourceProperty` erläutert.

```
---
- name: waitUntilInstanceStateRunning
  action: aws:waitForAwsResourceProperty
  # timeout is strongly encouraged for action - aws:waitForAwsResourceProperty
  timeoutSeconds: 60
  inputs:
    Service: ec2
    Api: DescribeInstanceStatus
    InstanceIds:
    - "{{ launchOneInstance.InstanceId }}"
    PropertySelector: "$.InstanceStatuses[0].InstanceState.Name"
    DesiredValues:
    - running
...
```

Im `aws:waitForAwsResourceProperty`-Schritt`waitUntilInstanceStateRunning` ruft die Automatisierung die `DescribeInstanceStatus`-API-Operation auf und empfängt eine Antwort von `ec2`. Die Automatisierung wendet dann `PropertySelector - "$.InstanceStatuses[0].InstanceState.Name"` auf die Antwort an und prüft, ob der angegebene zurückgegebene Wert einem Wert in der Liste `DesiredValues` entspricht (in diesem

Fall `running`). Der Schritt wiederholt den Prozess, bis die Antwort den Instance-Status `running` zurückgibt.

Verwendung JSONPath in Runbooks

Ein JSONPath Ausdruck ist eine Zeichenfolge, die mit „$“ beginnt. die zur Auswahl einer oder mehrerer Komponenten in einem JSON-Element verwendet wird. Die folgende Liste enthält Informationen zu JSONPath Operatoren, die von Systems Manager Automation unterstützt werden:

- Dot-notated child (.): Verwendung mit einem JSON-Objekt. Dieser Operator wählt den Wert eines bestimmten Schlüssels aus.
- Deep-scan (..): Verwendung mit einem JSON-Element. Dieser Operator untersucht das JSON-Element Ebene für Ebene und wählt eine Liste von Werten mit dem spezifischen Schlüssel aus. Der Rückgabetyp dieses Operators ist immer ein JSON-Array. Im Kontext eines Ausgabetyps einer Automatisierungsaktion kann der Operator entweder StringList oder sein MapList.
- Array-Index ([ ]): Verwendung mit einem JSON-Array. Dieser Operator ruft den Wert eines bestimmten Index ab.
- Filter ([? (***expression***)]): Wird mit einem JSON-Array verwendet. Dieser Operator filtert JSON-Array-Werte, die den im Filterausdruck definierten Kriterien entsprechen. Filterausdrücke können nur die folgenden Operatoren verwenden: ==,! =, >, <, >= oder <=. Die Kombination mehrerer Filterausdrücke mit AND (&&) oder OR (||) wird nicht unterstützt. Der Rückgabetyp dieses Operators ist immer ein JSON-Array.

Um JSONPath Operatoren besser zu verstehen, lesen Sie sich die folgende JSON-Antwort aus dem `DescribeInstances` ec2-API-Vorgang durch. Auf diese Antwort folgen mehrere Beispiele, die unterschiedliche Ergebnisse zeigen, indem unterschiedliche JSONPath Ausdrücke auf die Antwort aus dem `DescribeInstances` API-Vorgang angewendet werden.

```
{
    "NextToken": "abcdefg",
    "Reservations": [
        {
            "OwnerId": "123456789012",
            "ReservationId": "r-abcd12345678910",
            "Instances": [
                {
                    "ImageId": "ami-12345678",
                    "BlockDeviceMappings": [
                        {
```

```
                            "Ebs": {
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-000000000000"
                            },
                            "DeviceName": "/dev/xvda"
                        }
                    ],
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    }
                }
            ],
            "Groups": []
        },
        {
            "OwnerId": "123456789012",
            "ReservationId": "r-12345678910abcd",
            "Instances": [
                {
                    "ImageId": "ami-12345678",
                    "BlockDeviceMappings": [
                        {
                            "Ebs": {
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-111111111111"
                            },
                            "DeviceName": "/dev/xvda"
                        }
                    ],
                    "State": {
                        "Code": 80,
                        "Name": "stopped"
                    }
                }
            ],
            "Groups": []
        }
    ]
}
```

JSONPath Beispiel 1: Rufen Sie eine bestimmte Zeichenfolge aus einer JSON-Antwort ab

```
JSONPath:
$.Reservations[0].Instances[0].ImageId

Returns:
"ami-12345678"

Type: String
```

JSONPath Beispiel 2: Rufen Sie einen bestimmten booleschen Wert aus einer JSON-Antwort ab

```
JSONPath:
$.Reservations[0].Instances[0].BlockDeviceMappings[0].Ebs.DeleteOnTermination

Returns:
true

Type: Boolean
```

JSONPath Beispiel 3: Holen Sie sich eine bestimmte Ganzzahl aus einer JSON-Antwort

```
JSONPath:
$.Reservations[0].Instances[0].State.Code

Returns:
16

Type: Integer
```

JSONPath Beispiel 4: Eine JSON-Antwort gründlich scannen und dann alle Werte für VolumeId als StringList

```
JSONPath:
$.Reservations..BlockDeviceMappings..VolumeId

Returns:
[
   "vol-000000000000",
   "vol-111111111111"
]
```

```
Type: StringList
```

JSONPath Beispiel 5: Holen Sie sich ein bestimmtes BlockDeviceMappings Objekt als StringMap

```
JSONPath:
$.Reservations[0].Instances[0].BlockDeviceMappings[0]

Returns:
{
   "Ebs" : {
      "DeleteOnTermination" : true,
      "Status" : "attached",
      "VolumeId" : "vol-000000000000"
   },
   "DeviceName" : "/dev/xvda"
}

Type: StringMap
```

JSONPath Beispiel 6: Tiefenscan einer JSON-Antwort und Abrufen aller State-Objekte als MapList

```
JSONPath:
$.Reservations..Instances..State

Returns:
[
   {
      "Code" : 16,
      "Name" : "running"
   },
   {
      "Code" : 80,
      "Name" : "stopped"
   }
]

Type: MapList
```

JSONPath Beispiel 7: Filtern Sie nach Instanzen im **`running`** Bundesstaat

```
JSONPath:
$.Reservations..Instances[?(@.State.Name == 'running')]
```

```
Returns:
[
  {
    "ImageId": "ami-12345678",
    "BlockDeviceMappings": [
      {
        "Ebs": {
          "DeleteOnTermination": true,
          "Status": "attached",
          "VolumeId": "vol-000000000000"
        },
        "DeviceName": "/dev/xvda"
      }
    ],
    "State": {
      "Code": 16,
      "Name": "running"
    }
  }
]

Type: MapList
```

JSONPath Beispiel 8: Gibt die Anzahl **`ImageId`** der Instanzen zurück, die sich nicht im **`running`** Status befinden

```
JSONPath:
$.Reservations..Instances[?(@.State.Name != 'running')].ImageId

Returns:
[
  "ami-12345678"
]

Type: StringList | String
```

Erstellen von Webhook-Integrationen für Automation

Um während einer Automatisierung Nachrichten über Webhooks zu senden, erstellen Sie eine Integration. Integrationen können während einer Automatisierung mithilfe der neuen Aktion `aws:invokeWebhook` in Ihrem Runbook aufgerufen werden. Wenn Sie noch keinen

Webhook erstellt haben, finden Sie weitere Informationen unter Erstellen von Webhooks für Integrationen. Weitere Informationen über die Aktion `aws:invokeWebhook` finden Sie unter `aws:invokeWebhook` – Automation-Webhook-Integration aufrufen.

Wie in den folgenden Verfahren gezeigt, können Sie Integrationen über die Automation-Konsole von Systems Manager oder mit Ihrem bevorzugten Befehlszeilen-Tool erstellen.

Erstellen von Integrationen (Konsole)

So erstellen Sie eine Integration für Automation (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation.
3. Wählen Sie die Registerkarte Integrations (Integrationen) aus.
4. Wählen Sie Add integration (Integration hinzufügen) und dann Webhook aus.
5. Geben Sie die erforderlichen Werte und optionale Werte ein, die Sie für die Integration einbeziehen möchten.
6. Wählen Sie Add (Hinzufügen) aus, um die Integration zu erstellen.

Erstellen von Integrationen (Befehlszeile)

Um eine Integration mit Befehlszeilen-Tools zu erstellen, muss der erforderliche `SecureString`-Parameter für eine Integration erstellt werden. Automation nutzt einen reservierten Namespace in Parameter Store, einem Tool in Systems Manager, um Informationen über Ihre Integration zu speichern. Wenn Sie eine Integration über die AWS-Managementkonsole erstellen, übernimmt Automation diesen Vorgang für Sie. Nach dem Namespace geben Sie den Typ der zu erstellenden Integration und dann deren Namen an. Derzeit unterstützt Automation Integrationen vom Typ `webhook`.

Folgende Felder werden für Integrationen vom Typ `webhook` unterstützt:

- Beschreibung
- Header
- Nutzlast
- URL

Bevor Sie beginnen

Falls noch nicht erfolgt, installieren und konfigurieren Sie die AWS Command Line Interface (AWS CLI) oder AWS -Tools für PowerShell. Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.

So erstellen Sie eine Integration für Automation (Befehlszeile)

- Führen Sie die folgenden Befehle aus, um den erforderlichen `SecureString`-Parameter für eine Integration zu erstellen. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen. Der Namespace `/d9d01087-4a3f-49e0-b0b4-d568d7826553/ssm/integrations/webhook/` ist im Parameter Store für Integrationen reserviert. Im Namen des Parameters muss dieser Namespace verwendet werden, gefolgt vom Namen der Integration. Zum Beispiel `/d9d01087-4a3f-49e0-b0b4-d568d7826553/ssm/integrations/webhook/`*myWebhookIntegration*.

  Linux & macOS

  ```
  aws ssm put-parameter \
      --name "/d9d01087-4a3f-49e0-b0b4-d568d7826553/ssm/integrations/
  webhook/myWebhookIntegration" \
      --type "SecureString" \
      --data-type "aws:ssm:integration" \
      --value '{"description": "My first webhook integration for Automation.",
   "url": "myWebHookURL"}'
  ```

  Windows

  ```
  aws ssm put-parameter ^
      --name "/d9d01087-4a3f-49e0-b0b4-d568d7826553/ssm/integrations/
  webhook/myWebhookIntegration" ^
      --type "SecureString" ^
      --data-type "aws:ssm:integration" ^
      --value  "{\"description\":\"My first webhook integration for Automation.\",
  \"url\":\"myWebHookURL\"}"
  ```

  PowerShell

  ```
  Write-SSMParameter `
      -Name "/d9d01087-4a3f-49e0-b0b4-d568d7826553/ssm/integrations/
  webhook/myWebhookIntegration" `
  ```

```
    -Type "SecureString"
    -DataType "aws:ssm:integration"
    -Value '{"description": "My first webhook integration for Automation.",
"url": "myWebHookURL"}'
```

Erstellen von Webhooks für Integrationen

Beachten Sie beim Erstellen von Webhooks bei Ihrem Anbieter Folgendes:

- Das Protokoll muss HTTPS lauten.
- Benutzerdefinierte Anforderungs-Header werden unterstützt.
- Ein Standardanforderungstext kann angegeben werden.
- Der Standardanforderungstext kann überschrieben werden, wenn eine Integration mit der Aktion `aws:invokeWebhook` aufgerufen wird.

Behandeln von Timeouts in Runbooks

Die Eigenschaft `timeoutSeconds` wird von allen Automatisierungsaktionen gemeinsam genutzt. Sie können diese Eigenschaft verwenden, um den Ausführungstimeout-Wert für eine Aktion anzugeben. Außerdem können Sie die Auswirkung des Timeouts einer Aktion auf die Automatisierung und den gesamten Ausführungsstatus ändern. Zu diesem Zweck definieren Sie auch die gemeinsam genutzten Eigenschaften `onFailure` und `isCritical` für eine Aktion.

Je nach Anwendungsfall möchten Sie vielleicht, dass Ihre Automatisierung mit einer anderen Aktion fortgesetzt wird und der Gesamtstatus der Automation nicht betroffen ist, wenn es zum Timeout einer Aktion kommt. In diesem Beispiel geben Sie mit der Eigenschaft `timeoutSeconds` an, wie lange gewartet werden soll, bevor es zum Timeout der Aktion kommt. Anschließend geben Sie die Aktion oder den Schritt an, zu dem die Automatisierung bei einem Timeout übergehen soll. Geben Sie einen Wert im Format `step:`*`step name`* für die Eigenschaft `onFailure` anstelle des Standardwerts `Abort` an. Beim Timeout einer Aktion wird der Automatisierungs-Ausführungsstatus standardmäßig `Timed Out` lauten. Um zu verhindern, dass sich ein Timeout auf den Automatisierungs-Ausführungsstatus auswirkt, geben Sie `false` für die Eigenschaft `isCritical` an.

Das folgende Beispiel zeigt, wie die gemeinsam genutzten Eigenschaften für eine in diesem Szenario beschriebene Aktion definiert werden.

YAML

```
- name: verifyImageAvailability
  action: 'aws:waitForAwsResourceProperty'
  timeoutSeconds: 600
  isCritical: false
  onFailure: 'step:getCurrentImageState'
  inputs:
    Service: ec2
    Api: DescribeImages
    ImageIds:
      - '{{ createImage.newImageId }}'
    PropertySelector: '$.Images[0].State'
    DesiredValues:
      - available
  nextStep: copyImage
```

JSON

```
{
    "name": "verifyImageAvailability",
    "action": "aws:waitForAwsResourceProperty",
    "timeoutSeconds": 600,
    "isCritical": false,
    "onFailure": "step:getCurrentImageState",
    "inputs": {
        "Service": "ec2",
        "Api": "DescribeImages",
        "ImageIds": [
            "{{ createImage.newImageId }}"
        ],
        "PropertySelector": "$.Images[0].State",
        "DesiredValues": [
            "available"
        ]
    },
    "nextStep": "copyImage"
}
```

Weitere Informationen zu Eigenschaften, die von allen Automatisierungsaktionen gemeinsam genutzt werden, finden Sie unter Von allen Aktionen gemeinsam genutzte Eigenschaften.

Runbook-Referenz für Systems Manager Automation

AWS Systems Manager Stellt vordefinierte Runbooks bereit, damit Sie schnell loslegen können. Diese Runbooks werden von Amazon Web Services verwaltet, AWS -Support, und AWS Config. Die Runbook-Referenz beschreibt jedes der vordefinierten Runbooks, die von Systems Manager Support, und bereitgestellt werden. AWS Config Weitere Informationen finden Sie unter Referenz zum Systems Manager Automation-Runbook.

Tutorials

Die folgenden Tutorials unterstützen Sie bei der Verwendung von AWS Systems Manager Automation zur Bewältigung gängiger Anwendungsfälle. Diese Tutorials veranschaulichen, wie Sie eigene Runbooks, von Automation bereitgestellte Runbooks und andere Systems-Manager-Tools mit anderen AWS-Services verwenden.

Inhalt

- Aktualisieren von AMIs
  - Aktualisieren eines Linux AMI
  - Aktualisieren eines Linux AMI (AWS CLI)
  - Aktualisieren eines Windows Server-AMI
  - Aktualisieren Sie ein Golden AMI mithilfe von Automation, AWS Lambda, und Parameter Store
    - Aufgabe 1: Erstellen eines Parameters im Systems Manager-Parameter Store
    - Aufgabe 2: Erstellen einer IAM-Rolle für AWS Lambda
    - Aufgabe 3: Erstellen einer AWS Lambda -Funktion
    - Aufgabe 4: Erstellen eines Runbooks und Patchen des AMI
  - Aktualisierung von AMIs mithilfe von Automation und Jenkins
  - Aktualisieren von AMIs für Auto-Scaling-Gruppen
    - Erstellen Sie das Patch AMIAnd UpdateASG-Runbook
- Verwendung von AWS -Support-Self-Service-Runbooks
  - Führen Sie das EC2 Rescue-Tool auf nicht erreichbaren Instanzen aus
    - Funktionsweise
    - Bevor Sie beginnen
      - Gewähren von AWSSupport-EC2Rescue-Berechtigungen zum Durchführen von Aktionen auf Ihren Instances

            - Erteilen von Berechtigungen mithilfe von IAM-Richtlinien
            - Erteilen von Berechtigungen mithilfe einer CloudFormation Vorlage
        - Ausführen der Automation
    - Passwörter und SSH-Schlüssel auf EC2 Instanzen zurücksetzen
        - Funktionsweise
        - Bevor Sie beginnen
            - Erteilen Sie AWSSupport-EC 2Rescue-Berechtigungen zur Durchführung von Aktionen auf Ihren Instances
                - Erteilen von Berechtigungen mithilfe von IAM-Richtlinien
                - Erteilen von Berechtigungen mithilfe einer CloudFormation Vorlage
        - Ausführen der Automation
- Übergabe von Daten an Automation mithilfe von Eingangstransformatoren

Aktualisieren von AMIs

Die folgenden Tutorials erläutern, wie Sie Amazon Machine Image (AMIs) aktualisieren, um die neuesten Patches einzubeziehen.

Themen

- Aktualisieren eines Linux AMI
- Aktualisieren eines Linux AMI (AWS CLI)
- Aktualisieren eines Windows Server-AMI
- Aktualisieren Sie ein Golden AMI mithilfe von Automation, AWS Lambda, und Parameter Store
- Aktualisierung von AMIs mithilfe von Automation und Jenkins
- Aktualisieren von AMIs für Auto-Scaling-Gruppen

Aktualisieren eines Linux AMI

Diese Vorgehensweise für Systems Manager Automation zeigt Ihnen, wie Sie die Konsole oder das AWS CLI - und das `AWS-UpdateLinuxAmi`-Runbook verwenden, um ein Linux-AMI mit den neuesten Patches der von Ihnen angegebenen Pakete zu aktualisieren. Automation ist ein Tool in AWS Systems Manager. Das `AWS-UpdateLinuxAmi`-Runbook automatisiert auch die Installation zusätzlicher websitespezifischer Pakete und Konfigurationen. Sie können mit dieser Anleitung eine Vielzahl von Linux-Distributionen aktualisieren, einschließlich Ubuntu Server, Red Hat Enterprise

Linux (RHEL) oder Amazon Linux AMIs. Eine vollständige Liste der unterstützten Linux-Versionen finden Sie unter Patch Manager-Voraussetzungen.

Mit dem `AWS-UpdateLinuxAmi`-Runbook können Sie Aufgaben zur Imagewartung automatisieren, ohne das Runbook in JSON oder YAML erstellen zu müssen. Sie können das Runbook `AWS-UpdateLinuxAmi` verwenden, um die folgenden Arten von Aufgaben auszuführen.

- Aktualisieren Sie alle Distributionspakete und jegliche Amazon-Software in einem Amazon Machine Image (AMI) mit Amazon Linux, Red Hat Enterprise Linux oder Ubuntu Server. Dies ist das Runbook-Standardverhalten.
- Installieren Sie es AWS Systems Manager SSM Agent auf einem vorhandenen Image, um Systems Manager Manager-Tools zu aktivieren, z. B. das Ausführen von Remotebefehlen mithilfe von AWS Systems Manager Run Command oder die Softwareinventurerfassung mithilfe von Inventar.
- Installieren Sie zusätzliche Softwarepakete.

Bevor Sie beginnen

Bevor Sie mit der Arbeit mit Runbooks beginnen, konfigurieren Sie Rollen und optional die Funktionen EventBridge für die Automatisierung. Weitere Informationen finden Sie unter Einrichten der Automatisierung. Für diese exemplarische Vorgehensweise müssen Sie außerdem den Namen eines AWS Identity and Access Management (IAM-) Instanzprofils angeben. Weitere Informationen zum Erstellen eines IAM-Instance-Profils finden Sie unter Konfigurieren von erforderlichen Instance-Berechtigungen für Systems Manager.

Das Runbook `AWS-UpdateLinuxAmi` akzeptiert die folgenden Eingabeparameter.

Parameter	Typ	Description
SourceAmiId	String	(Erforderlich) Die Quell-AMI-ID.
IamInstanceProfileName	String	(Erforderlich) Der Name der IAM-Instance-Profilrolle, die Sie unter Erforderliche Instance-Berechtigungen für Systems Manager konfigurieren erstellt haben. Die Instance-Profilrolle erteilt der

Parameter	Typ	Description
		Automation die Berechtigung, auf Ihren Instances Aktionen durchzuführen, wie etwa das Ausführen von Befehlen oder das Starten und Beenden von Services. Das Runbook verwendet nur den Namen der Instance-Profilrolle. Wenn Sie den Amazon-Ressourcenn amen (ARN) angeben, schlägt die Automatisierung fehl.
AutomationAssumeRole	String	(Erforderlich) Der Name der IAM-Servicerolle, die Sie in Einrichten der Automatis ierung erstellt haben. Mit der Servicerolle (auch als assume-Rolle bezeichnet) gestatten Sie der Automatis ierung, Ihre IAM-Rolle zu übernehmen und in Ihrem Auftrag Aktionen auszuführen. Mit der Servicerolle gestatten Sie der Automation beispiels weise beim Ausführen der Aktion `aws:createImage` in einem Runbook, ein neues AMI zu erstellen. Für diesen Parameter muss der vollständ ige ARN angegeben werden.

Parameter	Typ	Description
TargetAmiName	String	(Optional) Der Name des neuen AMI nach seiner Erstellung. Der Standardn ame ist eine systemgenerierte Zeichenfolge, die die Quell-AMI-ID sowie Uhrzeit und Datum der Erstellung enthält.
InstanceType	String	(Optional) Der Typ der zu startenden Instance als Arbeitsbereich hosten. Die Instance-Typen sind je nach Region unterschiedlich. Der Standardtyp ist t2.micro.
PreUpdateScript	String	(Optional) Die URL eines Skripts, das ausgeführt werden muss, bevor Updates übernommen werden. Standard („none“) ist die Ausführung keines Skripts.
PostUpdateScript	String	(Optional) Die URL eines Skripts, das ausgeführt werden muss, nachdem Paketupdates angewendet werden. Standard („none“) ist die Ausführung keines Skripts.
IncludePackages	String	(Optional) Aktualisieren Sie nur diese benannten Pakete. Standardmäßig werden alle (\"all\") verfügbaren Updates übernommen.

Parameter	Typ	Description
ExcludePackages	String	(Optional) Namen der Pakete, die bei Updates unter allen Umständen zurückgehalten werden müssen. Standardm äßig wird kein (\"none\") Paket ausgeschlossen.

Automation-Schritte

Das `AWS-UpdateLinuxAmi`-Runbook umfasst standardmäßig die folgenden Automatisierungsaktionen.

Schritt 1: launchInstance (**`aws:runInstances`**-Aktion)

In diesem Schritt wird eine Instance mit Amazon Elastic Compute Cloud (Amazon EC2) -Benutzerdaten und einer IAM-Instance-Profilrolle gestartet. Userdata installiert je nach Betriebssystem den entsprechenden SSM Agent. Durch Installieren von SSM Agent können Sie Systems-Manager-Tools wie Run Command, State Manager und Inventory verwenden.

Schritt 2: Aktualisieren OSSoftware (Aktion) **`aws:runCommand`**

Dieser Schritt führt die folgenden Befehle auf der gestarteten Instance aus:

- Lädt ein Update-Skript aus Amazon S3 herunter.
- Führt ein optionales Pre-Update-Skript aus.
- Aktualisiert Verteilungspakete und Amazon-Software.
- Führt ein optionales Post-Update-Skript aus.

Das Ausführungsprotokoll wird im Ordner /tmp gespeichert, damit es der Benutzer zu einem späteren Zeitpunkt ansehen kann.

Falls Sie eine bestimmte Reihe von Paketen aktualisieren möchten, können Sie die Liste mithilfe des `IncludePackages`-Parameters bereitstellen. Bei der Bereitstellung versucht das System nur diese Pakete und deren abhängige Objekte zu aktualisieren. Es werden keine weiteren Updates vorgenommen. Wenn standardmäßig keine include-Pakete festgelegt sind, aktualisiert das Programm alle verfügbaren Pakete.

Falls Sie eine bestimmte Reihe von Paketen von der Aktualisierung ausschließen möchten, können Sie die Liste mithilfe des `ExcludePackages`-Parameters bereitstellen. Wenn diese Pakete bereitgestellt werden, bleiben sie in ihrer aktuellen Version, unabhängig von anderen festgelegten Optionen. Wenn keine exclude-Pakete festgelegt sind, werden standardmäßig keine Pakete ausgeschlossen.

Schritt 3: StopInstance (**`aws:changeInstanceState`**-Aktion)

Dieser Schritt stoppt die aktualisierte Instance.

Schritt 4: CreateImage (**`aws:createImage`**-Aktion)

Dieser Schritt erstellt ein neues AMI mit einem aussagekräftigen Namen, der es mit der Quell-ID und dem Zeitpunkt der Erstellung verknüpft. Zum Beispiel: „AMIGeneriert durch EC2 Automatisierung am {{global:Date_Time}} von {{}SourceAmiId}“, wobei DATE_TIME und SourceID Automatisierungsvariablen darstellen.

Schritt 5: TerminateInstance (**`aws:changeInstanceState`**-Aktion)

Dieser Schritt bereinigt die Automatisierung durch Beenden der ausgeführten Instance.

Output

Die Automatisierung gibt die neue AMI-ID als Ausgabe zurück.

> **Note**
>
> Standardmäßig erstellt das System eine temporäre Instance in der Standard-VPC (172.30.0.0/16), wenn Automation das `AWS-UpdateLinuxAmi`-Runbook ausführt. Wenn Sie die Standard-VPC gelöscht haben, erhalten Sie den folgenden Fehler:
> `VPC not defined 400`
> Zur Behebung dieses Problems erstellen Sie eine Kopie des `AWS-UpdateLinuxAmi`-Runbooks und geben eine Subnetz-ID an. Weitere Informationen finden Sie unter VPC nicht definiert 400.

So erstellen Sie ein gepatchtes AMImit der Automation (AWS Systems Manager)

1. AWS Systems Manager https://console.aws.amazon.com/systems-manager/Öffnen Sie die Konsole unter.
2. Klicken Sie im Navigationsbereich auf Automation.

3. Wählen Sie Automatisierung ausführen.
4. Wählen Sie in der Liste Automation-Dokument **`AWS-UpdateLinuxAmi`**.
5. Überprüfen Sie im Abschnitt Document details (Dokumentdetails), ob Document version (Dokumentversion) auf Default version at runtime (Standardversion bei Laufzeit) gesetzt ist.
6. Wählen Sie Weiter aus.
7. Wählen Sie im Abschnitt Execution mode (Ausführungsmodus) die Option Simple Execution (Einfache Ausführung) aus.
8. Geben Sie im Abschnitt Input parameters (Eingabeparameter) die Informationen ein, die Sie im Abschnitt Before You Begin (Bevor Sie beginnen) erfasst haben.
9. Wählen Sie Ausführen. Die Konsole zeigt den Status der Automation-Ausführung an.

Nach dem Abschluss der Automatisierung starten Sie eine Test-Instance über das aktualisierte AMI, um die Änderungen zu überprüfen.

> **Note**
>
> Falls ein Schritt in der Automatisierung fehlschlägt, werden die Informationen zu dem Fehler auf der Seite Automation Executions (Automation-Ausführungen) aufgelistet. Die Automatisierung ist so konzipiert, dass sie die temporäre Instance nach erfolgreichem Abschluss aller Aufgaben beendet. Wenn ein Schritt fehlschlägt, beendet das System die Instance möglicherweise nicht. Wenn also ein Schritt fehlschlägt, beenden Sie die temporäre Instance manuell.

Aktualisieren eines Linux AMI (AWS CLI)

Diese exemplarische Vorgehensweise zur AWS Systems Manager Automatisierung zeigt Ihnen, wie Sie das Runbook AWS Command Line Interface (AWS CLI) und das Systems Manager `AWS-UpdateLinuxAmi` Manager-Runbook verwenden, um ein Linux Amazon Machine Image (AMI) automatisch mit den neuesten Versionen der von Ihnen angegebenen Pakete zu patchen. Automation ist ein Tool in AWS Systems Manager. Das `AWS-UpdateLinuxAmi`-Runbook automatisiert auch die Installation zusätzlicher websitespezifischer Pakete und Konfigurationen. Sie können mit dieser Anleitung eine Vielzahl von Linux-Distributionen aktualisieren, einschließlich Ubuntu Server, Red Hat Enterprise Linux (RHEL) oder Amazon Linux AMIs. Eine vollständige Liste der unterstützten Linux-Versionen finden Sie unter Patch Manager-Voraussetzungen.

Das `AWS-UpdateLinuxAmi`-Runbook ermöglicht Ihnen die Automatisierung von Image-Verwaltungsaufgaben ohne Erstellen des Runbooks in JSON oder YAML. Sie können das Runbook `AWS-UpdateLinuxAmi` verwenden, um die folgenden Arten von Aufgaben auszuführen.

- Aktualisieren Sie alle Distributionspakete und jegliche Amazon-Software in einem Amazon Machine Image (AMI) mit Amazon Linux, RHEL oder Ubuntu Server. Dies ist das Runbook-Standardverhalten.
- Installieren Sie AWS Systems Manager SSM Agent auf einem vorhandenen Image, um Systems Manager Manager-Funktionen zu aktivieren, z. B. das Ausführen von Remotebefehlen mithilfe von AWS Systems Manager Run Command oder die Erfassung von Softwareinventar mithilfe von Inventar.
- Installieren Sie zusätzliche Softwarepakete.

Bevor Sie beginnen

Bevor Sie mit der Arbeit mit Runbooks beginnen, konfigurieren Sie Rollen und optional die Funktionen EventBridge für die Automatisierung. Weitere Informationen finden Sie unter Einrichten der Automatisierung. Für diese exemplarische Vorgehensweise müssen Sie außerdem den Namen eines AWS Identity and Access Management (IAM-) Instanzprofils angeben. Weitere Informationen zum Erstellen eines IAM-Instance-Profils finden Sie unter Konfigurieren von erforderlichen Instance-Berechtigungen für Systems Manager.

Das Runbook `AWS-UpdateLinuxAmi` akzeptiert die folgenden Eingabeparameter.

Parameter	Typ	Description
SourceAmiId	String	(Erforderlich) Die Quell-AMI-ID. Mithilfe eines AWS Systems Manager Parameter Store öffentlichen Parameter s können Sie automatisch auf die neueste ID eines Amazon EC2 AMI für Linux verweisen . Weitere Informationen finden Sie unter Query for the latest Amazon Linux AMI IDs using

Parameter	Typ	Description
		AWS Systems ManagerParameter Store.
IamInstanceProfileName	String	(Erforderlich) Der Name der IAM-Instance-Profilrolle, die Sie unter Erforderliche Instance-Berechtigungen für Systems Manager konfigurieren erstellt haben. Die Instance-Profilrolle erteilt der Automation die Berechtigung, auf Ihren Instances Aktionen durchzuführen, wie etwa das Ausführen von Befehlen oder das Starten und Beenden von Services. Das Runbook verwendet nur den Namen der Instance-Profilrolle.

Parameter	Typ	Description
AutomationAssumeRole	String	(Erforderlich) Der Name der IAM-Servicerolle, die Sie in Einrichten der Automatisierung erstellt haben. Mit der Servicerolle (auch als assume-Rolle bezeichnet) gestatten Sie der Automatisierung, Ihre IAM-Rolle zu übernehmen und in Ihrem Auftrag Aktionen auszuführen. Mit der Servicerolle gestatten Sie der Automation beispielsweise beim Ausführen der Aktion `aws:createImage` in einem Runbook, ein neues AMI zu erstellen. Für diesen Parameter muss der vollständige ARN angegeben werden.
TargetAmiName	String	(Optional) Der Name des neuen AMI nach seiner Erstellung. Der Standardname ist eine systemgenerierte Zeichenfolge, die die Quell-AMI-ID sowie Uhrzeit und Datum der Erstellung enthält.
InstanceType	String	(Optional) Der Typ der zu startenden Instance als Arbeitsbereich hosten. Die Instance-Typen sind je nach Region unterschiedlich. Der Standardtyp ist t2.micro.

Parameter	Typ	Description
PreUpdateScript	String	(Optional) Die URL eines Skripts, das ausgeführt werden muss, bevor Updates übernommen werden. Standard („none“) ist die Ausführung keines Skripts.
PostUpdateScript	String	(Optional) Die URL eines Skripts, das ausgeführt werden muss, nachdem Paketupdates angewendet werden. Standard („none“) ist die Ausführung keines Skripts.
IncludePackages	String	(Optional) Aktualisieren Sie nur diese benannten Pakete. Standardmäßig werden alle (\"all\") verfügbaren Updates übernommen.
ExcludePackages	String	(Optional) Namen der Pakete, die bei Updates unter allen Umständen zurückgehalten werden müssen. Standardm äßig wird kein (\"none\") Paket ausgeschlossen.

Automation-Schritte

Das `AWS-UpdateLinuxAmi`-Runbook enthält standardmäßig die folgenden Schritte.

Schritt 1: launchInstance (**`aws:runInstances`**-Aktion)

In diesem Schritt wird eine Instance mit Amazon Elastic Compute Cloud (Amazon EC2) -Benutzerdaten und einer IAM-Instance-Profilrolle gestartet. Userdata installiert je nach

Betriebssystem den entsprechenden SSM-Agent. Durch Installieren von SSM Agent können Sie Systems-Manager-Tools wie Run Command, State Manager und Inventory verwenden.

Schritt 2: Aktualisieren OSSoftware (**`aws:runCommand`**Aktion)

Dieser Schritt führt die folgenden Befehle auf der gestarteten Instance aus:

- Lädt ein Update-Skript von Amazon Simple Storage Service (Amazon S3) herunter.
- Führt ein optionales Pre-Update-Skript aus.
- Aktualisiert Verteilungspakete und Amazon-Software.
- Führt ein optionales Post-Update-Skript aus.

Das Ausführungsprotokoll wird im Ordner /tmp gespeichert, damit es der Benutzer zu einem späteren Zeitpunkt ansehen kann.

Falls Sie eine bestimmte Reihe von Paketen aktualisieren möchten, können Sie die Liste mithilfe des `IncludePackages`-Parameters bereitstellen. Bei der Bereitstellung versucht das System nur diese Pakete und deren abhängige Objekte zu aktualisieren. Es werden keine weiteren Updates vorgenommen. Wenn standardmäßig keine include-Pakete festgelegt sind, aktualisiert das Programm alle verfügbaren Pakete.

Falls Sie eine bestimmte Reihe von Paketen von der Aktualisierung ausschließen möchten, können Sie die Liste mithilfe des `ExcludePackages`-Parameters bereitstellen. Wenn diese Pakete bereitgestellt werden, bleiben sie in ihrer aktuellen Version, unabhängig von anderen festgelegten Optionen. Wenn keine exclude-Pakete festgelegt sind, werden standardmäßig keine Pakete ausgeschlossen.

Schritt 3: StopInstance (**`aws:changeInstanceState`**-Aktion)

Dieser Schritt stoppt die aktualisierte Instance.

Schritt 4: CreateImage (**`aws:createImage`**-Aktion)

Dieser Schritt erstellt ein neues AMI mit einem aussagekräftigen Namen, der es mit der Quell-ID und dem Zeitpunkt der Erstellung verknüpft. Zum Beispiel: „AMI, generiert durch EC2 Automatisierung am {{global:Date_Time}} von {{}SourceAmiId}“, wobei DATE_TIME und SourceID Automatisierungsvariablen darstellen.

Schritt 5: TerminateInstance (**`aws:changeInstanceState`**-Aktion)

Dieser Schritt bereinigt die Automatisierung durch Beenden der ausgeführten Instance.

Output

Die Automatisierung gibt die neue AMI-ID als Ausgabe zurück.

> **Note**
>
> Standardmäßig erstellt das System eine temporäre Instance in der Standard-VPC (172.30.0.0/16), wenn Automation das `AWS-UpdateLinuxAmi`-Runbook ausführt. Wenn Sie die Standard-VPC gelöscht haben, erhalten Sie den folgenden Fehler:
> `VPC not defined 400`
> Zur Behebung dieses Problems erstellen Sie eine Kopie des `AWS-UpdateLinuxAmi`-Runbooks und geben eine Subnetz-ID an. Weitere Informationen finden Sie unter VPC nicht definiert 400.

So erstellen Sie ein gepatchtes AMI mithilfe von Automation

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um das `AWS-UpdateLinuxAmi`-Runbook zu starten. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen.

   ```
   aws ssm start-automation-execution \
       --document-name "AWS-UpdateLinuxAmi" \
       --parameters \
       SourceAmiId=AMI ID, \
       IamInstanceProfileName=IAM instance profile, \
       AutomationAssumeRole='arn:aws:iam::
   {{global:ACCOUNT_ID}}:role/AutomationServiceRole'
   ```

   Der Befehl gibt eine Ausführungs-ID zurück. Kopieren Sie diese ID in die Zwischenablage. Sie werden diese ID zum Anzeigen des Status der Automatisierung verwenden.

   ```
   {
       "AutomationExecutionId": "automation execution ID"
   ```

```
}
```

3. Führen Sie den folgenden Befehl aus AWS CLI, um die Automatisierung mit dem anzuzeigen:

```
aws ssm describe-automation-executions
```

4. Führen Sie den folgenden Befehl aus, um Details über den Automatisierungsprozess anzuzeigen. *`automation execution ID`*Ersetzen Sie es durch Ihre eigenen Informationen.

```
aws ssm get-automation-execution --automation-execution-id automation execution ID
```

Die Aktualisierung kann 30 Minuten oder länger in Anspruch nehmen.

> **Note**
>
> Sie können auch den Status der Automatisierung in der Konsole überwachen. Wählen Sie in der Liste die Automatisierung, die Sie gerade ausgeführt haben, und wählen Sie dann die Registerkarte Steps (Schritte). Diese Registerkarte zeigt Ihnen den Status der Automatisierungsaktionen.

Nach dem Abschluss der Automatisierung starten Sie eine Test-Instance über das aktualisierte AMI, um die Änderungen zu überprüfen.

> **Note**
>
> Falls ein Schritt in der Automatisierung fehlschlägt, werden die Informationen zu dem Fehler auf der Seite Automation Executions (Automation-Ausführungen) aufgelistet. Die Automatisierung ist so konzipiert, dass sie die temporäre Instance nach erfolgreichem Abschluss aller Aufgaben beendet. Wenn ein Schritt fehlschlägt, beendet das System die Instance möglicherweise nicht. Wenn also ein Schritt fehlschlägt, beenden Sie die temporäre Instance manuell.

Aktualisieren eines Windows Server-AMI

Das Runbook `AWS-UpdateWindowsAmi` ermöglicht die Automatisierung von Image-Verwaltungsaufgaben auf Ihrem Amazon Windows Amazon Machine Image (AMI), ohne dass Sie das Runbook in JSON oder YAML erstellen müssen. Dieses Runbook wird unterstützt für Windows

Server 2008 R2 oder höher. Sie können das Runbook `AWS-UpdateWindowsAmi` verwenden, um die folgenden Arten von Aufgaben auszuführen.

- Installieren Sie alle Windows-Updates und aktualisieren Sie die Amazon-Software (Standardverhalten).
- Installieren Sie spezifische Windows-Updates und aktualisieren Sie die Amazon-Software.
- Passen Sie ein AMI mithilfe Ihrer Skripts an.

Bevor Sie beginnen

Bevor Sie mit Runbooks arbeiten, konfigurieren Sie Rollen für Automation, um eine `iam:PassRole`-Richtlinie hinzuzufügen, die auf den ARN des Instance-Profils verweist, dem Sie den Zugriff gewähren möchten. Konfigurieren Sie optional Amazon EventBridge for Automation, ein Tool in AWS Systems Manager. Weitere Informationen finden Sie unter Einrichten der Automatisierung. Für diese exemplarische Vorgehensweise müssen Sie außerdem den Namen eines AWS Identity and Access Management (IAM-) Instance-Profils angeben. Weitere Informationen zum Erstellen eines IAM-Instance-Profils finden Sie unter Konfigurieren von erforderlichen Instance-Berechtigungen für Systems Manager.

> **Note**
>
> Updates für AWS Systems Manager SSM Agent werden in der Regel für verschiedene Regionen zu verschiedenen Zeiten angeboten. Wenn Sie ein AMI anpassen oder aktualisieren, verwenden Sie nur die Quelle, die für die Region AMIs veröffentlicht wurde, in der Sie arbeiten. Auf diese Weise stellen Sie sicher, dass Sie mit dem neuesten SSM Agent für diese Region arbeiten und vermeiden Kompatibilitätsprobleme.

Das Runbook `AWS-UpdateWindowsAmi` akzeptiert die folgenden Eingabeparameter.

Parameter	Typ	Description
SourceAmiId	String	(Erforderlich) Die Quell-AMI -ID. Sie können automatis ch auf die neueste Windows Server-AMI mithilfe eines Systems ManagerParameter

Parameter	Typ	Description
		Store öffentlich-Parameter verweisen. Weitere Informationen finden Sie unter Query for the latest Windows AMI IDs using AWS Systems ManagerParameter Store.
SubnetId	String	(Optional) Das Subnetz, in dem Sie die temporäre Instance starten möchten. Sie müssen einen Wert für diesen Parameter angeben, wenn Sie Ihre Standard-VPC gelöscht haben.
IamInstanceProfileName	String	(Erforderlich) Der Name der IAM-Instance-Profilrolle, die Sie unter Erforderliche Instance-Berechtigungen für Systems Manager konfigurieren erstellt haben. Die Instance-Profilrolle erteilt der Automation die Berechtigung, auf Ihren Instances Aktionen durchzuführen, wie etwa das Ausführen von Befehlen oder das Starten und Beenden von Services. Das Runbook verwendet nur den Namen der Instance-Profilrolle.

Parameter	Typ	Description
AutomationAssumeRole	String	(Erforderlich) Der Name der IAM-Servicerolle, die Sie in Einrichten der Automatisierung erstellt haben. Mit der Servicerolle (auch als assume-Rolle bezeichnet) gestatten Sie der Automatisierung, Ihre IAM-Rolle zu übernehmen und in Ihrem Auftrag Aktionen auszuführen. Mit der Servicerolle gestatten Sie der Automation beispielsweise beim Ausführen der Aktion `aws:createImage` in einem Runbook, ein neues AMI zu erstellen. Für diesen Parameter muss der vollständige ARN angegeben werden.
TargetAmiName	String	(Optional) Der Name des neuen AMI nach seiner Erstellung. Der Standardname ist eine systemgenerierte Zeichenfolge, die die Quell-AMI-ID sowie Uhrzeit und Datum der Erstellung enthält.
InstanceType	String	(Optional) Der Typ der zu startenden Instance als Arbeitsbereich hosten. Die Instance-Typen sind je nach Region unterschiedlich. Der Standardtyp ist t2.medium.

Parameter	Typ	Description
PreUpdateScript	String	(Optional) Ein Skript, das ausgeführt werden muss, bevor das AMI aktualisiert wird. Geben Sie ein Skript im Runbook oder zur Laufzeit als Parameter an.
PostUpdateScript	String	(Optional) Ein Skript, das ausgeführt werden muss, nachdem das AMI aktualisiert wird. Geben Sie ein Skript im Runbook oder zur Laufzeit als Parameter an.
IncludeKbs	String	(Optional) Geben Sie einen oder mehrere Microsoft Knowledge Base-Artikel (KB) IDs an, die aufgenommen werden sollen. Sie können mehrere IDs mithilfe von kommagetrennten Werten installieren. Gültige Formate: KB9876543 oder 9876543.
ExcludeKbs	String	(Optional) Geben Sie einen oder mehrere Microsoft Knowledge Base-Artikel (KB) IDs an, die ausgeschl ossen werden sollen. Sie können mehrere IDs mithilfe von kommagetrennten Werten ausschließen. Gültige Formate: KB9876543 oder 9876543.

Parameter	Typ	Description
Kategorien	String	(Optional) Geben Sie mindestens eine Updatekategorie an. Sie können Kategorien anhand kommaseparierter Werte filtern. Optionen: Wichtiges Update, Sicherheitsupdate, Definitionsupdate, Update-Rollup, Service Pack, Tool, Update oder Treiber. Zu den gültigen Formaten gehört ein einzelner Eintrag. Beispiel: Wichtiges Update. Sie können auch eine kommaseparierte Liste angeben: Wichtiges Update,Sicherheitsupdate,Definitionsupdate.
SeverityLevels	String	(Optional) Geben Sie mindestens eine MSRC-Ebene an, die einem Update zugeordnet ist. Sie können Dringlichkeitsstufen anhand kommaseparierter Werte filtern. Optionen: Kritisch, Wichtig, Niedrige, Mittel oder Nicht angegeben. Zu den gültigen Formaten gehört ein einzelner Eintrag. Beispiel: Wichtig. Sie können auch eine kommaseparierte Liste angeben: Kritisch,Wichtig,Niedrig.

Automation-Schritte

Das `AWS-UpdateWindowsAmi`-Runbook enthält standardmäßig die folgenden Schritte.

Schritt 1: launchInstance (**`aws:runInstances`**-Aktion)

Dieser Schritt startet eine Instance mit einer IAM-Instance-Profilrolle über das angegebene `SourceAmiID`.

Schritt 2: runPreUpdate Skript (Aktion) **`aws:runCommand`**

Mit diesem Schritt können Sie ein Skript als Zeichenfolge angeben, das ausgeführt wird, bevor Updates installiert werden.

Schritt 3: EC2 Config aktualisieren (**`aws:runCommand`**Aktion)

In diesem Schritt wird das `AWS-InstallPowerShellModule` Runbook verwendet, um ein AWS öffentliches PowerShell Modul herunterzuladen. Systems Manager überprüft die Integrität des Modul mithilfe eines SHA-256-Hash. Systems Manager überprüft dann das Betriebssystem, um festzustellen, ob EC2 Config oder EC2 Launch aktualisiert werden soll. EC2Config läuft auf Windows Server 2008 R2 bis Windows Server 2012 R2. EC2Launch läuft auf Windows Server 2016.

Schritt 4: Update SSMAgent (**`aws:runCommand`**Aktion)

Dieser Schritt aktualisiert SSM Agent mithilfe des `AWS-UpdateSSMAgent`-Runbooks.

Schritt 5: Update AWSPVDriver (**`aws:runCommand`**Aktion)

In diesem Schritt werden die AWS PV-Treiber mithilfe des `AWS-ConfigureAWSPackage` Runbooks aktualisiert.

Schritt 6: updateAwsEna NetworkDriver (**`aws:runCommand`**Aktion)

In diesem Schritt werden die AWS ENA-Netzwerktreiber mithilfe des `AWS-ConfigureAWSPackage` Runbooks aktualisiert.

Schritt 7: installWindowsUpdates (**`aws:runCommand`**Aktion)

Dieser Schritt installiert Windows-Updates mithilfe des `AWS-InstallWindowsUpdates`-Runbooks. Standardmäßig sucht und installiert Systems Manager alle fehlenden Updates. Sie können das Standardverhalten ändern, indem Sie einen der folgenden Parameter festlegen: `IncludeKbs`, `ExcludeKbs`, `Categories` oder `SeverityLevels`.

Schritt 8: runPostUpdate Script (**`aws:runCommand`**Aktion)

Mit diesem Schritt können Sie ein Skript als Zeichenfolge angeben, das ausgeführt wird, nachdem Updates installiert wurden.

Schritt 9: runSysprepGeneralize (**`aws:runCommand`**Aktion)

In diesem Schritt wird das `AWS-InstallPowerShellModule` Runbook verwendet, um ein AWS öffentliches PowerShell Modul herunterzuladen. Systems Manager überprüft die Integrität des Modul mithilfe eines SHA-256-Hash. Systems Manager führt dann Sysprep mit AWS unterstützten Methoden für EC2 Launch (Windows Server 2016) oder EC2 Config (Windows Server 2008 R2 bis 2012 R2) aus.

Schritt 10: stopInstance (**`aws:changeInstanceState`**-Aktion)

Dieser Schritt stoppt die aktualisierte Instance.

Schritt 11: createImage (**`aws:createImage`**-Aktion)

Dieser Schritt erstellt ein neues AMI mit einem aussagekräftigen Namen, der es mit der Quell-ID und dem Zeitpunkt der Erstellung verknüpft. Zum Beispiel: „AMI, generiert durch EC2 Automatisierung am {{global:Date_Time}} von {{}SourceAmiId}“, wobei DATE_TIME und SourceID Automatisierungsvariablen darstellen.

TerminateInstance **`aws:changeInstanceState`**Schritt 12: (Aktion)

Dieser Schritt bereinigt die Automatisierung durch Beenden der ausgeführten Instance.

Output

In diesem Abschnitt können Sie die Ausgabe verschiedener Schritte oder Werte eines beliebigen Parameters als die Automation-Ausgabe bestimmen. Standardmäßig ist die Ausgabe die ID des aktualisierten Windows-AMI, das von der Automatisierung erstellt wurde.

> **Note**
>
> Standardmäßig verwendet das System die Standard-VPC (172.30.0.0/16), wenn Automation das `AWS-UpdateWindowsAmi`-Runbbok ausführt und eine temporäre Instance erstellt. Wenn Sie die Standard-VPC gelöscht haben, erhalten Sie den folgenden Fehler:
> VPC nicht definiert 400

Zur Behebung dieses Problems erstellen Sie eine Kopie des `AWS-UpdateWindowsAmi`-Runbooks und geben eine Subnetz-ID an. Weitere Informationen finden Sie unter VPC nicht definiert 400.

So erstellen Sie ein gepatchtes Windows-AMI mit der Automation

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um das `AWS-UpdateWindowsAmi`-Runbook zu starten. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen. Der folgende Beispielbefehl verwendet eine aktuelle Version von Amazon EC2 AMI, um die Anzahl der Patches zu minimieren, die angewendet werden müssen. Wenn Sie diesen Befehl mehrmals ausführen, müssen Sie einen eindeutigen Wert für `targetAMIname` angeben. AMI-Namen müssen einzigartig sein.

   ```
   aws ssm start-automation-execution \
       --document-name="AWS-UpdateWindowsAmi" \
       --parameters SourceAmiId='AMI ID',IamInstanceProfileName='IAM
    instance profile',AutomationAssumeRole='arn:aws:iam::
   {{global:ACCOUNT_ID}}:role/AutomationServiceRole'
   ```

   Der Befehl gibt eine Ausführungs-ID zurück. Kopieren Sie diese ID in die Zwischenablage. Sie werden diese ID zum Anzeigen des Status der Automatisierung verwenden.

   ```
   {
       "AutomationExecutionId": "automation execution ID"
   }
   ```

3. Führen Sie den folgenden Befehl aus AWS CLI, um die Automatisierung mithilfe von anzuzeigen:

   ```
   aws ssm describe-automation-executions
   ```

4. Führen Sie den folgenden Befehl aus, um Details über den Automatisierungsprozess anzuzeigen.

```
aws ssm get-automation-execution
    --automation-execution-id automation execution ID
```

> **Note**
>
> Abhängig von der Anzahl der angewendeten Patches kann der Windows-Patch-Vorgang in dieser Beispielautomatisierung 30 Minuten oder länger in Anspruch nehmen.

Aktualisieren Sie ein Golden AMI mithilfe von Automation, AWS Lambda, und Parameter Store

Im folgenden Beispiel wird das Modell verwendet, bei dem eine Organisation ihre eigenen, proprietären Systeme verwaltet und regelmäßig patcht, AMIs anstatt auf Amazon Elastic Compute Cloud (Amazon EC2) aufzubauenAMIs.

Das folgende Verfahren zeigt, wie Betriebssystem-Patches (OS) automatisch auf ein System angewendet werdenAMI, das bereits als die aktuellsten up-to-date oder aktuellsten giltAMI. In diesem Beispiel `SourceAmiId` wird der Standardwert des Parameters durch einen AWS Systems Manager Parameter Store Parameter definiert, der aufgerufen wird`latestAmi`. Der Wert von `latestAmi` wird durch eine AWS Lambda Funktion aktualisiert, die am Ende der Automatisierung aufgerufen wird. Durch diesen Automatisierungsprozess werden der Zeit- und Arbeitsaufwand für das Patchen minimiert, da AMIs das Patchen immer auf die meisten angewendet wird. up-to-date AMI Parameter Storeund Automatisierung sind Werkzeuge von. AWS Systems Manager

Bevor Sie beginnen

Konfigurieren Sie Automatisierungsrollen und optional Amazon EventBridge for Automation. Weitere Informationen finden Sie unter Einrichten der Automatisierung.

Inhalt

- Aufgabe 1: Erstellen eines Parameters im Systems Manager-Parameter Store
- Aufgabe 2: Erstellen einer IAM-Rolle für AWS Lambda
- Aufgabe 3: Erstellen einer AWS Lambda -Funktion
- Aufgabe 4: Erstellen eines Runbooks und Patchen des AMI

Aufgabe 1: Erstellen eines Parameters im Systems Manager-Parameter Store

Erstellen Sie einen Zeichenfolgen-Parameter in Parameter Store, der die folgenden Informationen verwendet:

- Name: `latestAmi`.
- Value (Wert): Eine AMI-ID. Zum Beispiel: `ami-188d6e0e`.

Informationen zur Erstellung eines Parameter Store-Zeichenfolgenparameters finden Sie unter Parameter Store-Parameter im Systems Manager erstellen.

Aufgabe 2: Erstellen einer IAM-Rolle für AWS Lambda

Verwenden Sie das folgende Verfahren, um eine IAM-Dienstrolle für AWS Lambda zu erstellen. Diese Richtlinien erteilen Lambda die Berechtigung zum Aktualisieren des Werts des `latestAmi`-Parameters mithilfe einer Lambda-Funktion und von Systems Manager.

So erstellen Sie eine IAM-Service-Rolle für Lambda

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im Navigationsbereich Richtlinien und dann Richtlinie erstellen.
3. Wählen Sie den Tab JSON.
4. Ersetzen Sie den Standardinhalt durch die folgende Richtlinie. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:CreateLogGroup",
            "Resource": "arn:aws:logs:us-east-1:111122223333:*"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:/aws/
lambda/function name:*"
            ]
        }
    ]
}
```

5. Wählen Sie Weiter: Tags aus.
6. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Richtlinie zu organisieren, zu verfolgen oder zu steuern.
7. Wählen Sie Weiter: Prüfen aus.
8. Geben Sie auf der Seite Richtlinie prüfen im Feld Name einen Namen für die Inline-Richtlinie ein, z. B. **amiLambda**.
9. Wählen Sie Richtlinie erstellen aus.
10. Wiederholen Sie die Schritte 2 und 3.
11. Fügen Sie die folgende Richtlinie ein. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

    JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ssm:PutParameter",
            "Resource": "arn:aws:ssm:us-
east-1:111122223333:parameter/latestAmi"
        },
        {
            "Effect": "Allow",
            "Action": "ssm:DescribeParameters",
            "Resource": "*"
        }
    ]
```

```
}
```

12. Wählen Sie Weiter: Tags aus.
13. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Richtlinie zu organisieren, zu verfolgen oder zu steuern.
14. Wählen Sie Weiter: Prüfen aus.
15. Geben Sie auf der Seite Richtlinie prüfen im Feld Name einen Namen für die Inline-Richtlinie ein, z. B. **`amiParameter`**.
16. Wählen Sie Richtlinie erstellen aus.
17. Wählen Sie im Navigationsbereich Rollen und dann Rolle erstellen.
18. Wählen Sie direkt unter Anwendungsfall die Option Lambda und dann Weiter aus.
19. Suchen Sie auf der Seite Berechtigungsrichtlinien anfügen im Feld Suche die beiden Richtlinien, die Sie zuvor erstellt haben.
20. Aktivieren Sie das Kontrollkästchen neben den Richtlinien und wählen Sie anschließend Weiter aus.
21. Geben Sie unter Role name (Rollenname) einen Namen für Ihre neue Rolle, wie z. B. **`lambda-ssm-role`**, oder einen anderen von Ihnen bevorzugten Namen ein.

    > **Note**
    >
    > Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung nicht geändert werden.

22. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Rolle zu organisieren, nachzuverfolgen oder zu steuern, und wählen Sie dann Rolle erstellen aus.

Aufgabe 3: Erstellen einer AWS Lambda -Funktion

Führen Sie die folgenden Schritte zum Erstellen einer Lambda-Funktion aus, die den Wert des `latestAmi`-Parameters automatisch aktualisiert.

So erstellen Sie eine Lambda-Funktion

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.

2. Wählen Sie Funktion erstellen.
3. Wählen Sie auf der Seite Create function die Option Author from scratch.
4. Geben Sie für Function name (Funktionsname) **`Automation-UpdateSsmParam`** ein.
5. Wählen Sie für Runtime Python 3.11.
6. Wählen Sie unter Architektur den Computerprozessortyp aus, den Lambda zum Ausführen der Funktion verwenden soll, x86_64 oder arm64,
7. Erweitern Sie im Abschnitt Berechtigungen die Option Standardausführungsrolle ändern.
8. Wählen Sie Use an existing role (Vorhandene Rolle verwenden) aus und wählen Sie dann die Servicerolle für Lambda aus, die Sie in Aufgabe 2 erstellt haben.
9. Wählen Sie Funktion erstellen.
10. Löschen Sie im Bereich Code-Quelle in der Registerkarte lambda_function den vorab ausgefüllten Code im Feld und fügen Sie das folgende Codebeispiel ein.

```
from __future__ import print_function

import json
import boto3

print('Loading function')


#Updates an SSM parameter
#Expects parameterName, parameterValue
def lambda_handler(event, context):
    print("Received event: " + json.dumps(event, indent=2))

    # get SSM client
    client = boto3.client('ssm')

    #confirm  parameter exists before updating it
    response = client.describe_parameters(
       Filters=[
          {
           'Key': 'Name',
           'Values': [ event['parameterName'] ]
          },
        ]
    )
```

```
    if not response['Parameters']:
        print('No such parameter')
        return 'SSM parameter not found.'

    #if parameter has a Description field, update it PLUS the Value
    if 'Description' in response['Parameters'][0]:
        description = response['Parameters'][0]['Description']

        response = client.put_parameter(
          Name=event['parameterName'],
          Value=event['parameterValue'],
          Description=description,
          Type='String',
          Overwrite=True
        )

    #otherwise just update Value
    else:
        response = client.put_parameter(
          Name=event['parameterName'],
          Value=event['parameterValue'],
          Type='String',
          Overwrite=True
        )

    responseString = 'Updated parameter %s with value %s.' %
 (event['parameterName'], event['parameterValue'])

    return responseString
```

11. Klicken Sie auf Datei, Speichern.
12. Um die Lambda-Funktion zu testen, wählen Sie im Menü Test die Option Testereignis konfigurieren aus.
13. Geben Sie für Event name (Ereignisname) einen Namen für das Testereignis ein, z. B. **`MyTestEvent`**.
14. Ersetzen Sie den vorhandenen Text durch folgendes JSON-Objekt. *`AMI ID`*Ersetzen Sie es durch Ihre eigenen Informationen, um Ihren `latestAmi` Parameterwert festzulegen.

```
{
   "parameterName":"latestAmi",
   "parameterValue":"AMI ID"
```

```
}
```

15. Wählen Sie Speichern.
16. Wählen Sie Test aus, um die Funktion zu testen. Auf der Registerkarte Ausführungsergebnis sollte der Status als Erfolgreich gemeldet werden, zusammen mit anderen Details zur Aktualisierung.

Aufgabe 4: Erstellen eines Runbooks und Patchen des AMI

Verwenden Sie die folgende Vorgehensweise zum Erstellen und Ausführen eines Runbooks, das das von Ihnen angegebene AMI für den latestAmi-Parameter patcht. Nach dem Abschluss der Automatisierung wird der Wert latestAmi mit der ID des neu gepatchten AMI aktualisiert. Bei nachfolgenden Automatisierungen verwenden Sie das AMI, das in der vorherigen Ausführung erstellt wurde.

Erstellen und Ausführen des Runbooks

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie für Dokument erstellen die Option Automatisierung aus.
4. Geben Sie unter Name **`UpdateMyLatestWindowsAmi`** ein.
5. Wählen Sie die Registerkarte Editor und wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie bei Aufforderung OK aus.
7. Ersetzen Sie im Feld Dokument-Editor den Standardinhalt durch den folgenden Inhalt des YAML-Beispiel-Runbooks.

```
---
description: Systems Manager Automation Demo - Patch AMI and Update ASG
schemaVersion: '0.3'
assumeRole: '{{ AutomationAssumeRole }}'
parameters:
  AutomationAssumeRole:
    type: String
    description: '(Required) The ARN of the role that allows Automation to perform
 the actions on your behalf. If no role is specified, Systems Manager Automation
 uses your IAM permissions to execute this document.'
    default: ''
```

```
  SourceAMI:
    type: String
    description: The ID of the AMI you want to patch.
    default: '{{ ssm:latestAmi }}'
  SubnetId:
    type: String
    description: The ID of the subnet where the instance from the SourceAMI
 parameter is launched.
  SecurityGroupIds:
    type: StringList
    description: The IDs of the security groups to associate with the instance
 that's launched from the SourceAMI parameter.
  NewAMI:
    type: String
    description: The name of of newly patched AMI.
    default: 'patchedAMI-{{global:DATE_TIME}}'
  InstanceProfile:
    type: String
    description: The name of the IAM instance profile you want the source instance
 to use.
  SnapshotId:
    type: String
    description: (Optional) The snapshot ID to use to retrieve a patch baseline
 snapshot.
    default: ''
  RebootOption:
    type: String
    description: '(Optional) Reboot behavior after a patch Install operation. If
 you choose NoReboot and patches are installed, the instance is marked as non-
compliant until a subsequent reboot and scan.'
    allowedValues:
      - NoReboot
      - RebootIfNeeded
    default: RebootIfNeeded
  Operation:
    type: String
    description: (Optional) The update or configuration to perform on the instance.
 The system checks if patches specified in the patch baseline are installed on the
 instance. The install operation installs patches missing from the baseline.
    allowedValues:
      - Install
      - Scan
    default: Install
mainSteps:
```

```
  - name: startInstances
    action: 'aws:runInstances'
    timeoutSeconds: 1200
    maxAttempts: 1
    onFailure: Abort
    inputs:
      ImageId: '{{ SourceAMI }}'
      InstanceType: m5.large
      MinInstanceCount: 1
      MaxInstanceCount: 1
      IamInstanceProfileName: '{{ InstanceProfile }}'
      SubnetId: '{{ SubnetId }}'
      SecurityGroupIds: '{{ SecurityGroupIds }}'
  - name: verifyInstanceManaged
    action: 'aws:waitForAwsResourceProperty'
    timeoutSeconds: 600
    inputs:
      Service: ssm
      Api: DescribeInstanceInformation
      InstanceInformationFilterList:
        - key: InstanceIds
          valueSet:
            - '{{ startInstances.InstanceIds }}'
      PropertySelector: '$.InstanceInformationList[0].PingStatus'
      DesiredValues:
        - Online
    onFailure: 'step:terminateInstance'
  - name: installPatches
    action: 'aws:runCommand'
    timeoutSeconds: 7200
    onFailure: Abort
    inputs:
      DocumentName: AWS-RunPatchBaseline
      Parameters:
        SnapshotId: '{{SnapshotId}}'
        RebootOption: '{{RebootOption}}'
        Operation: '{{Operation}}'
      InstanceIds:
        - '{{ startInstances.InstanceIds }}'
  - name: stopInstance
    action: 'aws:changeInstanceState'
    maxAttempts: 1
    onFailure: Continue
    inputs:
```

```
      InstanceIds:
        - '{{ startInstances.InstanceIds }}'
      DesiredState: stopped
  - name: createImage
    action: 'aws:createImage'
    maxAttempts: 1
    onFailure: Continue
    inputs:
      InstanceId: '{{ startInstances.InstanceIds }}'
      ImageName: '{{ NewAMI }}'
      NoReboot: false
      ImageDescription: Patched AMI created by Automation
  - name: terminateInstance
    action: 'aws:changeInstanceState'
    maxAttempts: 1
    onFailure: Continue
    inputs:
      InstanceIds:
        - '{{ startInstances.InstanceIds }}'
      DesiredState: terminated
  - name: updateSsmParam
    action: aws:invokeLambdaFunction
    timeoutSeconds: 1200
    maxAttempts: 1
    onFailure: Abort
    inputs:
        FunctionName: Automation-UpdateSsmParam
        Payload: '{"parameterName":"latestAmi",
 "parameterValue":"{{createImage.ImageId}}"}'
outputs:
- createImage.ImageId
```

8. Wählen Sie Create automation (Automation erstellen).
9. Wählen Sie im Navigationsbereich Automatisierung und Automatisierung ausführen aus.
10. Wählen Sie auf der Seite Choose document (Dokument wählen), die Registerkarte Owned by me (In meinem Besitz).
11. Suchen Sie nach dem UpdateMyLatestWindowsAmiRunbook und wählen Sie die Schaltfläche auf der UpdateMyLatestWindowsAmiKarte aus.
12. Wählen Sie Weiter aus.
13. Wählen Sie Simple execution (Einfache Ausführung) aus.
14. Geben Sie Werte für die Eingabeparameter an.

15. Wählen Sie Ausführen.
16. Wählen Sie nach dem Abschluss der Automatisierung Parameter Store im Navigationsbereich und bestätigen Sie, dass der neue Wert für `latestAmi`-Treffer, die von der Automatisierung zurückgegeben werden. Sie können auch überprüfen, ob die neue AMI ID mit der Automation-Ausgabe im AMIsBereich der EC2 Amazon-Konsole übereinstimmt.

Aktualisierung von AMIs mithilfe von Automation und Jenkins

Wenn Ihr Unternehmen Jenkins Software in einer CI/CD Pipeline verwendet, können Sie Automatisierung als Post-Build-Schritt hinzufügen, um Anwendungsversionen in Amazon Machine Images () vorzuinstallieren. AMIs Automation ist ein Tool in AWS Systems Manager. Sie können auch die Jenkins-Funktion zum Planen verwenden, um Automation aufzurufen, und Ihr eigenes Patchingintervall für das Betriebssystem erstellen.

Das folgende Beispiel zeigt, wie Automation von einem Jenkins Server aus aufgerufen wird, der entweder vor Ort oder in Amazon Elastic Compute Cloud (Amazon EC2) läuft. Für die Authentifizierung verwendet der Jenkins Server AWS Anmeldeinformationen, die auf einer IAM-Richtlinie basieren, die Sie im Beispiel erstellen und an Ihr Instance-Profil anhängen.

> **Note**
>
> Befolgen Sie beim Konfigurieren Ihrer Instance die bewährten Methoden von Jenkins für die Sicherheit.

Bevor Sie beginnen

Schließen Sie die folgenden Aufgaben ab, bevor Sie Automation mit Jenkins konfigurieren:

- Schließen Sie das Aktualisieren Sie ein Golden AMI mithilfe von Automation, AWS Lambda, und Parameter Store-Beispiel ab. Im folgenden Beispiel wird das in diesem Beispiel erstellte UpdateMyLatestWindowsAmiRunbook verwendet.
- Konfigurieren Sie IAM-Rollen für Automation. Systems Manager benötigt eine Instance-Profilrolle und einen Servicerollen-ARN zur Verarbeitung von Automatisierungen. Weitere Informationen finden Sie unter Einrichten der Automatisierung.

So erstellen Sie eine IAM-Richtlinie für den Jenkins-Server

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im Navigationsbereich Richtlinien und dann Richtlinie erstellen.
3. Wählen Sie den Tab JSON.
4. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ssm:StartAutomationExecution",
            "Resource": [
                "arn:aws:ssm:us-
east-1:111122223333:document/UpdateMyLatestWindowsAmi",
                "arn:aws:ssm:us-east-1:111122223333:automation-execution/*"
            ]
        }
    ]
}
```

5. Wählen Sie Richtlinie prüfen.
6. Geben Sie auf der Seite Richtlinie prüfen im Feld Name einen Namen für die Inline-Richtlinie ein, z. B. **`JenkinsPolicy`**.
7. Wählen Sie Richtlinie erstellen aus.
8. Wählen Sie im Navigationsbereich Rollen.
9. Wählen Sie das Instance-Profil aus, das Ihrem Jenkins-Server angefügt ist.
10. Wählen Sie auf der Registerkarte Berechtigungen die Option Berechtigungen hinzufügen, Richtlinien anfügen.
11. Geben Sie im Abschnitt Andere Berechtigungsrichtlinien den Namen der Richtlinie ein, die Sie in den vorherigen Schritten erstellt haben. Beispiel, JenkinsPolicy.
12. Aktivieren Sie das Kontrollkästchen neben Ihrer Richtlinie, und wählen Sie Richtlinien anfügen aus.

Gehen Sie wie folgt vor, um das AWS CLI auf Ihrem Jenkins Server zu konfigurieren.

So konfigurieren Sie den Jenkins-Server für Automation

1. Verbinden Sie sich mit Ihrem Jenkins-Server auf Port 8080 über Ihren bevorzugten Browser, um auf die Verwaltungsschnittstelle zuzugreifen.
2. Geben Sie das Passwort ein, welches Sie unter `/var/lib/jenkins/secrets/initialAdminPassword` finden. Um Ihr Kennwort anzuzeigen, führen Sie den folgenden Befehl aus.

   ```
   sudo cat /var/lib/jenkins/secrets/initialAdminPassword
   ```
3. Das Jenkins-Installationsskript leitet Sie zur Seite Anpassen von Jenkins weiter. Wählen Sie Installieren von empfohlenen Plugins.
4. Nach abgeschlossener Installation wählen Sie Administrator-Zugangsdaten, wählen Sie dann Zugansdaten speichern und klicken Sie dann auf Verwendung von Jenkins beginnen aus.
5. Wählen Sie im linken Navigationsbereich Jenkins verwalten und dann Plugins verwalten.
6. Wählen Sie die Registerkarte Available (Verfügbar) und geben Sie dann **`Amazon EC2 plugin`** ein.
7. Aktivieren Sie das Kontrollkästchen für **`Amazon EC2 plugin`** und klicken Sie dann auf Installation ohne Neustart.
8. Nach abgeschlossener Installation wählen Sie Zurück zur oberen Seite.
9. Wählen Sie Jenkins verwalten und anschließend Knoten und Clouds verwalten aus.
10. Wählen Sie im Abschnitt Clouds konfigurieren die Option Neue Cloud hinzufügen und dann Amazon aus EC2.
11. Geben Sie Ihre Daten in die verbleibenden Felder ein. Stellen Sie sicher, dass Sie die Option EC2 Instanzprofil verwenden, um Anmeldeinformationen zu erhalten ausgewählt haben.

Führen Sie die folgenden Schritte zum Konfigurieren Ihres Jenkins-Projekts aus, um Automation zu konfigurieren.

So konfigurieren Sie Ihren Jenkins-Server zum Aufrufen von Automation

1. Öffnen Sie die Jenkins-Konsole in einem Webbrowser.
2. Wählen Sie das Projekt, das Sie mit Automation konfigurieren möchten, und wählen Sie dann Configure (Konfigurieren).

3. Wählen Sie auf der Registerkarte Build Add Build Step (Build-Schritt hinzufügen) aus.
4. Wählen Sie je nach Betriebssystem Execute shell (Shell ausführen) oder Execute Windows batch command (Windows-Batchbefehl ausführen).
5. Führen Sie im Feld Befehl einen AWS CLI Befehl wie den folgenden aus. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

```
aws ssm start-automation-execution \
        --document-name runbook name \
        --region AWS-Region of your source AMI \
        --parameters runbook parameters
```

Der folgende Beispielbefehl verwendet das UpdateMyLatestWindowsAmiRunbook und den Systems Manager Manager-Parameter, die in `latestAmi` Aktualisieren Sie ein Golden AMI mithilfe von Automation, AWS Lambda, und Parameter Store erstellt wurden.

```
aws ssm start-automation-execution \
        --document-name UpdateMyLatestWindowsAmi \
        --parameters \
            "sourceAMIid='{{ssm:latestAmi}}'"
        --region region
```

In Jenkins sieht der Befehl wie im Beispiel im folgenden Screenshot aus.

6. Wählen Sie im Jenkins-Projekt Build Now aus. Jenkins gibt etwa die folgende Ausgabe zurück.

Aktualisieren von AMIs für Auto-Scaling-Gruppen

Im folgenden Beispiel wird eine Auto-Scaling-Gruppe mit einem neu gepatchten AMI aktualisiert. Dieser Ansatz gewährleistet, dass neue Images automatisch den verschiedenen Computing-Umgebungen zur Verfügung gestellt werden, die Auto–Scaling-Gruppen verwenden.

Der letzte Schritt der Automatisierung in diesem Beispiel verwendet eine Python-Funktion, um eine neue Startvorlage zu erstellen, die das neu gepatchte AMI verwendet. Anschließend wird die Auto-Scaling-Gruppe aktualisiert, um die neue Startvorlage zu verwenden. In diesem Auto–Scaling-Szenariotyp können Benutzer vorhandene Instances in der Auto–Scaling-Gruppe beenden, um den Start einer neuen Instance zu erzwingen, die das neue Image verwendet. Andernfalls konnten Benutzer warten und das Skalieren der Ereignisse nach oben oder unten zulassen, um auf natürliche Weise neuere Instances zu starten.

Bevor Sie beginnen

Bevor Sie mit diesem Beispiel beginnen, führen Sie die folgenden Aufgaben aus.

- Konfigurieren Sie IAM-Rollen für Automation, ein Tool in AWS Systems Manager. Systems Manager benötigt eine Instance-Profilrolle und einen Servicerollen-ARN zur Verarbeitung von Automatisierungen. Weitere Informationen finden Sie unter Einrichten der Automatisierung.

Erstellen Sie das Patch AMIAnd UpdateASG-Runbook

Gehen Sie wie folgt vor, um das Patch AMIAnd UpdateAsg-Runbook zu erstellen, das die Patches durchführt, die AMI Sie für den SourceAMI-Parameter angeben. Das Runbook aktualisiert auch eine Auto-Scaling-Gruppe, um das neueste, gepatchte AMI zu verwenden.

Erstellen und Ausführen des Runbooks

1. Öffnen Sie die Konsole unter. AWS Systems Manager https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie Automation im Dropdown-Menü Erstellen eines Dokuments.
4. Geben Sie im Feld Name **`PatchAMIAndUpdateASG`** ein.
5. Wählen Sie die Registerkarte Editor und wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie OK aus, wenn Sie dazu aufgefordert werden, und löschen Sie den Inhalt im Feld Document editor (Dokumenteditor).
7. Fügen Sie im Feld Document editor (Dokumenteditor) den folgenden Inhalt des YAML-Beispiel-Runbooks ein.

```
---
description: Systems Manager Automation Demo - Patch AMI and Update ASG
schemaVersion: '0.3'
assumeRole: '{{ AutomationAssumeRole }}'
parameters:
  AutomationAssumeRole:
    type: String
    description: '(Required) The ARN of the role that allows Automation to perform
 the actions on your behalf. If no role is specified, Systems Manager Automation
 uses your IAM permissions to execute this document.'
    default: ''
  SourceAMI:
    type: String
    description: '(Required) The ID of the AMI you want to patch.'
  SubnetId:
    type: String
    description: '(Required) The ID of the subnet where the instance from the
 SourceAMI parameter is launched.'
  SecurityGroupIds:
    type: StringList
    description: '(Required) The IDs of the security groups to associate with the
 instance launched from the SourceAMI parameter.'
  NewAMI:
    type: String
    description: '(Optional) The name of of newly patched AMI.'
    default: 'patchedAMI-{{global:DATE_TIME}}'
  TargetASG:
```

```
    type: String
    description: '(Required) The name of the Auto Scaling group you want to
 update.'
  InstanceProfile:
    type: String
    description: '(Required) The name of the IAM instance profile you want the
 source instance to use.'
  SnapshotId:
    type: String
    description: (Optional) The snapshot ID to use to retrieve a patch baseline
 snapshot.
    default: ''
  RebootOption:
    type: String
    description: '(Optional) Reboot behavior after a patch Install operation. If
 you choose NoReboot and patches are installed, the instance is marked as non-
compliant until a subsequent reboot and scan.'
    allowedValues:
      - NoReboot
      - RebootIfNeeded
    default: RebootIfNeeded
  Operation:
    type: String
    description: (Optional) The update or configuration to perform on the instance.
 The system checks if patches specified in the patch baseline are installed on the
 instance. The install operation installs patches missing from the baseline.
    allowedValues:
      - Install
      - Scan
    default: Install
mainSteps:
  - name: startInstances
    action: 'aws:runInstances'
    timeoutSeconds: 1200
    maxAttempts: 1
    onFailure: Abort
    inputs:
      ImageId: '{{ SourceAMI }}'
      InstanceType: m5.large
      MinInstanceCount: 1
      MaxInstanceCount: 1
      IamInstanceProfileName: '{{ InstanceProfile }}'
      SubnetId: '{{ SubnetId }}'
      SecurityGroupIds: '{{ SecurityGroupIds }}'
```

```
  - name: verifyInstanceManaged
    action: 'aws:waitForAwsResourceProperty'
    timeoutSeconds: 600
    inputs:
      Service: ssm
      Api: DescribeInstanceInformation
      InstanceInformationFilterList:
        - key: InstanceIds
          valueSet:
            - '{{ startInstances.InstanceIds }}'
      PropertySelector: '$.InstanceInformationList[0].PingStatus'
      DesiredValues:
        - Online
    onFailure: 'step:terminateInstance'
  - name: installPatches
    action: 'aws:runCommand'
    timeoutSeconds: 7200
    onFailure: Abort
    inputs:
      DocumentName: AWS-RunPatchBaseline
      Parameters:
        SnapshotId: '{{SnapshotId}}'
        RebootOption: '{{RebootOption}}'
        Operation: '{{Operation}}'
      InstanceIds:
        - '{{ startInstances.InstanceIds }}'
  - name: stopInstance
    action: 'aws:changeInstanceState'
    maxAttempts: 1
    onFailure: Continue
    inputs:
      InstanceIds:
        - '{{ startInstances.InstanceIds }}'
      DesiredState: stopped
  - name: createImage
    action: 'aws:createImage'
    maxAttempts: 1
    onFailure: Continue
    inputs:
      InstanceId: '{{ startInstances.InstanceIds }}'
      ImageName: '{{ NewAMI }}'
      NoReboot: false
      ImageDescription: Patched AMI created by Automation
  - name: terminateInstance
```

```
    action: 'aws:changeInstanceState'
    maxAttempts: 1
    onFailure: Continue
    inputs:
      InstanceIds:
        - '{{ startInstances.InstanceIds }}'
      DesiredState: terminated
  - name: updateASG
    action: 'aws:executeScript'
    timeoutSeconds: 300
    maxAttempts: 1
    onFailure: Abort
    inputs:
      Runtime: python3.11
      Handler: update_asg
      InputPayload:
        TargetASG: '{{TargetASG}}'
        NewAMI: '{{createImage.ImageId}}'
      Script: |-
        from __future__ import print_function
        import datetime
        import json
        import time
        import boto3

        # create auto scaling and ec2 client
        asg = boto3.client('autoscaling')
        ec2 = boto3.client('ec2')

        def update_asg(event, context):
            print("Received event: " + json.dumps(event, indent=2))

            target_asg = event['TargetASG']
            new_ami = event['NewAMI']

            # get object for the ASG we're going to update, filter by name of
target ASG
            asg_query =
asg.describe_auto_scaling_groups(AutoScalingGroupNames=[target_asg])
            if 'AutoScalingGroups' not in asg_query or not
asg_query['AutoScalingGroups']:
                return 'No ASG found matching the value you specified.'
```

```
            # gets details of an instance from the ASG that we'll use to model the
 new launch template after
            source_instance_id = asg_query.get('AutoScalingGroups')[0]['Instances']
[0]['InstanceId']
            instance_properties = ec2.describe_instances(
                InstanceIds=[source_instance_id]
            )
            source_instance = instance_properties['Reservations'][0]['Instances']
[0]

            # create list of security group IDs
            security_groups = []
            for group in source_instance['SecurityGroups']:
                security_groups.append(group['GroupId'])

            # create a list of dictionary objects for block device mappings
            mappings = []
            for block in source_instance['BlockDeviceMappings']:
                volume_query = ec2.describe_volumes(
                    VolumeIds=[block['Ebs']['VolumeId']]
                )
                volume_details = volume_query['Volumes']
                device_name = block['DeviceName']
                volume_size = volume_details[0]['Size']
                volume_type = volume_details[0]['VolumeType']
                device = {'DeviceName': device_name, 'Ebs': {'VolumeSize':
 volume_size, 'VolumeType': volume_type}}
                mappings.append(device)

            # create new launch template using details returned from instance in
 the ASG and specify the newly patched AMI
            time_stamp = time.time()
            time_stamp_string =
 datetime.datetime.fromtimestamp(time_stamp).strftime('%m-%d-%Y_%H-%M-%S')
            new_template_name = f'{new_ami}_{time_stamp_string}'
            try:
                ec2.create_launch_template(
                    LaunchTemplateName=new_template_name,
                    LaunchTemplateData={
                        'BlockDeviceMappings': mappings,
                        'ImageId': new_ami,
                        'InstanceType': source_instance['InstanceType'],
                        'IamInstanceProfile': {
                            'Arn': source_instance['IamInstanceProfile']['Arn']
```

```
                        },
                        'KeyName': source_instance['KeyName'],
                        'SecurityGroupIds': security_groups
                    }
                )
            except Exception as e:
                return f'Exception caught: {str(e)}'
            else:
                # update ASG to use new launch template
                asg.update_auto_scaling_group(
                    AutoScalingGroupName=target_asg,
                    LaunchTemplate={
                        'LaunchTemplateName': new_template_name
                    }
                )
                return f'Updated ASG {target_asg} with new launch template
 {new_template_name} which uses AMI {new_ami}.'
outputs:
- createImage.ImageId
```

8. Wählen Sie Create automation (Automation erstellen).
9. Wählen Sie im Navigationsbereich Automatisierung und Automatisierung ausführen aus.
10. Wählen Sie auf der Seite Choose document (Dokument wählen), die Registerkarte Owned by me (In meinem Besitz).
11. Suchen Sie nach dem Runbook Patch AMIAnd UpdateAsg und wählen Sie die Schaltfläche auf der Karte Patch AMIAnd UpdateAsg aus.
12. Wählen Sie Weiter aus.
13. Wählen Sie Simple execution (Einfache Ausführung) aus.
14. Geben Sie Werte für die Eingabeparameter an. Stellen Sie sicher, dass die von Ihnen angegebenen `SubnetId` und `SecurityGroupIds` den Zugriff auf die öffentlichen Systems-Manager-Endpunkte oder Ihre Schnittstellenendpunkte für Systems Manager zulassen.
15. Wählen Sie Ausführen.
16. Wählen Sie nach Abschluss der Automatisierung in der EC2 Amazon-Konsole Auto Scaling und dann Launch Templates aus. Stellen Sie sicher, dass die neue Startvorlage angezeigt wird und dass sie das neue AMI verwendet.
17. Klicken Sie auf Auto Scaling und wählen Sie dann Auto-Scaling-Gruppen. Stellen Sie sicher, dass die Auto-Scaling-Gruppe die neue Startkonfiguration verwendet.

18. Beenden Sie mindestens eine Instance in Ihrer Auto–Scaling-Gruppe. Ersatz-Instances werden unter Verwendung der neuen AMI gestartet.

Verwendung von AWS -Support-Self-Service-Runbooks

In diesem Abschnitt wird beschrieben, wie Sie einige der Self-Service-Automatisierungen verwenden, die vom AWS -Support-Team erstellt wurden. Diese Automatisierungen unterstützen Sie bei der Verwaltung Ihrer AWS-Ressourcen.

Support Automation Workflows

Support Automation Workflows (SAW) sind Automatisierungs-Runbooks, die vomAWS -Support-Team erstellt wurden und verwaltet werden. Diese Runbooks helfen Ihnen bei der Behebung häufiger Probleme mit Ihren AWS-Ressourcen, überwachen proaktiv und identifizieren Netzwerkprobleme, erfassen und analysieren Protokolle und vieles mehr.

SAW-Runbooks verwenden das **`AWSSupport`**-Präfix. Beispiel, `AWSSupport-ActivateWindowsWithAmazonLicense`.

Außerdem haben AWS-Enterprise- und Business Support-Kunden auch Zugriff auf Runbooks, welche das **`AWSPremiumSupport`**-Präfix verwenden. Beispiel, `AWSPremiumSupport-TroubleshootEC2DiskUsage`.

Weitere Informationen zu AWS -Support finden Sie unter Erste Schritte mit AWS -Support.

Themen

- Führen Sie das EC2 Rescue-Tool auf nicht erreichbaren Instanzen aus
- Passwörter und SSH-Schlüssel auf EC2 Instanzen zurücksetzen

Führen Sie das EC2 Rescue-Tool auf nicht erreichbaren Instanzen aus

EC2Rescue kann Ihnen bei der Diagnose und Behebung von Problemen auf Amazon Elastic Compute Cloud (Amazon EC2) -Instances für Linux und helfenWindows Server. Sie können das Tool manuell ausführen, wie unter EC2 Rescue für Linux Server verwenden und EC2Rescue für Windows Server verwenden beschrieben. Sie können das Tool auch automatisch mit der Systems Manager Automation und dem **`AWSSupport-ExecuteEC2Rescue`**-Runbook ausführen. Automation ist ein Tool in AWS Systems Manager. Das **`AWSSupport-ExecuteEC2Rescue`**Runbook ist für die Ausführung einer Kombination von Systems Manager Manager-Aktionen, CloudFormation Aktionen

und Lambda-Funktionen konzipiert, mit denen die Schritte automatisiert werden, die normalerweise für die Verwendung EC2 von Rescue erforderlich sind.

Sie können das **`AWSSupport-ExecuteEC2Rescue`**-Runbook verwenden, um verschiedene Arten von Problemen bei Betriebssystemen (OS) zu behandeln und möglicherweise zu lösen. Instances mit verschlüsselten Root-Volumes werden nicht unterstützt. Eine vollständige Liste finden Sie in den folgenden Themen:

Windows: Weitere Informationen finden Sie unter EC2 Rescue für Windows Server über die Befehlszeile verwenden.

Linux und macOS: Einige Module von EC2 Rescue for Linux erkennen Probleme und versuchen, sie zu beheben. Weitere Informationen finden Sie in der `aws-ec2rescue-linux`-Dokumentation für jedes Modul auf GitHub.

Funktionsweise

Die Fehlerbehebung bei einer Instance mit Automation und dem **`AWSSupport-ExecuteEC2Rescue`**-Runbook funktioniert folgendermaßen:

- Sie geben die ID der nicht erreichbaren Instance an und starten das Runbook.
- Das System erstellt eine temporäre VPC und führt dann eine Reihe von Lambda-Funktionen aus, um die VPC zu konfigurieren.
- Das System identifiziert ein Subnetz für Ihre temporäre VPC in derselben Availability Zone wie die ursprüngliche Instance.
- Das System startet eine temporäre, SSM-fähige Helferobjekt-Instance.
- Das System stoppt Ihre ursprüngliche Instance und erstellt einen Backup. Anschließend fügt es das ursprüngliche Stamm-Volume an die Helferobjekt-Instance an.
- Das System verwendetRun Command, um EC2 Rescue auf der Helper-Instanz auszuführen. EC2Rescue identifiziert Probleme auf dem angehängten, ursprünglichen Root-Volume und versucht, diese zu beheben. Wenn der Vorgang abgeschlossen ist, fügt EC2 Rescue das Root-Volume wieder der ursprünglichen Instanz hinzu.
- Das System startet die ursprüngliche Instance neu und beendet die temporäre Instance. Das System beendet ebenso die temporäre VPC und die Lambda-Funktionen, die zu Beginn der Automatisierung erstellt wurden.

Bevor Sie beginnen

Bevor Sie die folgende Automation ausführen, führen Sie die folgenden Schritte aus:

- Kopieren Sie die Instance-ID der nicht erreichbaren Instance. Sie legen diese ID im Verfahren fest.
- Erfassen Sie optional die ID eines Subnetzes in derselben Availability Zone wie Ihre unerreichbare Instance. Die EC2 Rescue-Instanz wird in diesem Subnetz erstellt. Wenn Sie kein Subnetz angeben, erstellt Automation eine neue temporäre VPC in Ihrem. AWS-Konto Stellen Sie sicher, AWS-Konto dass mindestens eine VPC verfügbar ist. Standardmäßig können Sie fünf VPCs in einer Region erstellen. Wenn Sie VPCs in der Region bereits fünf erstellt haben, schlägt die Automatisierung fehl, ohne dass Änderungen an Ihrer Instanz vorgenommen werden. Weitere Informationen zu Amazon VPC-Kontingenten finden Sie unter VPC und Subnetze im Amazon VPC-Benutzerhandbuch.
- Optional können Sie eine AWS Identity and Access Management (IAM-) Rolle für die Automatisierung erstellen und angeben. Falls Sie diese Rolle nicht festlegen, wird die Automatisierung im Kontext des Benutzers ausgeführt, der die Automatisierung ausgeführt hat.

Gewähren von **`AWSSupport-EC2Rescue`**-Berechtigungen zum Durchführen von Aktionen auf Ihren Instances

EC2Rescue benötigt die Erlaubnis, während der Automatisierung eine Reihe von Aktionen auf Ihren Instances durchzuführen. Diese Aktionen rufen die EC2 Services AWS Lambda, IAM und Amazon auf, um sicher und geschützt zu versuchen, Probleme mit Ihren Instances zu beheben. Wenn Sie in Ihrer AWS-Konto und/oder Ihrer VPC über Administratorberechtigungen verfügen, können Sie die Automatisierung möglicherweise ausführen, ohne Berechtigungen zu konfigurieren, wie in diesem Abschnitt beschrieben. Falls Sie keine Administratorberechtigungen besitzen, müssen Sie oder ein Administrator Berechtigungen anhand einer der folgenden Optionen konfigurieren.

- Erteilen von Berechtigungen mithilfe von IAM-Richtlinien
- Erteilen von Berechtigungen mithilfe einer CloudFormation Vorlage

Erteilen von Berechtigungen mithilfe von IAM-Richtlinien

Sie können entweder die folgende IAM-Richtlinie als eingebundene Richtlinie an Ihren Benutzer, Ihre Gruppe oder Ihre Rolle anfügen. Sie können aber auch eine neue verwaltete IAM-Richtlinie erstellen und diese an Ihren Benutzer, Ihre Gruppe oder Ihre Rolle anfügen. Weitere Informationen zum Hinzufügen einer eingebundenen Richtlinie zu Ihrem Benutzerkonto, Ihrer Gruppe oder Ihrer Rolle

finden Sie unter Verwenden von eingebundenen Richtlinien. Weitere Informationen zum Erstellen einer neuen verwalteten Richtlinien finden Sie unter Verwenden von eingebundenen Richtlinien.

> **Note**
>
> Wenn Sie eine neue IAM-verwaltete Richtlinie erstellen, müssen Sie ihr auch die verwaltete Amazon SSMAutomation Role Policy hinzufügen, damit Ihre Instances mit der Systems Manager Manager-API kommunizieren können.

IAM-Richtlinie für 2Rescue AWSSupport-EC

Ersetzen Sie es *account ID* durch Ihre eigenen Informationen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "lambda:InvokeFunction",
                "lambda:DeleteFunction",
                "lambda:GetFunction"
            ],
            "Resource": "arn:aws:lambda:*:111122223333:function:AWSSupport-
EC2Rescue-*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::awssupport-ssm.*/*.template",
                "arn:aws:s3:::awssupport-ssm.*/*.zip"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
```

```
                "iam:CreateRole",
                "iam:CreateInstanceProfile",
                "iam:GetRole",
                "iam:GetInstanceProfile",
                "iam:PutRolePolicy",
                "iam:DetachRolePolicy",
                "iam:AttachRolePolicy",
                "iam:PassRole",
                "iam:AddRoleToInstanceProfile",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:DeleteRole",
                "iam:DeleteRolePolicy",
                "iam:DeleteInstanceProfile"
            ],
            "Resource": [
                "arn:aws:iam::111122223333:role/AWSSupport-EC2Rescue-*",
                "arn:aws:iam::111122223333:instance-profile/AWSSupport-EC2Rescue-
*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "lambda:CreateFunction",
                "ec2:CreateVpc",
                "ec2:ModifyVpcAttribute",
                "ec2:DeleteVpc",
                "ec2:CreateInternetGateway",
                "ec2:AttachInternetGateway",
                "ec2:DetachInternetGateway",
                "ec2:DeleteInternetGateway",
                "ec2:CreateSubnet",
                "ec2:DeleteSubnet",
                "ec2:CreateRoute",
                "ec2:DeleteRoute",
                "ec2:CreateRouteTable",
                "ec2:AssociateRouteTable",
                "ec2:DisassociateRouteTable",
                "ec2:DeleteRouteTable",
                "ec2:CreateVpcEndpoint",
                "ec2:DeleteVpcEndpoints",
                "ec2:ModifyVpcEndpoint",
                "ec2:Describe*",
                "autoscaling:DescribeAutoScalingInstances"
```

```
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Erteilen von Berechtigungen mithilfe einer CloudFormation Vorlage

CloudFormation automatisiert den Prozess der Erstellung von IAM-Rollen und -Richtlinien mithilfe einer vorkonfigurierten Vorlage. Gehen Sie wie folgt vor, um die erforderlichen IAM-Rollen und -Richtlinien für EC2 Rescue Automation zu erstellen, indem Sie CloudFormation

So erstellen Sie die erforderlichen IAM-Rollen und -Richtlinien für Rescue EC2

1. Laden Sie `AWSSupport-EC2RescueRole.zip` herunter und extrahieren Sie die `AWSSupport-EC2RescueRole.json`-Datei in ein Verzeichnis auf Ihrem lokalen Computer.
2. Wenn Sie AWS-Konto sich in einer speziellen Partition befinden, bearbeiten Sie die Vorlage, um die ARN-Werte in die für Ihre Partition zu ändern.

   Ändern Sie beispielsweise für `arn:aws` alle Fälle von in `arn:aws-cn`.
3. Melden Sie sich bei https://console.aws.amazon.com/cloudformation an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole.
4. Klicken Sie auf Create stack (Stack erstellen), With new resources (standard) (Mit neuen Ressourcen (Standard)).
5. Wählen Sie auf der Seite Create stack (Stack erstellen) unter Prerequisite - Prepare template (Voraussetzung – Vorlage vorbereiten) die Option Template is ready (Vorlage ist bereit) aus.
6. Wählen Sie unter Vorlage angeben die Option Vorlagendatei hochladen aus.
7. Wählen Sie Choose file (Datei auswählen) aus, navigieren Sie dann zu der `AWSSupport-EC2RescueRole.json`-Datei aus dem Verzeichnis, in dem Sie sie extrahiert haben, und wählen Sie sie aus.
8. Wählen Sie Weiter aus.
9. Geben Sie auf der Seite Specify stack details (Stack-Details angeben) für das Feld Stack name (Stack-Name) einen Namen ein, um diesen Stack zu identifizieren. Wählen Sie dann Next (Weiter) aus.

10. (Optional) Wenden Sie im Bereich „Tags“ ein oder mehrere name/value Tag-Schlüsselpaare auf den Stack an.

    Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Beispielsweise können Sie einen Stack kennzeichnen, um den Typ der ausgeführten Aufgaben, die Typen von Zielen oder anderen Ressourcen und die Umgebung zu identifizieren, in der er ausgeführt wird.
11. Wählen Sie Next (Weiter)
12. Überprüfen Sie auf der Seite Überprüfen die Stack-Details, scrollen Sie dann nach unten und wählen Sie die Option Ich bestätige, dass CloudFormation möglicherweise IAM-Ressourcen erstellt werden.
13. Wählen Sie Stack erstellen aus.

    CloudFormation zeigt für einige Minuten den Status CREATE_IN_PROGRESS an. Nach dem Erstellen des Stacks ändert sich der Status in CREATE_COMPLETE. Sie können auch auf das Aktualisierungssymbol klicken, um den Status des Erstellungsprozesses zu überprüfen.
14. Wählen Sie in der Stacks-Liste die Option neben den Stack, den Sie gerade erstellt haben, und wählen Sie dann die Registerkarte Outputs (Ausgaben).
15. Notieren Sie sich den Wert. Das ist der ARN von AssumeRole. Sie geben diesen ARN an, wenn Sie die Automatisierung in der nächsten Prozedur ausführen, Ausführen der Automation.

Ausführen der Automation

> **Important**
>
> Der folgende Automatisierung hält die nicht erreichbare Instance an. Das Anhalten der Instance kann zu Datenverlusten auf den angehängten Instance-Speicher-Volumes (sofern vorhanden) führen. Das Anhalten der Instance kann auch dazu führen, dass die öffentliche IP-Adresse geändert wird, wenn keine elastische IP-Adresse zugeordnet ist.

Führen Sie die **`AWSSupport-ExecuteEC2Rescue`**-Automation aus.

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation.

3. Wählen Sie Automatisierung ausführen.
4. Wählen Sie im Abschnitt Automation document (Automatisierungsdokument) die Option Owned by Amazon (Im Besitz von Amazon) aus der Liste aus.
5. Wählen Sie in der Runbooks-Liste die Schaltfläche auf der Karte für **`AWSSupport-ExecuteEC2Rescue`** und wählen Sie danach Weiter.
6. Klicken Sie auf der Seite Execute automation document (Automation-Dokument ausführen) auf Simple execution (Einfache Ausführung).
7. Überprüfen Sie im Abschnitt Document details (Dokumentdetails), ob Document version (Dokumentversion) auf die höchste Standardversion gesetzt ist. Beispiel: $DEFAULT oder 3 (default) (3 (Standard)).
8. Geben Sie im Abschnitt Input Parameters die folgenden Parameter an.

   a. Geben Sie für UnreachableInstanceIddie ID der nicht erreichbaren Instanz an.

   b. (Optional) Geben Sie für EC2RescueInstanceTypeeinen Instanztyp für die EC2 Rescue-Instanz an. Der Standard-Instance-Typ lautet `t2.medium`.

   c. Denn AutomationAssumeRolewenn Sie Rollen für diese Automatisierung mithilfe des weiter oben in diesem Thema beschriebenen CloudFormation Verfahrens erstellt haben, wählen Sie den ARN aus AssumeRole , den Sie in der CloudFormation Konsole erstellt haben.

   d. (Optional) Geben Sie für einen S3-Bucket an LogDestination, wenn Sie bei der Fehlerbehebung für Ihre Instance Protokolle auf Betriebssystemebene sammeln möchten. Protokolle werden automatisch in den angegebenen Bucket hochgeladen.

   e. Geben Sie für SubnetIdein Subnetz in einer vorhandenen VPC in derselben Availability Zone wie die nicht erreichbare Instance an. Standardmäßig erstellt Systems Manager eine neue VPC, aber Sie können ein Subnetz in einer vorhandenen VPC angeben, wenn Sie möchten.

   > **Note**
   >
   > Wenn Sie die Option zum Erstellen eines Buckets oder einer Subnetz-ID nicht sehen, überprüfen Sie, ob Sie die neueste Default-Version des Runbooks verwenden.

9. (Optional) Wenden Sie im Bereich „Tags“ ein oder mehrere name/value Tag-Schlüsselpaare an, um beispielsweise die Automatisierung leichter zu identifizieren. `Key=Purpose,Value=EC2Rescue`
10. Wählen Sie Ausführen.

Das Runbook erstellt ein Backup AMI als Teil der Automatisierung. Alle anderen von der Automatisierung erstellten Ressourcen werden automatisch gelöscht, aber dieses AMI verbleibt in Ihrem Konto. Der AMI-Name wird unter Verwendung der folgenden Konvention generiert:

Backup-AMI: AWSSupport-EC 2Rescue: *UnreachableInstanceId*

Sie finden dies AMI in der EC2 Amazon-Konsole, indem Sie nach der Automation-Ausführungs-ID suchen.

Passwörter und SSH-Schlüssel auf EC2 Instanzen zurücksetzen

Sie können das `AWSSupport-ResetAccess` Runbook verwenden, um die lokale Administratorkennwortgenerierung auf Amazon Elastic Compute Cloud (Amazon EC2) -Instances automatisch wieder zu aktivieren Windows Server und einen neuen SSH-Schlüssel auf EC2 Instances für Linux zu generieren. Das `AWSSupport-ResetAccess` Runbook ist so konzipiert, dass es eine Kombination aus AWS Systems Manager Aktionen, AWS CloudFormation Aktionen und AWS Lambda Funktionen ausführt, die die Schritte automatisieren, die normalerweise zum Zurücksetzen des lokalen Administratorkennworts erforderlich sind.

Mithilfe von Automation, einem im `AWSSupport-ResetAccess` Runbook enthaltenen Tool AWS Systems Manager, können Sie die folgenden Probleme lösen:

Windows

Sie haben das EC2 key pair verloren: Um dieses Problem zu lösen, können Sie das AWSSupport-ResetAccessRunbook verwenden, um eine kennwortfähige Instance AMI aus Ihrer aktuellen Instance zu erstellen, eine neue Instance über das AMI zu starten und ein key pair auszuwählen, das Ihnen gehört.

Sie haben das lokale Administratorkennwort verloren: Um dieses Problem zu lösen, können Sie das `AWSSupport-ResetAccess` Runbook verwenden, um ein neues Passwort zu generieren, das Sie mit dem aktuellen EC2 key pair entschlüsseln können.

Linux

Sie haben Ihr EC2 key pair verloren, oder Sie haben den SSH-Zugriff auf die Instance mit einem Schlüssel konfiguriert, den Sie verloren haben: Um dieses Problem zu lösen, können Sie das `AWSSupport-ResetAccess` Runbook verwenden, um einen neuen SSH-Schlüssel für Ihre aktuelle Instance zu erstellen, mit dem Sie sich erneut mit der Instance verbinden können.

> **Note**
>
> Wenn Ihre EC2 Instanz für für Systems Manager konfiguriert Windows Server ist, können Sie auch Ihr lokales Administratorkennwort mithilfe von EC2 Rescue und zurücksetzen AWS Systems Manager Run Command. Weitere Informationen finden Sie unter Verwenden von EC2 Rescue for Windows Server mit Systems Manager Run Command im EC2 Amazon-Benutzerhandbuch.

Ähnliche Informationen

Stellen Sie mithilfe von PuTTY im EC2 Amazon-Benutzerhandbuch von Windows aus eine Connect zu Ihrer Linux-Instance her

Funktionsweise

Die Fehlerbehebung bei einer Instance mit Automation und dem `AWSSupport-ResetAccess`-Runbook funktioniert folgendermaßen:

- Sie geben die ID der Instance an und führen das Runbook aus.
- Das System erstellt eine temporäre VPC und führt dann eine Reihe von Lambda-Funktionen aus, um die VPC zu konfigurieren.
- Das System identifiziert ein Subnetz für Ihre temporäre VPC in derselben Availability Zone wie die ursprüngliche Instance.
- Das System startet eine temporäre, SSM-fähige Helferobjekt-Instance.
- Das System stoppt Ihre ursprüngliche Instance und erstellt einen Backup. Anschließend fügt es das ursprüngliche Stamm-Volume an die Helferobjekt-Instance an.
- Das System verwendetRun Command, um EC2 Rescue auf der Helper-Instance auszuführen. Unter Windows ermöglicht EC2 Rescue die Passwortgenerierung für den lokalen Administrator mithilfe von EC2 Config oder EC2 Launch auf dem angehängten, ursprünglichen Root-Volume. Unter Linux generiert EC2 Rescue einen neuen SSH-Schlüssel, fügt ihn ein und speichert den privaten Schlüssel verschlüsselt unter. Parameter Store Wenn der Vorgang abgeschlossen ist, fügt EC2 Rescue das Root-Volume wieder der ursprünglichen Instanz hinzu.
- Das System erstellt eine neue Amazon Machine Image (AMI) Ihrer Instance, nachdem die Kennwortgenerierung aktiviert wurde. Sie können dies verwendenAMI, um eine neue EC2 Instanz zu erstellen und bei Bedarf ein neues key pair zuzuordnen.

- Das System startet die ursprüngliche Instance neu und beendet die temporäre Instance. Das System beendet ebenso die temporäre VPC und die Lambda-Funktionen, die zu Beginn der Automatisierung erstellt wurden.
- Windows: Ihre Instance generiert ein neues Passwort, das Sie mit dem aktuellen key pair, das der Instance zugewiesen ist, von der EC2 Amazon-Konsole aus dekodieren können.

  Linux: Sie können eine SSH-Verbindung zur Instance herstellen, indem Sie den SSH-Schlüssel verwenden, der im Systems Manager Parameter Store als /ec2rl/openssh/ ***instance ID*** /key gespeichert ist.

Bevor Sie beginnen

Bevor Sie die folgende Automation ausführen, führen Sie die folgenden Schritte aus:

- Kopieren Sie die Instance-ID der Instance, auf der Sie das Administratorpasswort zurücksetzen möchten. Sie legen diese ID im Verfahren fest.
- Erfassen Sie optional die ID eines Subnetzes in derselben Availability Zone wie Ihre unerreichbare Instance. Die Rescue-Instanz wird in diesem Subnetz erstellt. EC2 Wenn Sie kein Subnetz angeben, erstellt Automation eine neue temporäre VPC in Ihrem. AWS-Konto Stellen Sie sicher, AWS-Konto dass mindestens eine VPC verfügbar ist. Standardmäßig können Sie fünf VPCs in einer Region erstellen. Wenn Sie VPCs in der Region bereits fünf erstellt haben, schlägt die Automatisierung fehl, ohne dass Änderungen an Ihrer Instanz vorgenommen werden. Weitere Informationen zu Amazon VPC-Kontingenten finden Sie unter VPC und Subnetze im Amazon VPC-Benutzerhandbuch.
- Optional können Sie eine AWS Identity and Access Management (IAM-) Rolle für die Automatisierung erstellen und angeben. Falls Sie diese Rolle nicht festlegen, wird die Automatisierung im Kontext des Benutzers ausgeführt, der die Automatisierung ausgeführt hat.

Erteilen Sie AWSSupport-EC 2Rescue-Berechtigungen zur Durchführung von Aktionen auf Ihren Instances

EC2Rescue benötigt die Erlaubnis, während der Automatisierung eine Reihe von Aktionen auf Ihren Instances durchzuführen. Diese Aktionen rufen die EC2 Services AWS Lambda, IAM und Amazon auf, um sicher und geschützt zu versuchen, Probleme mit Ihren Instances zu beheben. Wenn Sie in Ihrer AWS-Konto und/oder Ihrer VPC über Administratorberechtigungen verfügen, können Sie die Automatisierung möglicherweise ausführen, ohne Berechtigungen zu konfigurieren, wie in diesem

Abschnitt beschrieben. Falls Sie keine Administratorberechtigungen besitzen, müssen Sie oder ein Administrator Berechtigungen anhand einer der folgenden Optionen konfigurieren.

- Erteilen von Berechtigungen mithilfe von IAM-Richtlinien
- Erteilen von Berechtigungen mithilfe einer CloudFormation Vorlage

Erteilen von Berechtigungen mithilfe von IAM-Richtlinien

Sie können entweder die folgende IAM-Richtlinie als eingebundene Richtlinie an Ihren Benutzer, Ihre Gruppe oder Ihre Rolle anfügen. Sie können aber auch eine neue verwaltete IAM-Richtlinie erstellen und diese an Ihren Benutzer, Ihre Gruppe oder Ihre Rolle anfügen. Weitere Informationen zum Hinzufügen einer eingebundenen Richtlinie zu Ihrem Benutzerkonto, Ihrer Gruppe oder Ihrer Rolle finden Sie unter Verwenden von eingebundenen Richtlinien. Weitere Informationen zum Erstellen einer neuen verwalteten Richtlinien finden Sie unter Verwenden von eingebundenen Richtlinien.

> **Note**
>
> Wenn Sie eine neue IAM-verwaltete Richtlinie erstellen, müssen Sie ihr auch die verwaltete Amazon SSMAutomation Role Policy hinzufügen, damit Ihre Instances mit der Systems Manager Manager-API kommunizieren können.

IAM-Richtlinie für **`AWSSupport-ResetAccess`**

Ersetzen Sie es *`account ID`* durch Ihre eigenen Informationen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "lambda:InvokeFunction",
                "lambda:DeleteFunction",
                "lambda:GetFunction"
            ],
            "Resource": "arn:aws:lambda:*:111122223333:function:AWSSupport-
EC2Rescue-*",
            "Effect": "Allow"
```

```
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::awssupport-ssm.*/*.template",
                "arn:aws:s3:::awssupport-ssm.*/*.zip"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "iam:CreateRole",
                "iam:CreateInstanceProfile",
                "iam:GetRole",
                "iam:GetInstanceProfile",
                "iam:PutRolePolicy",
                "iam:DetachRolePolicy",
                "iam:AttachRolePolicy",
                "iam:PassRole",
                "iam:AddRoleToInstanceProfile",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:DeleteRole",
                "iam:DeleteRolePolicy",
                "iam:DeleteInstanceProfile"
            ],
            "Resource": [
                "arn:aws:iam::111122223333:role/AWSSupport-EC2Rescue-*",
                "arn:aws:iam::111122223333:instance-profile/AWSSupport-EC2Rescue-
*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "lambda:CreateFunction",
                "ec2:CreateVpc",
                "ec2:ModifyVpcAttribute",
                "ec2:DeleteVpc",
                "ec2:CreateInternetGateway",
                "ec2:AttachInternetGateway",
                "ec2:DetachInternetGateway",
```

```
                "ec2:DeleteInternetGateway",
                "ec2:CreateSubnet",
                "ec2:DeleteSubnet",
                "ec2:CreateRoute",
                "ec2:DeleteRoute",
                "ec2:CreateRouteTable",
                "ec2:AssociateRouteTable",
                "ec2:DisassociateRouteTable",
                "ec2:DeleteRouteTable",
                "ec2:CreateVpcEndpoint",
                "ec2:DeleteVpcEndpoints",
                "ec2:ModifyVpcEndpoint",
                "ec2:Describe*"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Erteilen von Berechtigungen mithilfe einer CloudFormation Vorlage

CloudFormation automatisiert den Prozess der Erstellung von IAM-Rollen und -Richtlinien mithilfe einer vorkonfigurierten Vorlage. Gehen Sie wie folgt vor, um die erforderlichen IAM-Rollen und -Richtlinien für EC2 Rescue Automation zu erstellen, indem Sie CloudFormation

So erstellen Sie die erforderlichen IAM-Rollen und -Richtlinien für Rescue EC2

1. Laden Sie AWSSupport-EC2RescueRole.zip herunter und extrahieren Sie die `AWSSupport-EC2RescueRole.json`-Datei in ein Verzeichnis auf Ihrem lokalen Computer.
2. Wenn Sie AWS-Konto sich in einer speziellen Partition befinden, bearbeiten Sie die Vorlage, um die ARN-Werte in die für Ihre Partition zu ändern.

   Ändern Sie beispielsweise für `arn:aws` alle Fälle von in `arn:aws-cn`.
3. Melden Sie sich bei https://console.aws.amazon.com/cloudformation an AWS-Managementkonsole und öffnen Sie die CloudFormation Konsole.
4. Klicken Sie auf Create stack (Stack erstellen), With new resources (standard) (Mit neuen Ressourcen (Standard)).

5. Wählen Sie auf der Seite Create stack (Stack erstellen) unter Prerequisite - Prepare template (Voraussetzung – Vorlage vorbereiten) die Option Template is ready (Vorlage ist bereit) aus.
6. Wählen Sie unter Vorlage angeben die Option Vorlagendatei hochladen aus.
7. Wählen Sie Choose file (Datei auswählen) aus, navigieren Sie dann zu der `AWSSupport-EC2RescueRole.json`-Datei aus dem Verzeichnis, in dem Sie sie extrahiert haben, und wählen Sie sie aus.
8. Wählen Sie Weiter aus.
9. Geben Sie auf der Seite Specify stack details (Stack-Details angeben) für das Feld Stack name (Stack-Name) einen Namen ein, um diesen Stack zu identifizieren. Wählen Sie dann Next (Weiter) aus.
10. (Optional) Wenden Sie im Bereich „Tags“ ein oder mehrere name/value Tag-Schlüsselpaare auf den Stack an.

    Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Beispielsweise können Sie einen Stack kennzeichnen, um den Typ der ausgeführten Aufgaben, die Typen von Zielen oder anderen Ressourcen und die Umgebung zu identifizieren, in der er ausgeführt wird.
11. Wählen Sie Next (Weiter)
12. Überprüfen Sie auf der Seite Überprüfen die Stack-Details, scrollen Sie dann nach unten und wählen Sie die Option Ich bestätige, dass CloudFormation möglicherweise IAM-Ressourcen erstellt werden.
13. CloudFormation zeigt für einige Minuten den Status CREATE_IN_PROGRESS an. Nach dem Erstellen des Stacks ändert sich der Status in CREATE_COMPLETE. Sie können auch auf das Aktualisierungssymbol klicken, um den Status des Erstellungsprozesses zu überprüfen.
14. Wählen Sie in der Stackliste die Option neben dem Stack, den Sie gerade erstellt haben, und wählen Sie dann die Registerkarte Outputs (Ausgaben) aus.
15. Kopieren Sie den Value (Wert). Das ist der ARN von AssumeRole. Sie geben diesen ARN bei der Ausführung der Automation an.

Ausführen der Automation

Im folgenden Verfahren wird beschrieben, wie Sie mithilfe der AWS Systems Manager -Konsole das `AWSSupport-ResetAccess`-Runbook ausführen.

> **Important**
>
> Die folgende Automatisierung hält die Instance an. Das Anhalten der Instance kann zu Datenverlusten auf den angehängten Instance-Speicher-Volumes (sofern vorhanden) führen. Das Anhalten der Instance kann auch dazu führen, dass die öffentliche IP-Adresse geändert wird, wenn keine elastische IP-Adresse zugeordnet ist. Um diese Konfigurationsänderungen zu vermeiden, verwenden Sie Run Command, um den Zugriff zurückzusetzen. Weitere Informationen finden Sie unter Verwenden von EC2 Rescue for Windows Server mit Systems Manager Run Command im EC2 Amazon-Benutzerhandbuch.

Um die AWSSupport-ResetAccess Automatisierung auszuführen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation.
3. Wählen Sie Automatisierung ausführen.
4. Wählen Sie im Abschnitt Automation document (Automatisierungsdokument) die Option Owned by Amazon (Im Besitz von Amazon) aus der Liste aus.
5. Wählen Sie in der Runbooks-Liste die Schaltfläche auf der Karte für AWSSupport-ResetAccess und wählen Sie danach Weiter.
6. Klicken Sie auf der Seite Execute automation document (Automation-Dokument ausführen) auf Simple execution (Einfache Ausführung).
7. Überprüfen Sie im Abschnitt Document details (Dokumentdetails), ob Document version (Dokumentversion) auf die höchste Standardversion gesetzt ist. Beispiel: $DEFAULT oder 3 (default) (3 (Standard)).
8. Geben Sie im Abschnitt Input Parameters die folgenden Parameter an.
    a. Geben Sie für InstanceID die ID der nicht erreichbaren Instance an.
    b. Geben Sie für SubnetIdein Subnetz in einer vorhandenen VPC in derselben Availability Zone wie die angegebene Instance an. Standardmäßig erstellt Systems Manager eine neue VPC, aber Sie können ein Subnetz in einer vorhandenen VPC angeben, wenn Sie möchten.

> **Note**
>
> Wenn Sie die Option zur Angabe einer Subnetz-ID nicht sehen, überprüfen Sie, ob Sie die neueste Default-Version des Runbooks verwenden.

c. Geben Sie für EC2RescueInstanceTypeeinen Instanztyp für die EC2 Rescue-Instanz an. Der Standard-Instance-Typ lautet `t2.medium`.

d. Denn AssumeRolewenn Sie Rollen für diese Automatisierung mithilfe des weiter oben in diesem Thema beschriebenen CloudFormation Verfahrens erstellt haben, geben Sie den AssumeRole ARN an, den Sie in der CloudFormation Konsole notiert haben.

9. (Optional) Wenden Sie im Bereich „Tags“ ein oder mehrere name/value Tag-Schlüsselpaare an, um beispielsweise die Automatisierung leichter zu identifizierenKey=Purpose,Value=ResetAccess.
10. Wählen Sie Ausführen.
11. Zur Überwachung des Fortschritts der Automatisierung wählen Sie die laufende Automatisierung und dann die Registerkarte Steps (Schritte). Wenn die Automatisierung abgeschlossen ist, wählen Sie die Registerkarte Descriptions (Beschreibungen) und dann View output (Ausgabe anzeigen), um die Ergebnisse anzuzeigen. Zum Anzeigen der Ausgabe der einzelnen Schritte wählen Sie die Registerkarte Steps (Schritte) und dann neben einem Schritt View Outputs (Ausgabe anzeigen) aus.

Das Runbook erstellt ein Backup AMI und ein kennwortaktiviertes AMI als Teil der Automatisierung. Alle anderen von der Automatisierung erstellten Ressourcen werden automatisch gelöscht, aber diese AMIs verbleiben in Ihrem Konto. Die AMIs-Namen werden unter Verwendung der folgenden Konvention generiert:

- Backup AMI: `AWSSupport-EC2Rescue:`*`InstanceID`*
- Passwort-fähiges AMI: AWSSupport-EC 2Rescue: Passwort-fähiges AMI von *`Instance ID`*

Sie finden diese AMIs, indem Sie nach der Automation-Ausführungs-ID suchen.

Für Linux wird der neue private SSH-Schlüssel für Ihre Instance verschlüsselt in Parameter Store gespeichert. Der Parametername ist /ec2rl/openssh/ /key. ***`instance ID`***

Übergabe von Daten an Automation mithilfe von Eingangstransformatoren

Dieses AWS Systems Manager Automation-Tutorial zeigt, wie Sie die Input-Transformer-Funktion von Amazon verwenden EventBridge , um die einer Amazon Elastic Compute Cloud (Amazon EC2) -Instance aus einem Ereignis zur Änderung `instance-id` des Instance-Status zu extrahieren. Automation ist ein Tool in AWS Systems Manager. Wir verwenden den Eingangstransformator, um diese Daten als `InstanceId`-Eingabeparameter an das `AWS-CreateImage`-Runbook zu übergeben. Die Regel wird ausgelöst, wenn eine beliebige Instance in den Status „`stopped`“ übergeht.

Weitere Informationen zur Arbeit mit Eingangstransformatoren finden Sie unter Tutorial: Use Input Transformer to Customize What to the Event Target im EventBridge Amazon-Benutzerhandbuch.

Bevor Sie beginnen

Stellen Sie sicher, dass Sie Ihrer Systems Manager Automation-Servicerolle EventBridge die erforderlichen Berechtigungen und Vertrauensrichtlinien für hinzugefügt haben. Weitere Informationen finden Sie unter Überblick über die Verwaltung von Zugriffsberechtigungen für Ihre EventBridge Ressourcen im EventBridge Amazon-Benutzerhandbuch.

So verwenden Sie Eingangstransformatoren mit Automatisierung

1. Öffnen Sie die EventBridge Amazon-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Regeln aus.
3. Wählen Sie Regel erstellen aus.
4. Geben Sie einen Namen und eine Beschreibung für die Regel ein.

   Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben Region und auf demselben Event Bus haben.
5. Wählen Sie für Event Bus den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel auf entsprechende Ereignisse reagiert, die von Ihnen selbst stammen AWS-Konto, wählen Sie Standard. Wenn ein AWS-Service in Ihrem Konto ein Ereignis ausgibt, wird es immer an den Standard-Event-Bus Ihres Kontos weitergeleitet.
6. Bei Regeltyp wählen Sie Regel mit einem Ereignismuster aus.
7. Wählen Sie Weiter aus.
8. Wählen Sie als Eventquelle AWS Events oder EventBridge Partnerevents aus.
9. Wählen Sie im Abschnitt Ereignismuster die Option Musterformular verwenden aus.
10. Als Event source (Ereignisquelle) wählen Sie AWS -Services aus.

11. Wählen Sie unter AWS -Service die Option EC2 aus.
12. Wählen Sie in Event Type (Ereignistyp) EC2 EC2 Instance State-change Notification (Benachrichtigung über die Statusänderung der EC2-Instance) aus.
13. Wählen Sie für Ereignistyp-Spezifikation 1 die Option Spezifische Zustände und anschließend Beendet aus.
14. Wählen Sie für Ereignistyp-Spezifikation 2 die Option Beliebige Instanz oder wählen Sie Spezifische Instanz-ID (s) IDs aus und geben Sie die zu überwachenden Instanzen ein.
15. Wählen Sie Weiter aus.
16. Bei Zieltypen wählen Sie AWS -Service aus.
17. Für Select target (Ziel auswählen), wählen Sie Systems Manager Automation.
18. Wählen Sie für Dokument die Option AWS- CreateImage.
19. Wählen Sie im Abschnitt Configure automation parameter(s) (Automatisierungsparameter konfigurieren) Input Transformer (Eingangstransformator) aus.
20. Geben Sie für Input path (Eingabepfad) den Wert **`{"instance":"$.detail.instance-id"}`** ein.
21. Geben Sie für Template (Vorlage) den Wert **`{"InstanceId":[<instance>]}`** ein.
22. Wählen Sie für Execution role (Ausführungsrolle) die Option Use existing role (Vorhandene Rolle) verwenden und wählen Sie Ihre Automation-Servicerolle.
23. Wählen Sie Weiter aus.
24. (Optional) Geben Sie ein oder mehrere Tags für die Regel ein. Weitere Informationen finden Sie unter Tagging Your Amazon EventBridge Resources im EventBridge Amazon-Benutzerhandbuch.
25. Wählen Sie Weiter aus.
26. Überprüfen Sie die Details der Regel und wählen Sie dann Regel erstellen aus.

Erfahren Sie mehr über die von Systems Manager Automation zurückgegebenen Status

AWS Systems Manager-Automation meldet detaillierte Statusinformationen zu den zahlreichen Status einer/einem Automatisierungsaktion oder -schritt, wenn Sie eine Automatisierung ausführen, sowie für die gesamte Automatisierung. Automation ist ein Tool in AWS Systems Manager. Sie können Automatisierungsstatus mithilfe der folgenden Methoden überwachen:

- Überwachen Sie den Ausführungsstatus in der Systems Manager Automation-Konsole.

- Verwenden Sie Ihre bevorzugten Befehlszeilen-Tools. Für den AWS Command Line Interface (AWS CLI) können Sie describe-automation-step-executions oder get-automation-execution verwenden. Für den AWS Tools for Windows PowerShell können Sie Get-SSMAutomationStepExecution oder Get-SSMAutomationExecution verwenden.
- Konfigurieren Sie Amazon EventBridge, um auf Änderungen des Aktions- oder Automatisierungsstatus zu reagieren.

Weitere Informationen zur Handhabung von Timeouts in einer Automatisierung finden Sie unter Behandeln von Timeouts in Runbooks.

Informationen zu Automatisierungsstatus

Automation meldet Statusdetails für einzelne Automatisierungsaktionen zusätzlich zur Gesamtautomatisierung.

Der Gesamtautomatisierungsstatus kann von dem Status, der von einer einzelnen Aktion oder einem Schritt gemeldet wird, wie in den folgenden Tabellen angegeben, abweichen.

Detaillierter Status für Aktionen

Status	Details
Ausstehend	Der Schritt wurde noch nicht ausgeführt. Wenn Ihre Automatisierung bedingte Aktionen verwendet, bleiben die Schritte in diesem Zustand, nachdem eine Automatisierung abgeschlossen wurde, wenn die Bedingung für die Ausführung des Schritts nicht erfüllt wurde. Schritte bleiben auch in diesem Zustand, wenn die Automatisierung abgebrochen wird, bevor der Schritt ausgeführt wird.
InProgress	Der Schritt läuft.
Ein Moment	Der Schritt wartet auf Eingabe.
Herzlichen Glückwunsch	Der Schritt wurde erfolgreich ausgeführt. Diese ist ein Terminalstatus.

Status	Details
TimedOut	Ein Schritt oder eine Genehmigung wurde nicht vor dem angegebenen Zeitüberschreitung szeitraum abgeschlossen. Diese ist ein Terminalstatus.
Abbrechen	Der Schritt wird gerade angehalten, nachdem er von einem Anforderer abgebrochen wurde.
Abgebrochen	Der Schritt wurde von einem Anforderer angehalten, bevor er abgeschlossen wurde. Diese ist ein Terminalstatus.
Fehlgeschlagen	Der Schritt wurde nicht erfolgreich abgeschlo ssen. Diese ist ein Terminalstatus.
Exited	Nur durch die `aws:loop`-Aktion zurückgek ehrt. Die Schleife wurde nicht vollständig abgeschlossen. Ein Schritt innerhalb der Schleife wurde mithilfe der Eigenschaften `nextStep`, `onCancel` oder `onFailure` zu einem Schritt außerhalb der Schleife verschobe n.

Detaillierter Status für eine Automatisierung

Status	Details
Ausstehend	Die Automatisierung hat noch nicht begonnen.
InProgress	Die Automatisierung läuft.
Ein Moment	Die Automatisierung wartet auf Eingabe.
Herzlichen Glückwunsch	Die Automatisierung wurde erfolgreich abgeschlossen. Diese ist ein Terminalstatus.

Status	Details
TimedOut	Ein Schritt oder eine Genehmigung wurde nicht vor dem angegebenen Zeitüberschreitung szeitraum abgeschlossen. Diese ist ein Terminalstatus.
Abbrechen	Die Automatisierung wird gerade angehalten, nachdem sie von einem Anforderer abgebroch en wurde.
Abgebrochen	Die Automatisierung wurde von einem Anforderer angehalten, bevor diese abgeschlo ssen wurde. Diese ist ein Terminalstatus.
Fehlgeschlagen	Die Automatisierung wurde nicht erfolgreich abgeschlossen. Diese ist ein Terminalstatus.

Fehlerbehebung für Systems Manager Automation.

Im Folgenden finden Sie Informationen zur Behandlung von Problemen mit AWS Systems Manager Automation, einem Tool in AWS Systems Manager. Dieses Thema enthält spezifische Aufgaben zum Beheben von Problemen basierend auf Automation-Fehlermeldungen.

Themen

- Häufige Automation-Fehler
- Fehler beim Start der Automation-Ausführung
- Ausführung gestartet, aber Status ist fehlgeschlagen
- Ausführung gestartet, aber mit Zeitüberschreitung

Häufige Automation-Fehler

Dieser Abschnitt enthält Informationen zu gängigen Automation-Fehlern.

VPC nicht definiert 400

Standardmäßig erstellt das System eine temporäre Instance in der Standard-VPC (172.30.0.0/16), wenn Automation das Runbook `AWS-UpdateLinuxAmi` oder das Runbook `AWS-`

`UpdateWindowsAmi` ausführt. Wenn Sie die Standard-VPC gelöscht haben, erhalten Sie den folgenden Fehler:

```
VPC not defined 400
```

Um dieses Problem zu lösen, müssen Sie einen Wert für den `SubnetId`-Eingabeparameter angeben.

Fehler beim Start der Automation-Ausführung

Eine Automatisierung kann mit dem Fehler „Zugriff verweigert“ oder „Ungültige Rollenübernahme“ fehlschlagen, wenn Sie AWS Identity and Access Management (IAM)-Rollen und Richtlinien für die Automatisierung nicht ordnungsgemäß konfiguriert haben.

Zugriff verweigert

Die folgenden Beispiele beschreiben Situationen, in denen eine Automatisierung nicht gestartet werden konnte, weil der Zugriff verweigert wurde.

Zugriff auf die Systems Manager API verweigert

Fehlermeldung: `User: user arn isn't authorized to perform: ssm:StartAutomationExecution on resource: document arn (Service: AWSSimpleSystemsManagement; Status Code: 400; Error Code: AccessDeniedException; Request ID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)`

- Mögliche Ursache 1: Der Benutzer, der versucht, die Automatisierung zu starten, verfügt nicht über die Berechtigung zum Aufrufen der `StartAutomationExecution`-API. Um dieses Problem zu beheben, fügen Sie die erforderliche IAM-Richtlinie dem Benutzer an, der zum Starten der Automatisierung verwendet wurde.
- Mögliche Ursache 2: Der Benutzer, der versucht, die Automatisierung zu starten, verfügt über die Berechtigung zum Aufrufen der `StartAutomationExecution`-API, jedoch nicht über die Berechtigung zum Aufrufen der API mithilfe des spezifischen Runbooks. Um dieses Problem zu beheben, fügen Sie die erforderliche IAM-Richtlinie dem Benutzer an, der zum Starten der Automatisierung verwendet wurde.

Zugriff aufgrund fehlender PassRole-Berechtigungen verweigert

Fehlermeldung: `User: user arn isn't authorized to perform: iam:PassRole on resource: automation assume role arn (Service: AWSSimpleSystemsManagement;`

`Status Code: 400; Error Code: AccessDeniedException; Request ID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)`

Der Benutzer, der versucht, die Automatisierung zu starten, verfügt nicht über die PassRole-Berechtigung für die übernommene Rolle. Um dieses Problem zu beheben, fügen Sie die Richtlinie iam:PassRole der Rolle des Benutzers an, der versucht, die Automatisierung zu starten. Weitere Informationen finden Sie unter Aufgabe 2: Hängen Sie die iam: PassRole -Richtlinie an Ihre Automatisierungsrolle an.

Ungültige Übernahmerolle

Beim Ausführen einer Automatisierung wird eine Übernahmerolle entweder im Runbook bereitgestellt oder als Parameterwert für das Runbook weitergeleitet. Unterschiedliche Arten von Fehlern können auftreten, wenn die Übernahmerolle nicht angegeben oder nicht ordnungsgemäß konfiguriert ist.

Falsch formatierte Übernahmerolle

Fehlermeldung: `The format of the supplied assume role ARN isn't valid.`
Die Übernahmerolle ist falsch formatiert. Um dieses Problem zu lösen, stellen Sie sicher, dass beim Starten der Automatisierung eine gültige Übernahmerolle in Ihrem Runbook oder als Laufzeitparameter angegeben ist.

Die Übernahmerolle kann nicht übernommen werden

Fehlermeldung: `The defined assume role is unable to be assumed. (Service: AWSSimpleSystemsManagement; Status Code: 400; Error Code: InvalidAutomationExecutionParametersException; Request ID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)`

- Mögliche Ursache 1: Die Übernahmerolle ist nicht vorhanden. Sie lösen dieses Problem, indem Sie die Rolle erstellen. Weitere Informationen finden Sie unter the section called “Einrichten der Automatisierung”. Spezifische Details zum Erstellen dieser Rolle sind im folgenden Thema beschrieben: Aufgabe 1: Erstellen einer Servicerolle für Automation.
- Mögliche Ursache 2: Die Rollenübernahme hat keine Vertrauensbeziehung zum Systems Manager-Service. Um dieses Problem zu lösen, erstellen Sie die Vertrauensbeziehung. Weitere Informationen finden Sie unter Ich kann keine Rolle übernehmen im IAM-Benutzerhandbuch.

Ausführung gestartet, aber Status ist fehlgeschlagen

Aktionsspezifische Fehler

Runbooks enthalten Schritte und die Schritte werden der Reihe nach ausgeführt. Jeder Schritt ruft mindestens eine AWS-Service-API auf. Die APIs bestimmen die Eingaben, das Verhalten und die Ausgaben des Schritts. Es gibt mehrere Stellen, an denen ein Fehler kann dazu führen, dass ein Schritt fehlschlägt. Fehlermeldungen geben an, wann und wo ein Fehler aufgetreten ist.

Wenn Sie eine Fehlermeldung in der Amazon Elastic Compute Cloud (Amazon EC2)-Konsole sehen, wählen Sie den Link View Outputs (Ausgaben anzeigen) für den fehlgeschlagenen Link. Wenn Sie eine Fehlermeldung aus der Befehlszeilen-Schnittstelle (AWS CLI) sehen, rufen Sie `get-automation-execution` auf und suchen Sie nach dem `FailureMessage`-Attribut in der fehlgeschlagenen `StepExecution`.

In den folgenden Beispielen ist ein Schritt im Zusammenhang mit der `aws:runInstance`-Aktion fehlgeschlagen. Jedes Beispiel untersucht einen anderen Fehlertyp.

Fehlendes Image

Fehlermeldung: `Automation Step Execution fails when it's launching the instance(s). Get Exception from RunInstances API of ec2 Service. Exception Message from RunInstances API: [The image id '[ami id]' doesn't exist (Service: AmazonEC2; Status Code: 400; Error Code: InvalidAMIID.NotFound; Request ID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)]. Please refer to Automation Service Troubleshooting Guide for more diagnosis details.`

Die `aws:runInstances`-Aktion erhält eine Eingabe für eine nicht vorhandene `ImageId`. Um dieses Problem zu beheben, aktualisieren Sie das Runbook oder Parameterwerte mit der richtigen AMI-ID.

Annahme, dass die Rollenrichtlinie über keine ausreichenden Berechtigungen verfügt

Fehlermeldung: `Automation Step Execution fails when it's launching the instance(s). Get Exception from RunInstances API of ec2 Service. Exception Message from RunInstances API: [You aren't authorized to perform this operation. Encoded authorization failure message: xxxxxxx (Service: AmazonEC2; Status Code: 403; Error Code: UnauthorizedOperation; Request ID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)]. Please refer to Automation Service Troubleshooting Guide for more diagnosis details.`

Die Übernahmerolle verfügt nicht über ausreichende Berechtigungen zum Aufrufen der `RunInstances`-API auf EC2-Instances. Um dieses Problem zu beheben, fügen Sie der Übernahmerolle eine IAM-Richtlinie hinzu, die über die Berechtigung zum Aufrufen der `RunInstances`-API verfügt. Weitere Informationen hierzu finden Sie unter Erstellen Sie die Servicerollen für Automation mithilfe der Konsole.

Unerwarteter Status

Fehlermeldung: `Step fails when it's verifying launched instance(s) are ready to be used. Instance i-xxxxxxxxx entered unexpected state: shutting-down. Please refer to Automation Service Troubleshooting Guide for more diagnosis details.`

- Mögliche Ursache 1: Es liegt ein Problem mit der Instance oder dem Amazon EC2-Service vor. Um dieses Problem zu beheben, melden Sie sich bei der Instance an oder überprüfen Sie das Instance-Systemprotokoll, um zu verstehen, warum die Instance mit dem Herunterfahren begonnen hat.
- Mögliche Ursache 2: Das angegebene Benutzerdatenskript für die `aws:runInstances`-Aktion weist ein Problem oder eine falsche Syntax auf. Überprüfen Sie die Syntax des Benutzerdatenskripts. Stellen Sie außerdem sicher, dass die Benutzerdatenskripts die Instance nicht herunterfahren oder andere Skripts aufrufen, die die Instance herunterfahren.

Aktionsspezifische Fehlerverweise

Sollte ein Schritt fehlschlagen, gibt die Fehlermeldung an, welcher Service aufgerufen wurde, als der Fehler aufgetreten ist. In der folgenden Tabelle sind die von der jeweiligen Aktion aufgerufenen Services aufgelistet. Die Tabelle enthält außerdem Links zu Informationen über jeden Service.

Aktion	AWS-Services von dieser Aktion aufgerufen	Weitere Informationen zu diesem Service	Inhalt der Fehlerbehebung
`aws:runInstances`	Amazon EC2	Amazon-EC2-Benutzerhandbuch	Fehlerbehebung bei EC2-Instances
`aws:changeInstanceState`	Amazon EC2	Amazon-EC2-Benutzerhandbuch	Fehlerbehebung bei EC2-Instances

Aktion	AWS-Services von dieser Aktion aufgerufen	Weitere Informationen zu diesem Service	Inhalt der Fehlerbehebung
`aws:runCommand`	Systems Manager	AWS Systems Manager Run Command	Fehlerbehebung von Systems Manager Run Command
`aws:createImage`	Amazon EC2	Amazon Machine Images	
`aws:createStack`	CloudFormation	AWS CloudFormation-Benutzerhandbuch	CloudFormation-Fehlerbehebung
`aws:deleteStack`	CloudFormation	AWS CloudFormation-Benutzerhandbuch	CloudFormation-Fehlerbehebung
`aws:deleteImage`	Amazon EC2	Amazon Machine Images	
`aws:copyImage`	Amazon EC2	Amazon Machine Images	
`aws:createTag`	Amazon EC2, Systems Manager	EC2-Ressourcen und -Tags	
`aws:invokeLambdaFunction`	AWS Lambda	AWS Lambda-Entwicklerhandbuch	Problembehebung bei Lambda

Interner Fehler bei Automation-Service

Fehlermeldung: `Internal Server Error. Please refer to Automation Service Troubleshooting Guide for more diagnosis details.`

Ein Problem mit dem Automation-Service verhindert, dass das angegebene Runbook korrekt ausgeführt wird. Um dieses Problem zu lösen, wenden Sie sich bitte an AWS -Support. Geben Sie die Ausführungs-ID und Kunden-ID an, wenn verfügbar.

Ausführung gestartet, aber mit Zeitüberschreitung

Fehlermeldung: `Step timed out while step is verifying launched instance(s) are ready to be used. Please refer to Automation Service Troubleshooting Guide for more diagnosis details.`

Für einen Schritt in der `aws:runInstances`-Aktion ist eine Zeitüberschreitung aufgetreten. Dies kann der Fall sein, wenn die Schrittaktion länger dauert als der angegebene Wert für `timeoutSeconds` im Schritt. Um dieses Problem zu lösen, geben Sie einen längeren Wert für den `timeoutSeconds`-Parameter in der `aws:runInstances`-Aktion an. Wenn das Problem dadurch nicht behoben werden kann, untersuchen Sie, warum der Schritt länger dauert als erwartet

AWS Systems Manager Change Calendar

Change Calendar, ein Tool in AWS Systems Manager, ermöglicht es Ihnen, Datums- und Zeitbereiche einzurichten, in denen von Ihnen angegebene Aktionen (z. B. in Systems Manager Automation-Runbooks) in Ihrem AWS-Konto ausgeführt werden können oder nicht. In Change Calendar werden diese Bereiche Ereignisse genannt. Wenn Sie einen Change Calendar-Eintrag erstellen, erstellen Sie ein Systems Manager-Dokument vom Typ `ChangeCalendar`. In Change Calendar wird das Dokument als iCalendar 2.0-Daten im Klartextformat gespeichert. Ereignisse, die Sie dem Change Calendar-Eintrag hinzufügen, werden Teil des Dokuments. Um mit Change Calendar zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Change Calendar aus.

Sie können einen Kalender und seine Ereignisse in der Systems Manager Konsole erstellen. Sie können auch eine iCalendar (`.ics`)-Datei importieren, die Sie von einem unterstützten Drittanbieter-Kalenderanbieter exportiert haben, um dessen Ereignisse Ihrem Kalender hinzuzufügen. Zu den unterstützten Anbietern zählen Google Kalender, Microsoft Outlook und iCloud-Kalender.

Ein Change Calendar-Eintrag kann zwei Typen haben:

`DEFAULT_OPEN` oder "Standardmäßig geöffnet"

Alle Aktionen können standardmäßig ausgeführt werden, außer während Kalenderereignissen. Während der Ereignisse lautet der Status eines `DEFAULT_OPEN`-Kalenders `CLOSED`. Die Ausführung von Ereignissen wird dann blockiert.

`DEFAULT_CLOSED` oder "Standardmäßig geschlossen"

Alle Aktionen werden standardmäßig blockiert, außer während Kalenderereignissen. Während der Ereignisse lautet der Status eines `DEFAULT_CLOSED`-Kalenders `OPEN`. Aktionen dürfen ausgeführt werden.

Sie können festlegen, dass alle geplanten Automatisierungsworkflows, Wartungsfenster und State Manager-Zuordnungen automatisch zu einem Kalender hinzugefügt werden. Sie können außerdem jeden dieser einzelnen Typen aus der Kalenderanzeige entfernen.

An wen richtet sich Change Calendar?

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

- AWS Kunden, die die folgenden Aktionstypen ausführen:
  - Erstellen Sie Automatisierungs-Runbooks oder führen Sie sie aus.
  - Erstellen Sie Änderungsanforderungen in Change Manager.
  - Führen Sie die Wartungsfenster.
  - Erstellen Sie Assoziationen in State Manager.

  Automatisierung, Change ManagerMaintenance Windows, und State Manager sind alles Tools AWS Systems Manager. Durch die Integration dieser Tools in Change Calendar, können Sie diese Aktionstypen je nach dem aktuellen Status des Änderungskalenders, den Sie mit jedem einzelnen verknüpfen, zulassen oder blockieren.
- Administratoren, die dafür verantwortlich sind, die Konfigurationen von verwalteten Systems-Manager-Knoten konsistent, stabil und funktionsfähig zu halten.

Vorteile von Change Calendar

Im Folgenden sind einige Vorteile von Change Calendar aufgeführt.

- Änderungen überprüfen, bevor sie angewendet werden

  Ein Change Calendar-Eintrag kann dazu beitragen, dass potenziell destruktive Änderungen in Ihrer Umgebung überprüft werden, bevor sie angewendet werden.

- Änderungen nur zu angemessenen Zeiten anwenden

  Change Calendar-Einträge helfen, Ihre Umgebung während der Ereigniszeiten stabil zu halten. Beispielsweise können Sie einen Change Calendar-Eintrag erstellen, um Änderungen zu blockieren, wenn Sie eine hohe Nachfrage nach Ihren Ressourcen erwarten wird (z. B. während einer Konferenz oder einer öffentlichen Marketingaktion). Ein Kalendereintrag kann auch Änderungen blockieren, wenn Sie eine eingeschränkte Administratorunterstützung erwarten, z. B. während eines Urlaubs oder einer Urlaubszeit. Sie können einen Kalendereintrag verwenden, um Änderungen außer zu bestimmten Tages- oder Wochenzeiten zuzulassen, zu denen nur begrenzter Administratorsupport zur Fehlerbehebung bei fehlgeschlagenen Aktionen oder Bereitstellungen zur Verfügung steht.

- Aktuellen oder bevorstehenden Status des Kalenders abrufen

  Sie können die Systems Manager `GetCalendarState`-API-Operation ausführen, um Ihnen den aktuellen Status des Kalenders, den Status zu einer bestimmten Zeit oder den nächsten geplanten Wechsel des Kalenderstatus anzuzeigen.

  > **Note**
  >
  > Die `GetCalendarState`-API hat ein Kontingent von 10 Anforderungen pro Sekunde. Informationen zu Systems-Manager-Kontingenten finden Sie unter Systems Manager Service Quotas in der Allgemeine Amazon Web Services-Referenz.

- EventBridge Unterstützung

  Dieses Systems Manager Manager-Tool wird in den EventBridge Amazon-Regeln als Ereignistyp unterstützt. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge und Referenz: EventBridge Amazon-Ereignistypen und -muster für Systems Manager.

Themen

- Einrichten von Change Calendar
- Arbeiten mit Change Calendar

- Hinzufügen von Change Calendar-Abhängigkeiten zu Automation-Runbooks
- Fehlerbehebung für Change Calendar

Einrichten von Change Calendar

Führen Sie die folgenden Schritte aus, bevor Sie Change Calendar, ein Tool in AWS Systems Manager, verwenden.

Installieren der neuesten Befehlszeilen-Tools

Installieren Sie die neuesten Befehlszeilen-Tools, um Statusinformationen über Kalender zu erhalten.

Anforderung	Beschreibung
AWS CLI	(Optional) Um die AWS Command Line Interface (AWS CLI) zu verwenden, um Statusinformationen über Kalender zu erhalten, installieren Sie die neueste Version der AWS CLI auf Ihrem lokalen Computer. Weitere Informationen zum Installieren oder Upgraden der CLI finden Sie unter Installieren, Aktualisieren und Deinstallieren der AWS CLI im AWS Command Line Interface-Benutzer handbuch.
AWS -Tools für PowerShell	(Optional) Um die Tools for PowerShell zu verwenden, um Statusinformationen über Kalender zu erhalten, installieren Sie die neueste Version der Tools for PowerShell auf Ihrem lokalen Computer. Weitere Informationen zum Installieren oder Upgraden von Tools for PowerShell finden Sie unter Installieren der AWS -Tools für PowerShell im AWS -Tools für PowerShell-Benutzerhandbuch.

Berechtigungen einrichten

Wenn Ihrem Benutzer, Ihrer Gruppe oder Rolle Administratorrechte zugewiesen sind, haben Sie vollen Zugriff auf Change Calendar. Wenn Sie nicht über Administratorrechte verfügen, muss Ihnen ein Administrator die Berechtigung erteilen, indem er entweder die von `AmazonSSMFullAccess` verwaltete Richtlinie zuweist oder eine Richtlinie, die Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle die erforderlichen Berechtigungen erteilt, zuweist.

Die folgenden Berechtigungen sind erforderlich, um mit Change Calendar zu arbeiten.

Change Calendar-Einträge

Um einen Change Calendar-Eintrag zu erstellen, zu aktualisieren oder zu löschen, einschließlich des Hinzufügens und Entfernens von Ereignissen aus dem Eintrag, muss eine Richtlinie, die Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle zugeordnet ist, die folgenden Aktionen erlauben:

- `ssm:CreateDocument`
- `ssm:DeleteDocument`
- `ssm:DescribeDocument`
- `ssm:DescribeDocumentPermission`
- `ssm:GetCalendar`
- `ssm:ListDocuments`
- `ssm:ModifyDocumentPermission`
- `ssm:PutCalendar`
- `ssm:UpdateDocument`
- `ssm:UpdateDocumentDefaultVersion`

Kalenderstatus

Um Informationen über den aktuellen oder bevorstehenden Status des Kalenders zu erhalten, muss eine Richtlinie, die Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle zugeordnet ist, die folgende Aktion erlauben:

- `ssm:GetCalendarState`

Betriebliche Ereignisse

Um betriebliche Ereignisse wie Wartungsfenster, Zuordnungen und geplante Automatisierungen anzuzeigen, muss die Richtlinie, die Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle zugeordnet ist, die folgenden Aktionen zulassen:

- `ssm:DescribeMaintenanceWindows`
- `ssm:DescribeMaintenanceWindowExecution`
- `ssm:DescribeAutomationExecutions`
- `ssm:ListAssociations`

> **Note**
>
> Change Calendar-Einträge, die anderen Konten als Ihrem gehören (d. h. von anderen erstellt wurden), sind schreibgeschützt, auch wenn sie mit Ihrem Konto geteilt werden. Wartungsfenster, State Manager-Verknüpfungen und Automatisierungen werden nicht gemeinsam genutzt.

Arbeiten mit Change Calendar

Mit der AWS Systems Manager-Konsole können Sie Einträge in Change Calendar, einem Tool in AWS Systems Manager, hinzufügen, verwalten oder löschen. Sie können Ereignisse auch von unterstützten Drittanbieter-Kalenderanbietern importieren, indem Sie eine iCalendar (`.ics`) -Datei importieren, die Sie aus dem Quellkalender exportiert haben. Und Sie können die `GetCalendarState`-API-Operation oder den `get-calendar-state` AWS Command Line Interface (AWS CLI)-Befehl verwenden, um Informationen über den Status von Change Calendar zu einem bestimmten Zeitpunkt zu erhalten.

Themen

- Erstellen eines Änderungskalenders
- Erstellen und Verwalten von Ereignissen in Change Calendar
- Importieren und Verwalten von Ereignissen aus Drittanbieter-Kalendern
- Aktualisieren eines Änderungskalenders
- Freigeben eines Änderungskalenders
- Einen Änderungskalender löschen
- Abrufen des Status eines Änderungskalenders

Erstellen eines Änderungskalenders

Wenn Sie einen Eintrag in Change Calendar erstellen, einem Tool in AWS Systems Manager, erstellen Sie ein Systems-Manager-Dokument (SSM-Dokument), das das `text`-Format verwendet.

So erstellen Sie einen Änderungskalender

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Wählen Sie Create calendar (Kalender erstellen).

   –oder–

   Wenn die Homepage von Change Calendar zuerst geöffnet wird, wählen Sie Create change calendar (Änderungskalender erstellen) aus.
4. Geben Sie auf der Seite Create calendar (Kalender erstellen) unter Calendar details (Kalenderdetails) einen Namen für Ihren Kalendereintrag ein. Namen von Kalendereinträgen können Buchstaben, Zahlen, Punkte, Striche und Unterstriche enthalten. Der Name sollte spezifisch genug sein, um den Zweck des Kalendereintrags auf einen Blick zu erkennen. Ein Beispiel ist **`support-off-hours`**. Sie können diesen Namen nicht mehr aktualisieren, nachdem Sie den Kalendereintrag erstellt haben.
5. (Optional) Geben Sie unter Description (Beschreibung) eine Beschreibung für Ihren Kalendereintrag ein.
6. (Optional) Klicken Sie im Bereich Importieren eines Kalanders auf Datei auswählen, um eine iCalendar (`.ics`)-Datei auszuwählen, die Sie von einem Drittanbieter-Kalenderanbieter exportiert haben. Beim Importieren der Datei werden die Ereignisse zu Ihrem Kalender hinzugefügt.

   Zu den unterstützten Anbietern zählen Google Kalender, Microsoft Outlook und iCloud-Kalender.

   Weitere Informationen finden Sie unter Importieren von Ereignissen von Drittanbietern-Kalenderanbietern.
7. Wählen Sie in Calendar type (Kalendertyp) eine der folgenden Optionen.

   - Open by default (Standardmäßig geöffnet) - der Kalender ist geöffnet (Automatisierungsaktionen können bis zum Start eines Ereignisses ausgeführt werden) und dann für die Dauer eines zugehörigen Ereignisses geschlossen.

- Closed by default (Standardmäßig geschlossen) - der Kalender ist geschlossen (Automatisierungsaktionen können erst ab dem Beginn eines Ereignisses ausgeführt werden), aber für die Dauer eines zugehörigen Ereignisses geöffnet.

8. (Optional) Wählen Sie unter Änderungsverwaltungsereignisse die Option Hinzufügen von Änderungsverwaltungsereignissen zum Kalender aus. Mit dieser Auswahl werden alle geplanten Wartungsfenster, State Manager-Verknüpfungen, Automatisierungsworkflows und Change Manager-Änderungsanforderungen in Ihrer monatlichen Kalenderanzeige angezeigt.

   > **Tip**
   >
   > Wenn Sie diese Ereignistypen später dauerhaft aus der Kalenderanzeige entfernen möchten, bearbeiten Sie den Kalender, deaktivieren Sie dieses Kontrollkästchen und wählen Sie dann Speichern.

9. Wählen Sie Create calendar (Kalender erstellen).

   Nachdem der Kalendereintrag erstellt wurde, zeigt Systems Manager Ihren Kalendereintrag in der Liste Change Calendar an. Die Spalten zeigen die Kalenderversion und die AWS-Konto-Kontonummer des Kalenderbesitzers an. Ihr Kalendereintrag kann keine Aktionen verhindern oder zulassen, bis Sie mindestens ein Ereignis erstellt oder importiert haben. Informationen zum Erstellen eines Ereignisses finden Sie unter Erstellen eines Change Calendar-Ereignisses. Weitere Informationen zum Importieren von Ereignissen finden Sie unter Importieren von Ereignissen von Drittanbietern-Kalenderanbietern.

Erstellen und Verwalten von Ereignissen in Change Calendar

Nachdem Sie einen Kalender in AWS Systems Manager Change Calendar erstellt haben, können Sie Ereignisse erstellen, aktualisieren und löschen, die in Ihrem geöffneten oder geschlossenen Kalender enthalten sind. Change Calendar ist ein Tool in AWS Systems Manager.

> **Tip**
>
> Alternativ zum Erstellen von Ereignissen direkt in der Systems Manager-Konsole können Sie eine iCalendar (`.ics`)-Datei aus einer unterstützten Kalenderanwendung eines Drittanbieters importieren. Weitere Informationen finden Sie unter Importieren und Verwalten von Ereignissen aus Drittanbieter-Kalendern.

Themen

- Erstellen eines Change Calendar-Ereignisses
- Aktualisieren eines Change Calendar-Ereignisses
- Löschen eines Change Calendar-Ereignisses

Erstellen eines Change Calendar-Ereignisses

Wenn Sie einem Eintrag in Change Calendar, einem Tool in AWS Systems Manager, ein Ereignis hinzufügen, geben Sie einen Zeitraum an, in dem die Standardaktion des Kalendereintrags ausgesetzt wird. Wenn der Kalendereintragstyp beispielsweise standardmäßig geschlossen ist, ist der Kalender für Änderungen während der Ereignisse geöffnet. (Alternativ können Sie ein empfohlenes Ereignis erstellen, das im Kalender nur der Information dient.)

Derzeit können Sie ein Change Calendar-Ereignis nur über die Konsole erstellen. Ereignisse werden zum Change Calendar-Dokument hinzugefügt, das Sie beim Erstellen eines Change Calendar-Eintrags erstellen.

Erstellen eines Change Calendar-Ereignisses

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Wählen Sie in der Liste der Kalender den Namen des Kalendereintrags aus, dem Sie ein Ereignis hinzufügen möchten.
4. Wählen Sie auf der Detailseite des Kalendereintrags Create event (Ereignis erstellen).
5. Geben Sie auf der Seite Create scheduled event (Geplantes Ereignis erstellen) unter Event details (Ereignisdetails) einen Anzeigenamen für Ihr Ereignis ein. Ereignisnamen können Buchstaben, Zahlen, Punkte, Bindestriche und Unterstriche enthalten. Der Name sollte spezifisch genug sein, um den Zweck des Ereignisses zu identifizieren. Ein Beispiel ist **`nighttime-hours`**.
6. Geben Sie unter Description (Beschreibung) eine Beschreibung für Ihr Ereignis ein. Beispiel, **`The support team isn't available during these hours`**.
7. (Optional) Wenn dieses Ereignis nur als visuelle Benachrichtigung oder Erinnerung dienen soll, aktivieren Sie das Kontrollkästchen Advisory (Empfehlung). Empfohlene Ereignisse haben im Kalender keine konkrete Funktion. Sie dienen nur zu Information für diejenigen, die den Kalender anzeigen.

8. Geben Sie unter Event start date (Startdatum des Ereignisses) einen Tag im Format `MM/DD/YYYY` ein oder wählen Sie einen Tag aus, an dem das Ereignis gestartet werden soll. Geben Sie außerdem eine Uhrzeit an dem angegebenen Tag im Format `hh:mm:ss` (Stunden, Minuten und Sekunden) ein, zu der das Ereignis starten soll.
9. Geben Sie unter Event end date (Enddatum des Ereignisses) einen Tag im Format `MM/DD/YYYY` ein oder wählen Sie einen Tag aus, an dem das Ereignis enden soll. Geben Sie außerdem eine Uhrzeit an dem angegebenen Tag im Format `hh:mm:ss` (Stunden, Minuten und Sekunden) ein, zu der das Ereignis enden soll.
10. Wählen Sie unter Schedule time zone (Zeitzone des Zeitplans) eine Zeitzone, die für die Start- und Endzeit des Ereignisses gilt. Sie können einen Teil des Stadtnamens oder den Zeitzonenunterschied zu Greenwich Mean Time (GMT) eingeben, um eine Zeitzone schneller zu finden. Die Standardeinstellung ist Coordinated Universal Time (UTC).
11. (Optional) Um ein täglich, wöchentlich oder monatlich wiederkehrendes Ereignis zu erstellen, aktivieren Sie Recurrence (Wiederholung). Geben Sie dann die Häufigkeit und das optionale Enddatum der Wiederholung an.
12. Wählen Sie Create scheduled event (Zeitgesteuertes Ereignis erstellen). Das neue Ereignis wird zu Ihrem Kalendereintrag hinzugefügt und auf der Registerkarte Events (Ereignisse) der Detailseite des Kalendereintrags angezeigt.

Aktualisieren eines Change Calendar-Ereignisses

Gehen Sie wie folgt vor, um ein Change Calendar-Ereignis in der AWS Systems Manager-Konsole zu aktualisieren. Change Calendar ist ein Tool in AWS Systems Manager.

Aktualisieren eines Change Calendar-Ereignisses

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Wählen Sie in der Liste der Kalender den Namen des Kalendereintrags, für den Sie ein Ereignis bearbeiten möchten.
4. Wählen Sie auf der Detailseite des Kalendereintrags Events (Ereignisse).
5. Wählen Sie auf der Kalenderseite das Ereignis, das Sie bearbeiten möchten.

> **Tip**
>
> Verwenden Sie die Schaltflächen oben links, um ein Jahr oder einen Monat zurück oder vorwärts zu gehen. Ändern Sie bei Bedarf die Zeitzone, indem Sie die richtige Zeitzone in der Liste oben rechts auswählen.

6. Wählen Sie unter Event details (Ereignisdetails) die Option Edit (Bearbeiten) aus.

   Um den Namen und die Beschreibung des Ereignisses zu ändern, ergänzen oder ersetzen Sie den aktuellen Text.

7. Um den Wert Event start date (Startdatum des Ereignisses) zu ändern, wählen Sie das aktuelle Startdatum und dann ein neues Datum im Kalender aus. Um die Startzeit zu ändern, wählen Sie die aktuelle Startzeit und dann eine neue Uhrzeit in der Liste aus.
8. Um den Wert Event end date (Enddatum des Ereignisses) zu ändern, wählen Sie das aktuelle Enddatum und dann ein neues Datum im Kalender aus. Um die Endzeit zu ändern, wählen Sie die aktuelle Endzeit und dann eine neue Uhrzeit in der Liste aus.
9. Um den Wert Schedule time zone (Zeitzone des Zeitplans) zu ändern, wählen Sie eine Zeitzone aus, die für die Start- und Endzeit des Ereignisses gelten soll. Sie können einen Teil des Stadtnamens oder den Zeitzonenunterschied zu Greenwich Mean Time (GMT) eingeben, um eine Zeitzone schneller zu finden. Die Standardeinstellung ist Coordinated Universal Time (UTC).
10. (Optional) Wenn dieses Ereignis nur als visuelle Benachrichtigung oder Erinnerung dienen soll, aktivieren Sie das Kontrollkästchen Advisory (Empfehlung). Empfohlene Ereignisse haben im Kalender keine konkrete Funktion. Sie dienen nur zu Information für diejenigen, die den Kalender anzeigen.
11. Wählen Sie Speichern. Ihre Änderungen werden auf der Registerkarte Events (Ereignisse) der Detailseite des Kalendereintrags angezeigt. Wählen Sie das Ereignis, das Sie aktualisiert haben, um Ihre Änderungen anzuzeigen.

Löschen eines Change Calendar-Ereignisses

Sie können jeweils ein Ereignis mithilfe der AWS-Managementkonsole in Change Calendar löschen, einem Tool in AWS Systems Manager.

> **Tip**
>
> Wenn Sie bei der Erstellung des Kalenders die Option Hinzufügen von Änderungsverwaltungsereignissen zum Kalender gewählt haben, können Sie Folgendes tun:
>
> - Um einen Typ eines Änderungsverwaltungsereignisses vorübergehend aus der Kalenderanzeige auszublenden, wählen Sie für den Typ das X oben in der Monatsvorschau.
> - Um diese Typen dauerhaft aus der Kalenderanzeige zu entfernen, bearbeiten Sie den Kalender, deaktivieren Sie das Kontrollkästchen Hinzufügen von Änderungsverwaltungsereignissen zum Kalender und wählen Sie dann Speichern. Wenn Sie Typen aus der Kalenderanzeige entfernen, werden sie nicht aus Ihrem Konto gelöscht.

Löschen eines Change Calendar-Ereignisses

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Wählen Sie in der Liste der Kalender den Namen des Kalendereintrags aus, aus dem Sie ein Ereignis löschen möchten.
4. Wählen Sie auf der Detailseite des Kalendereintrags Events (Ereignisse).
5. Wählen Sie auf der Kalenderseite das Ereignis, das Sie löschen möchten.

   > **Tip**
   >
   > Verwenden Sie die Schaltflächen oben links, um den Kalender um ein Jahr oder einen Monat zurück oder vorwärts zu verschieben. Ändern Sie bei Bedarf die Zeitzone, indem Sie die richtige Zeitzone in der Liste oben rechts auswählen.

6. Wählen Sie auf der Seite Event details (Ereignisdetails(die Option Delete (Löschen). Wenn Sie aufgefordert werden, das Löschen des Ereignisses zu bestätigen, wählen Sie Confirm (Bestätigen) aus.

Importieren und Verwalten von Ereignissen aus Drittanbieter-Kalendern

Alternativ zum Erstellen von Ereignissen direkt in der AWS Systems Manager-Konsole können Sie eine iCalendar (`.ics`)-Datei aus einer unterstützten Kalenderanwendung eines Drittanbieters importieren. Der Kalender kann sowohl importierte Ereignisse als auch Ereignisse enthalten, die Sie in Change Calendar erstellen (ein Tool von AWS Systems Manager).

Bevor Sie beginnen

Bevor Sie versuchen, eine Kalenderdatei zu importieren, überprüfen Sie die folgenden Anforderungen und Einschränkungen:

Kalenderdateiformat

Nur gültige iCalendar-Dateien (`.ics`) werden unterstützt.

Unterstützte Kalenderanbieter

Nur `.ics`-Dateien, die von den folgenden Drittanbieter-Kalenderanbietern exportiert wurden, werden unterstützt:

- Google Calendar (Exportanweisungen)
- Microsoft Outlook (Exportanweisungen)
- iCloud Calendar (Exportanweisungen)

Dateigröße

Sie können eine beliebige Anzahl gültiger `.ics`-Dateien importieren. Die Gesamtgröße aller importierten Dateien für jeden Kalender darf jedoch 64 KB nicht überschreiten.

> **Tip**
>
> Zum Minimieren der Größe der `.ics`-Datei, stellen Sie sicher, dass Sie nur grundlegende Details zu Ihren Kalendereinträgen exportieren. Verringern Sie bei Bedarf die Länge des Zeitraums, den Sie exportieren.

Zeitzone

Neben einem Kalendernamen, einem Kalenderanbieter und mindestens einem Ereignis sollte Ihre exportierte `.ics`-Datei außerdem die Zeitzone für den Kalender angeben. Wenn dies nicht der Fall ist oder ein Problem bei der Identifizierung der Zeitzone auftritt, werden Sie nach dem Importieren der Datei aufgefordert, eine anzugeben.

Wiederkehrende Ereigniseinschränkung

Ihre exportierte `.ics`-Datei kann wiederkehrende Ereignisse enthalten. Wenn jedoch ein oder mehrere Vorkommen eines wiederkehrenden Ereignisses im Quellkalender gelöscht wurden, schlägt der Import fehl.

Themen

- Importieren von Ereignissen von Drittanbietern-Kalenderanbietern
- Aktualisieren aller Ereignisse von einem Drittanbieter-Kalenderanbieter
- Löschen aller Ereignisse, die aus einem Kalender eines Drittanbieters importiert wurden

Importieren von Ereignissen von Drittanbietern-Kalenderanbietern

Gehen Sie wie folgt vor, um eine iCalendar (`.ics`)-Datei aus einer unterstützten Kalenderanwendung eines Drittanbieters zu importieren. Die in der Datei enthaltenen Ereignisse werden in die Regeln für Ihren offenen oder geschlossenen Kalender integriert. Sie können eine Datei in einen neuen Kalender importieren, den Sie mit Change Calendar erstellen (ein Tool in AWS Systems Manager), oder in einen vorhandenen Kalender.

Nach dem Importieren der `.ics`-Datei, können Sie einzelne Ereignisse aus dieser mit der Change Calendar-Schnittstelle entfernen. Weitere Informationen finden Sie unter Löschen eines Change Calendar-Ereignisses. Sie können auch alle Ereignisse aus dem Quellkalender löschen, indem Sie die `.ics`-Datei löschen. Weitere Informationen finden Sie unter Löschen aller Ereignisse, die aus einem Kalender eines Drittanbieters importiert wurden.

So importieren Sie Ereignisse von Drittanbietern-Kalenderanbietern

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Um mit einem neuen Kalender zu beginnen, wählen Sie Kalender erstellen. Wählen Sie im Bereich Kalender Importieren Datei auswählen. Informationen zu anderen Schritten zum Erstellen eines neuen Kalenders finden Sie unter Erstellen eines Änderungskalenders.

   –oder–

   Um Ereignisse von Drittanbietern in einen vorhandenen Kalender zu importieren, wählen Sie den Namen eines vorhandenen Kalenders aus, um diesen zu öffnen.

4. Wählen Sie Actions, Edit (Aktionen, Bearbeiten) und dann im Bereich Import calendar (Kalender importieren) die Option Choose file (Datei auswählen) aus.
5. Navigieren Sie zu der exportierten `.ics`-Datei auf Ihrem lokalen Computer und wählen Sie sie aus.
6. Wenn Sie dazu aufgefordert werden, wählen Sie Select a time zone (Zeitzone auswählen), um zu bestimmen, welche Zeitzone für den Kalender gelten soll.
7. Wählen Sie Speichern.

Aktualisieren aller Ereignisse von einem Drittanbieter-Kalenderanbieter

Wenn mehrere Ereignisse zu Ihrem Quellkalender hinzugefügt oder aus diesem entfernt werden, nachdem Sie dessen iCalendar `.ics`-Datei importiert haben, können Sie diese Änderungen in Change Calendar widerspiegeln. Exportieren Sie zunächst den Quellkalender neu und importieren Sie die neue Datei in Change Calendar, ein Tool in AWS Systems Manager. Ereignisse in Ihrem Änderungskalender werden aktualisiert, um den Inhalt der neueren Datei wiederzugeben.

So aktualisieren Sie alle Ereignisse eines Drittanbieter-Kalenderanbieters

1. Fügen Sie in Ihrem Kalender eines Drittanbieters Ereignisse hinzu oder entfernen Sie sie, wenn Sie möchten, dass sie in Change Calendar wiedergegeben werden, und exportieren Sie den Kalender dann erneut in eine neue `.ics`-Datei.
2. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
3. Wählen Sie im Navigationsbereich Change Calendar aus.
4. Wählen Sie aus der Liste der Kalender den Kalendernamen aus der Liste aus.
5. Wählen Sie Datei auswählen, navigieren Sie zu der `.ics`-Ersatzdatei und wählen Sie diese aus.
6. Als Reaktion auf die Benachrichtigung über das Überschreiben der vorhandenen Datei wählen Sie Confirm (Bestätigen).

Löschen aller Ereignisse, die aus einem Kalender eines Drittanbieters importiert wurden

Wenn Sie nicht mehr möchten, dass eines der Ereignisse, das Sie von einem Drittanbieter importiert haben, in Ihren Kalender aufgenommen wird, können Sie die importierte iCalendar `.ics`-Datei löschen.

So löschen Sie alle Ereignisse, die aus einem Kalender eines Drittanbieters importiert wurden

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Wählen Sie aus der Liste der Kalender den Kalendernamen aus der Liste aus.
4. Suchen Sie den Namen des importierten Kalenders im Bereich Kalender importieren, unter Meine importierten Kalender und wählen Sie die Schaltfläche X in seiner Registerkarte.
5. Wählen Sie Speichern.

Aktualisieren eines Änderungskalenders

Sie können die Beschreibung eines Änderungskalenders aktualisieren, aber nicht seinen Namen. Obwohl Sie den Standardstatus eines Kalenders ändern können, beachten Sie, dass dies das Verhalten von Änderungsaktionen bei Ereignissen, die mit dem Kalender verbunden sind, umkehrt. Wenn Sie z. B. den Status eines Kalenders von Standardmäßig geöffnet auf Standardmäßig geschlossen ändern, können unerwünschte Änderungen während der Ereigniszeiträume vorgenommen werden, wenn die Benutzer, die die verknüpften Ereignisse erstellt haben, keine Änderungen erwarten.

Wenn Sie einen Änderungskalender aktualisieren, bearbeiten Sie das Change Calendar-Dokument, das Sie beim Erstellen des Eintrags erstellt haben. Change Calendar ist ein Tool in AWS Systems Manager.

So aktualisieren Sie einen Änderungskalender

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Wählen Sie in der Liste der Kalender den Namen des Kalenders aus, den Sie aktualisieren möchten.
4. Wählen Sie auf der Detailseite des Kalenders Actions, Edit (Aktionen, Bearbeiten) aus.
5. In Description (Beschreibung) können Sie den Beschreibungstext ändern. Sie können den Namen eines Änderungskalenders nicht bearbeiten.
6. Um den Kalenderstatus zu ändern, wählen Sie in Calendar type (Kalendertyp) einen anderen Wert. Beachten Sie, dass dies das Verhalten von Änderungsaktionen bei Ereignissen, die mit

dem Kalender verbunden sind, umkehrt. Bevor Sie den Kalendertyp ändern, sollten Sie sich bei anderen Change Calendar-Benutzern vergewissern, dass die Änderung des Kalendertyps keine unerwünschten Änderungen während der von ihnen erstellten Ereignisse zulässt.

- Open by default (Standardmäßig geöffnet) - Der Kalender ist geöffnet (Automatisierungsaktionen können bis zum Start eines Ereignisses ausgeführt werden) und dann für die Dauer eines zugehörigen Ereignisses geschlossen.
- Closed by default (Standardmäßig geschlossen) - Der Kalender ist geschlossen (Automatisierungsaktionen können erst ab dem Beginn eines Ereignisses ausgeführt werden), aber für die Dauer eines zugehörigen Ereignisses geöffnet.

7. Wählen Sie Speichern.

   Ihr Kalender kann keine Aktionen verhindern oder zulassen, bis Sie mindestens ein Ereignis hinzufügen. Weitere Informationen zum Hinzufügen eines Ereignisses finden Sie unter Erstellen eines Change Calendar-Ereignisses.

Freigeben eines Änderungskalenders

Sie können einen Kalender in Change Calendar, ein Tool in AWS Systems Manager, über die AWS Systems Manager-Konsole für andere AWS-Konten freigeben. Wenn Sie einen Kalender freigeben, ist der Kalender für Benutzer im freigegebenen Konto schreibgeschützt. Wartungsfenster, State Manager-Verknüpfungen und Automatisierungen werden nicht gemeinsam genutzt.

So geben Sie einen Änderungskalender frei

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Wählen Sie in der Liste der Kalender den Namen des Kalenders aus, den Sie freigeben möchten.
4. Wählen Sie auf der Detailseite des Kalenders die Registerkarte Sharing (Freigabe) aus.
5. Wählen Sie Actions, Share (Aktionen, Freigeben) aus.
6. Geben Sie in Share calendar (Kalender freigeben) für Account ID (Kontonummer) die ID-Nummer eines gültigen AWS-Konto ein und wählen Sie dann Share (Freigeben).

   Benutzer des freigegebenen genutzten Kontos können den Änderungskalender lesen, aber keine Änderungen vornehmen.

Einen Änderungskalender löschen

Sie können einen Kalender in Change Calendar löschen, einem Tool in AWS Systems Manager, indem Sie entweder die Systems-Manager-Konsole oder die AWS Command Line Interface (AWS CLI) verwenden. Durch das Löschen eines Änderungskalenders werden alle zugehörigen Ereignisse gelöscht.

So löschen Sie einen Änderungskalender

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Calendar aus.
3. Wählen Sie in der Liste der Kalender den Namen des Kalenders aus, den Sie löschen möchten.
4. Wählen Sie auf der Detailseite des Kalenders Actions, Delete (Aktionen, Löschen) aus. Wenn Sie aufgefordert werden, zu bestätigen, dass Sie den Kalender löschen möchten, wählen Sie Delete (Löschen).

Abrufen des Status eines Änderungskalenders

Sie können den Gesamtzustand eines Kalenders oder den Zustand eines Kalenders zu einem bestimmten Zeitpunkt in Change Calendar abrufen, einem Tool in AWS Systems Manager. Sie können auch den nächsten Zeitpunkt anzeigen, an dem der Kalenderzustand von `OPEN` auf `CLOSED` oder umgekehrt wechselt.

> **Note**
>
> Informationen zur Integration von Change Calendar in Amazon EventBridge zur automatisierten Überwachung von Kalenderstatusänderungen finden Sie unter the section called “Change CalendarIntegration mit Amazon EventBridge”. Die EventBridge-Integration bietet ereignisgesteuerte Benachrichtigungen, wenn sich Kalenderstatus ändern, und ergänzt so den abfragebasierten Ansatz der API-Aktion `GetCalendarState`.

Diese Aufgabe können Sie nur mit der `GetCalendarState`-API-Operation ausführen. Die Prozedur in diesem Abschnitt verwendet die AWS Command Line Interface (AWS CLI).

So rufen Sie den Status eines Änderungskalenders ab

- Führen Sie den folgenden Befehl aus, um den Status eines oder mehrerer Kalender zu einer bestimmten Zeit anzuzeigen. Der Parameter `--calendar-names` ist erforderlich, `--at-time` ist jedoch optional. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

  Linux & macOS

  ```
  aws ssm get-calendar-state \
      --calendar-names "Calendar_name_or_document_ARN_1"
   "Calendar_name_or_document_ARN_2" \
      --at-time "ISO_8601_time_format"
  ```

  Im Folgenden wird ein Beispiel gezeigt.

  ```
  aws ssm get-calendar-state \
      --calendar-names "arn:aws:ssm:us-east-2:123456789012:document/
  MyChangeCalendarDocument" "arn:aws:ssm:us-east-2:123456789012:document/
  SupportOffHours" \
      --at-time "2020-07-30T11:05:14-0700"
  ```

  Windows

  ```
  aws ssm get-calendar-state ^
      --calendar-names "Calendar_name_or_document_ARN_1"
   "Calendar_name_or_document_ARN_2" ^
      --at-time "ISO_8601_time_format"
  ```

  Im Folgenden wird ein Beispiel gezeigt.

  ```
  aws ssm get-calendar-state ^
      --calendar-names "arn:aws:ssm:us-east-2:123456789012:document/
  MyChangeCalendarDocument" "arn:aws:ssm:us-east-2:123456789012:document/
  SupportOffHours" ^
      --at-time "2020-07-30T11:05:14-0700"
  ```

  Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "State": "OPEN",
    "AtTime": "2020-07-30T16:18:18Z",
    "NextTransitionTime": "2020-07-31T00:00:00Z"
}
```

Die Ergebnisse zeigen den Status des Kalenders (unabhängig davon, ob der Kalender vom Typ `DEFAULT_OPEN` oder `DEFAULT_CLOSED` ist) für die angegebenen Kalendereinträge, die Ihrem Konto gehören oder für dieses freigegeben sind, zu dem Zeitpunkt an, der als Wert von `--at-time` angegeben ist, sowie den Zeitpunkt des nächsten Übergangs. Wenn Sie den Parameter `--at-time` nicht hinzufügen, wird die aktuelle Zeit verwendet.

> **Note**
>
> Wenn Sie mehr als einen Kalender in einer Anforderung angeben, gibt der Befehl den Status von `OPEN` nur, wenn alle Kalender in der Anforderung geöffnet sind. Wenn ein oder mehrere Kalender in der Anforderung geschlossen sind, lautet der zurückgegebene Status `CLOSED`.

Hinzufügen von Change Calendar-Abhängigkeiten zu Automation-Runbooks

Damit Automatisierungsaktionen eingehalten werden Change Calendar AWS Systems Manager, fügen Sie einem Tool in einem Automatisierungs-Runbook einen Schritt hinzu, der die `aws:assertAwsResourceProperty`Aktion verwendet. Konfigurieren Sie die Aktion zur Ausführung von `GetCalendarState`, um zu überprüfen, ob sich ein bestimmter Kalendereintrag in dem gewünschten Zustand befindet (`OPEN` oder `CLOSED`). Das Automation-Runbook darf nur dann mit dem nächsten Schritt fortfahren, wenn der Kalenderstatus `OPEN` ist. Im Folgenden wird ein YAML-basierter Beispielausschnitt eines Automation-Runbooks gezeigt, das nicht zum nächsten Schritt `LaunchInstance` weitergehen kann, es sei denn, der Kalenderstatus entspricht `OPEN` (dem in `DesiredValues` festgelegten Status).

Im Folgenden wird ein -Beispiel gezeigt.

```
mainSteps:
  - name: MyCheckCalendarStateStep
    action: 'aws:assertAwsResourceProperty'
```

```
    inputs:
      Service: ssm
      Api: GetCalendarState
      CalendarNames: ["arn:aws:ssm:us-east-2:123456789012:document/SaleDays"]
      PropertySelector: '$.State'
      DesiredValues:
      - OPEN
    description: "Use GetCalendarState to determine whether a calendar is open or
 closed."
    nextStep: LaunchInstance
  - name: LaunchInstance
    action: 'aws:executeScript'
    inputs:
      Runtime: python3.11
...
```

Fehlerbehebung für Change Calendar

Im Folgenden finden Sie Informationen zur Behandlung von Problemen mit Change Calendar, einem Tool in AWS Systems Manager.

Themen

- Fehler ‚Calendar import failed' (Importieren des Kalenders fehlgeschlagen)

Fehler ‚Calendar import failed' (Importieren des Kalenders fehlgeschlagen)

Problem: Beim Importieren einer iCalendar (`.ics`)-Datei meldet das System, dass der Kalenderimport fehlgeschlagen ist.

- Lösung 1— Stellen Sie sicher, dass Sie eine Datei importieren, die von einem unterstützten Drittanbieter-Kalenderanbieter exportiert wurde. Unterstütze Anbieter sind:
  - Google Calendar (Exportanweisungen)
  - Microsoft Outlook (Exportanweisungen)
  - iCloud Calendar (Exportanweisungen)
- Lösung 2— Wenn der Quellkalender wiederkehrende Ereignisse enthält, stellen Sie sicher, dass keine einzelnen Ereignisse des Ereignisses abgebrochen oder gelöscht wurden. Derzeit unterstützt Change Calendar das Importieren von wiederkehrenden Ereignissen mit individuellen Stornierungen nicht. Um das Problem zu beheben, entfernen Sie das wiederkehrende Ereignis aus dem Quellkalender, exportieren Sie den Kalender erneut und importieren Sie ihn erneut in

Change Calendar und fügen Sie dann das wiederkehrende Ereignis mithilfe der Change Calendar-Schnittstelle hinzu. Weitere Informationen finden Sie unter Erstellen eines Change Calendar-Ereignisses.

- Lösung 3 – Stellen Sie sicher, dass der Quellkalender mindestens ein Ereignis enthält. Uploads von `.ics`-Dateien, die keine Ereignisse enthalten, werden fehlschlagen.
- Lösungs 4 – Wenn das System meldet, dass der Import fehlgeschlagen ist, weil die `.ics`-Datei zu groß ist, stellen Sie sicher, dass Sie nur grundlegende Details zu Ihren Kalendereinträgen exportieren. Verringern Sie bei Bedarf die Länge des Zeitraums, den Sie exportieren.
- Lösung 5 – Wenn Change Calendar die Zeitzone des exportierten Kalenders nicht ermitteln kann, wenn Sie versuchen, ihn aus der Registerkarte Ereignisse zu importieren, wird möglicherweise die folgende Meldung angezeigt: „Der Import des Kalenders ist fehlgeschlagen“ (Calendar import failed). Change Calendar konnte keine gültige Zeitzone finden. Sie können den Kalender aus dem Menü Bearbeiten importieren.“ Wählen Sie in diesem Fall die Option Actions, Edit (Aktionen, bearbeiten) und versuchen Sie dann, die Datei von der Seite Edit calendar (Kalender bearbeiten) zu importieren.
- Lösungs 6 – Bearbeiten Sie die `.ics`-Datei nicht vor dem Import. Der Versuch, den Inhalt der Datei zu ändern, kann die Kalenderdaten beschädigen. Wenn Sie die Datei vor dem Import verändert haben, exportieren Sie den Kalender erneut aus dem Quellkalender, und führen Sie einen erneuten Upload durch.

AWS Systems Manager Change Manager

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Change Manager, ein Tool in AWS Systems Manager, ist ein Change-Management-Framework für Unternehmen, mit dem betriebliche Änderungen an Ihrer Anwendungskonfiguration und -infrastruktur angefordert, genehmigt, implementiert und gemeldet werden können. Wenn Sie ein einziges delegiertes Administratorkonto verwenden AWS Organizations, können Sie Änderungen an mehreren Stellen AWS-Konten und übergreifend verwalten. AWS-Regionen Alternativ können Sie mit

einem lokalen Konto Änderungen für einen einzigen AWS-Konto verwalten. Wird Change Manager für die Verwaltung von Änderungen sowohl an AWS Ressourcen als auch an lokalen Ressourcen verwendet. Um mit Change Manager zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Change Manager aus.

Mit Change Manager können Sie vorab genehmigte Änderungsvorlagen verwenden, um Änderungsprozesse für Ihre Ressourcen zu automatisieren und unbeabsichtigte Ergebnisse bei betrieblichen Änderungen zu vermeiden. Jede Änderungsvorlage gibt Folgendes an:

- Eine oder mehrere Automation-Runbooks, aus denen ein Benutzer beim Erstellen einer Änderungsanforderung auswählen kann. Die Änderungen an Ihren Ressourcen werden in Automation-Runbooks definiert. Sie können benutzerdefinierte Runbooks oder AWS -verwaltete Runbooks in die von Ihnen erstellten Änderungsvorlagen aufnehmen. Wenn ein Benutzer einen Änderungsantrag erstellt, kann er auswählen, welches der verfügbaren Runbooks in die Anforderung aufgenommen werden soll. Darüber hinaus können Sie Änderungsvorlagen erstellen, mit denen der Benutzer, der die Anforderung stellt, jedes beliebige Runbook in der Änderungsanforderung angeben kann.
- Die Benutzer im Konto, die Änderungsanforderungen überprüfen müssen, die mit dieser Änderungsvorlage vorgenommen wurden.
- Das Amazon Simple Notification Service (Amazon SNS)-Thema, das verwendet wird, um zugewiesene Genehmiger darüber zu informieren, dass ein Änderungsantrag zur Überprüfung bereit ist.
- Der CloudWatch Amazon-Alarm, der zur Überwachung des Runbook-Workflows verwendet wird.
- Das Amazon SNS-Thema, das verwendet wird, um Benachrichtigungen über Statusänderungen für Änderungsanforderungen zu senden, die mit der Änderungsvorlage erstellt werden.
- Die Tags, die auf die Änderungsvorlage angewendet werden sollen, um die Änderungsvorlagen zu kategorisieren und zu filtern.
- Ob aus der Änderungsvorlage erstellte Änderungsanträge ohne Genehmigungsschritt ausgeführt werden können (automatisch genehmigte Anforderungen).

Durch die Integration mitChange Calendar, einem weiteren Tool in Systems Manager, können Sie Änderungen Change Manager auch sicher implementieren und gleichzeitig Zeitplankonflikte bei wichtigen Geschäftsereignissen vermeiden. Change ManagerIntegration mit AWS Organizations und Unterstützung AWS IAM Identity Center bei der Verwaltung von Änderungen in Ihrem gesamten Unternehmen von einem einzigen Konto aus unter Verwendung Ihres vorhandenen Identitätsmanagementsystems. Sie können den Änderungsfortschritt von Change Manager

überwachen und betriebliche Änderungen in Ihrer gesamten Organisation prüfen und für bessere Transparenz und Rechenschaftspflicht sorgen.

Change Managerergänzt die Sicherheitskontrollen Ihrer Verfahren zur kontinuierlichen Integration (CI) und der Methodik zur kontinuierlichen Bereitstellung (CD). Change Managerist nicht für Änderungen vorgesehen, die im Rahmen eines automatisierten Release-Prozesses, wie z. B. einer CI/CD Pipeline, vorgenommen werden, es sei denn, es ist eine Ausnahme oder eine Genehmigung erforderlich.

Funktionsweise von Change Manager

Wenn eine Standard- oder Notfalländerung benötigt wird, erstellt jemand in der Organisation einen Änderungsantrag, der auf einer der Änderungsvorlagen basiert, die für die Verwendung in Ihrer Organisation oder Ihrem Konto erstellt wurden.

Wenn die angeforderte Änderung manuelle Genehmigungen erfordert, benachrichtigt Change Manager die angegebenen Genehmiger durch eine Amazon SNS Benachrichtigung, dass ein Änderungsantrag zur Überprüfung bereit ist. Sie können Genehmiger für Änderungsanforderungen in der Änderungsvorlage benennen oder Benutzer Genehmiger in der Änderungsanforderung selbst benennen lassen. Sie können verschiedenen Vorlagen verschiedene Prüfer zuweisen. Ordnen Sie beispielsweise einen Benutzer, eine Benutzergruppe oder eine AWS Identity and Access Management (IAM)-Rolle zu, die Anforderungen für Änderungen an verwalteten Knoten genehmigen muss, und eine andere Benutzer-, Gruppen- oder IAM-Rolle für Datenbankänderungen. Wenn die Änderungsvorlage automatische Genehmigungen zulässt und die Benutzerrichtlinie eines Anforderers dies nicht verbietet, kann der Benutzer das Automation-Runbook für seine Anfrage auch ohne Überprüfungsschritt ausführen (mit Ausnahme von Ereignissen zum Einfrieren von Änderungen).

Für jede Änderungsvorlage können Sie bis zu fünf Genehmigerebenen hinzufügen. Sie können beispielsweise verlangen, dass technische Prüfer eine Änderungsanforderung, die aus einer Änderungsvorlage erstellt wurde, zuerst genehmigen und dann eine zweite Genehmigungsebene von einem oder mehreren Managern anfordern.

Change Manager ist in AWS Systems Manager Change Calendar integriert. Wenn eine angeforderte Änderung genehmigt wird, ermittelt das System zunächst, ob die Anforderung mit anderen geplanten Geschäftsvorgängen in Konflikt steht. Wenn ein Konflikt erkannt wird, kann Change Manager die Änderung blockieren oder zusätzliche Genehmigungen erfordern, bevor Sie den Runbook-Workflow starten. Beispielsweise können Sie Änderungen nur während der Geschäftszeiten erlauben, um

sicherzustellen, dass Teams zur Verfügung stehen, um unerwartete Probleme zu verwalten. Für alle Änderungen, die außerhalb dieser Zeiten ausgeführt werden sollen, können Sie eine Genehmigung für die Verwaltung auf höherer Ebene in Form vonChange–Freeze-Genehmigern fordern. Bei Notfalländerungen kann Change Manager den Schritt der Überprüfung von Change Calendar für Konflikte oder Blockieren von Ereignissen überspringen, nachdem eine Änderungsanforderung genehmigt wurde.

Wenn es an der Zeit ist, eine genehmigte Änderung zu implementieren Change Manager führt das Automation-Runbook aus, das in der zugeordneten Änderungsanforderung angegeben ist. Nur die Vorgänge, die in genehmigten Änderungsanforderungen definiert sind, sind zulässig, wenn Runbook-Workflows ausgeführt werden. Dieser Ansatz hilft Ihnen, unbeabsichtigte Ergebnisse zu vermeiden, während Änderungen implementiert werden.

Zusätzlich zum Einschränken der Änderungen, die bei der Ausführung eines Runbook-Workflows vorgenommen werden können, hilft Ihnen Change Manager außerdem beim Steuern der Gleichzeitigkeit und der Fehlerschwellenwerte. Sie legen fest, wie viele Ressourcen ein Runbook-Workflow gleichzeitig ausführen kann, auf wie vielen Konten die Änderung gleichzeitig ausgeführt werden kann und wie viele Fehler vor dem Beenden des Prozesses zugelassen werden und (wenn das Runbook ein Rollback-Skript enthält) zurückgesetzt werden sollen. Sie können den Fortschritt der vorgenommenen Änderungen auch mithilfe von CloudWatch Alarmen überwachen.

Nachdem ein Runbook-Workflow abgeschlossen wurde, können Sie Details zu den vorgenommenen Änderungen überprüfen. Diese Details beinhalten den Grund für einen Änderungsantrag, welche Änderungsvorlage verwendet wurde, wer die Änderungen angefordert und genehmigt hat und wie die Änderungen implementiert wurden.

Weitere Informationen

Einführung von AWS Systems ManagerChange Manager auf dem AWS -News Blog

Welche Vorteile bietet Change Manager meinen Vorgängen?

Change Manager bietet folgende Vorteile:

- Reduzieren Sie das Risiko von Service-Unterbrechungen und Ausfallzeiten

  Change Manager kann Betriebsänderungen sicherer machen, indem sichergestellt wird, dass nur genehmigte Änderungen implementiert werden, wenn ein Runbook-Workflow ausgeführt wird. Sie können ungeplante und nicht überprüfte Änderungen blockieren. Change Manager hilft Ihnen, die

Arten von unbeabsichtigten Ergebnissen zu vermeiden, die durch menschliche Fehler verursacht werden, die kostspielige Zeit an Forschung und Rückverfolgung erfordern.

- Detailliertes Prüfung und Berichterstattung zu Änderungshistorien

  Change Manager bietet Rechenschaftspflicht mit einer konsistenten Möglichkeit, Änderungen, die in Ihrem Unternehmen vorgenommen wurden, zu melden und zu prüfen, die Absicht der Änderungen und Details darüber, wer sie genehmigt und implementiert hat.

- Vermeiden Sie Konflikte oder Verstöße

  Change Manager kann Zeitplankonflikte wie Feiertagsereignisse oder neue Produktstarts basierend auf dem aktiven Änderungskalender für Ihre Organisation erkennen. Sie können die Ausführung von Runbook-Workflows nur während der Geschäftszeiten oder nur mit zusätzlichen Genehmigungen erlauben.

- Anpassung der Änderungsanforderungen an Ihr sich änderndes Geschäft

  In verschiedenen Geschäftsperioden können Sie unterschiedliche Anforderungen an das Änderungsmanagement stellen. Beispielsweise können Sie während der end-of-month Berichterstattung, in der Steuersaison oder in anderen kritischen Geschäftsperioden Änderungen blockieren oder für Änderungen, die unnötige betriebliche Risiken mit sich bringen könnten, die Genehmigung der Geschäftsleitung einholen.

- Zentrale Verwaltung von Änderungen über Konten hinweg

  Durch die Integration mit Organizations können Sie Änderungen in all Ihren Organisationseinheiten (OUs) von einem einzigen delegierten Administratorkonto aus verwalten. Change Manager Sie können es Change Manager für die Verwendung mit Ihrer gesamten Organisation oder nur für einige Ihrer OUs Mitarbeiter aktivieren.

An wen richtet sich Change Manager?

Change Managerist für die folgenden AWS Kunden und Organisationen geeignet:

- Jeder AWS Kunde, der die Sicherheit und Steuerung betrieblicher Änderungen an seinen Cloud- oder lokalen Umgebungen verbessern möchte.
- Organisationen, die die Zusammenarbeit und Transparenz über alle Teams hinweg verbessern, die Anwendungsverfügbarkeit durch Vermeidung von Ausfallzeiten verbessern und das mit manuellen und sich wiederholenden Aufgaben verbundene Risiko verringern möchten.
- Organisationen, die bewährte Methoden für das Änderungsmanagement einhalten müssen.

- Kunden, die eine vollständig überprüfbare Historie der Änderungen an ihrer Anwendungskonfiguration oder -infrastruktur benötigen.

Was sind die Hauptfeatures von Change Manager?

Zu den wichtigsten Features von Change Manager gehört Folgendes:

- Integrierte Unterstützung für bewährte Methoden für das Änderungsmanagement

  Mit Change Manager können Sie ausgewählte bewährte Methoden für das Änderungsmanagement auf Ihre Vorgänge anwenden. Sie können folgende Optionen aktivieren:

  - Überprüfen Sie Change Calendar, um zu sehen, ob Ereignisse derzeit eingeschränkt sind, sodass Änderungen nur während der offenen Kalenderperioden vorgenommen werden.
  - Zulassen von Änderungen bei eingeschränkten Ereignissen mit zusätzlichen Genehmigungen von Change-Freeze-Genehmigungsberechtigten.
  - Erfordern, dass CloudWatch Alarme für alle Änderungsvorlagen angegeben werden.
  - Verlangen Sie, dass alle in Ihrem Konto erstellten Änderungsvorlagen geprüft und genehmigt werden müssen, bevor sie zur Erstellung von Änderungsaufträgen verwendet werden können.
- Verschiede Genehmigungspfade für geschlossene Kalenderperioden und Notänderungsanträge

  Sie können eine Option zulassen, um Change Calendar auf eingeschränkte Ereignisse zu prüfen und genehmigte Änderungsanforderungen zu blockieren, bis das Ereignis abgeschlossen ist. Sie können jedoch auch eine zweite Gruppe von Genehmigern bestimmen, die Change-Freeze-Genehmiger, die eine Änderung auch dann zulassen können, wenn der Kalender geschlossen ist. Sie können auch Notfalländerungsvorlagen erstellen. Änderungsaufträge, die aus einer Notfalländerungsvorlage erstellt wurden, erfordern weiterhin regelmäßige Genehmigungen, unterliegen jedoch keinen Kalenderbeschränkungen und erfordern keine Change-Freeze-Freigaben.
- Steuern Sie, wie und wann Runbook-Workflows gestartet werden

  Runbook-Workflows können nach einem Zeitplan oder nach Abschluss der Genehmigungen gestartet werden (vorbehaltlich der Kalendereinschränkungsregeln).
- Integrierte Unterstützung für Benachrichtigungen

  Geben Sie an, wer in Ihrer Organisation Änderungsvorlagen und Änderungsanforderungen prüfen und genehmigen soll. Weisen Sie einer Änderungsvorlage ein Amazon SNS-Thema zu, um

Benachrichtigungen an die Abonnenten des Themas über Statusänderungen für Änderungsanträge zu senden, die mit dieser Änderungsvorlage erstellt wurden.

- Integration mit AWS Systems Manager Change Calendar

  Change Manager ermöglicht es Administratoren, Zeitplanänderungen während bestimmter Zeiträume einzuschränken. Sie können beispielsweise eine Richtlinie erstellen, die Änderungen nur während der Geschäftszeiten zulässt, um sicherzustellen, dass das Team für Probleme verfügbar ist. Sie können Änderungen auch bei wichtigen Geschäftsereignissen einschränken. Beispielsweise können Einzelhandelsunternehmen Änderungen bei großen Verkaufsereignissen einschränken. Sie können auch während eingeschränkter Zeiträume zusätzliche Genehmigungen verlangen.

- Integration mit AWS IAM Identity Center und Active Directory-Unterstützung

  Mit der IAM-Identity-Center-Integration können Mitglieder Ihrer Organisation auf AWS-Konten zugreifen und ihre Ressourcen mithilfe von Systems Manager basierend auf einer gemeinsamen Benutzeridentität verwalten. Mit IAM Identity Center können Sie Ihren Benutzern Zugriff auf Konten über AWS hinweg gewähren.

  Durch die Integration mit Active Directory können Benutzer in Ihrem Active Directory-Konto als Genehmiger für Änderungsvorlagen zugewiesen werden, die für Ihre Change Manager-Vorgänge erstellt wurden.

- Integration mit CloudWatch Amazon-Alarmen

  Change Managerist in CloudWatch Alarme integriert. Change Managerlauscht während des Runbook-Workflows auf CloudWatch Alarme und ergreift alle Aktionen, einschließlich des Sendens von Benachrichtigungen, die für den Alarm definiert sind.

- Integration mit Lake AWS CloudTrail

  Durch die Einrichtung eines Ereignisdatenspeichers in AWS CloudTrail Lake können Sie überprüfbare Informationen zu den Änderungen einsehen, die durch Änderungsanforderungen vorgenommen wurden, die in Ihrem Konto oder Ihrer Organisation ausgeführt werden. Die gespeicherten Ereignisinformationen enthalten u. a. folgende Details:

  - Die API-Aktionen, die ausgeführt wurden
  - Die für diese Aktionen enthaltenen Anforderungsparameter
  - Der Benutzer, der die Aktion ausgeführt hat
  - Die Ressourcen, die während des Vorgangs aktualisiert wurden

- Integration mit AWS Organizations

  Mithilfe der von Organizations bereitgestellten kontenübergreifenden Funktionen können Sie ein delegiertes Administratorkonto für die Verwaltung von Change Manager Vorgängen OUs in Ihrer Organisation verwenden. In Ihrem Organizations-Verwaltungskonto können Sie angeben, welches Konto das delegierte Administratorkonto sein soll. Sie können auch steuern, welche von Ihnen verwendet werden OUs Change Manager können.

Entstehen Kosten für die Verwendung von Change Manager?

Ja. Change Managerwird auf einer bestimmten pay-per-use Basis berechnet. Sie zahlen nur das, was Sie nutzen. Weitere Informationen finden Sie unter AWS Systems Manager – Preise.

Was sind die primären Komponenten von Change Manager?

Change Manager-Komponenten, die Sie zum Verwalten des Änderungsprozesses in Ihrer Organisation oder Ihrem Konto verwenden, umfassen Folgendes:

Delegiertes Administratorkonto

Wenn Sie den Change Manager in einer Organisation verwenden, verwenden Sie ein delegiertes Administratorkonto. Dies ist der AWS-Konto , der als Konto für die Verwaltung von Betriebsaktivitäten in Systems Manager festgelegt ist, einschließlich Change Manager. Das delegierte Administratorkonto verwaltet Änderungsaktivitäten in Ihrer gesamten Organisation. Wenn Sie Ihre Organisation für die Verwendung mit dem Change Manager einrichten, geben Sie an, welche Ihrer Konten in dieser Rolle verwendet werden. Das delegierte Administratorkonto muss das einzige Mitglied der Organisationseinheit (OU) sein, der es zugewiesen ist. Das delegierte Administratorkonto ist nicht erforderlich, wenn Sie es AWS-Konto nur Change Manager mit einem Konto verwenden.

> **Important**
>
> Wenn Sie den Change Manager in einer Organisation verwenden, empfehlen wir, Änderungen immer über das delegierte Administratorkonto vorzunehmen. Obwohl Sie Änderungen von anderen Konten in der Organisation vornehmen, werden diese Änderungen nicht im delegierten Administratorkonto gemeldet oder können nicht angezeigt werden.

Änderungsvorlage

Eine Änderungsvorlage ist eine Sammlung von Konfigurationseinstellungen in Change Manager, die beispielsweise erforderliche Genehmigungen, verfügbare Runbooks und Benachrichtigungsoptionen für Änderungsanforderungen definiert.

Sie können verlangen, dass die von Benutzern in Ihrer Organisation oder Ihrem Konto erstellten Änderungsvorlagen einen Genehmigungsprozess durchlaufen, bevor sie verwendet werden können.

Change Manager unterstützt zwei Arten von Änderungsvorlagen. Bei einer genehmigten Änderungsanforderung, die auf einer Notfalländerungsvorlage basiert, kann die angeforderte Änderung auch dann vorgenommen werden, wenn Sperrereignisse in Change Calendar vorliegen. Bei einer genehmigten Änderungsanforderung, die auf einer standardmäßigen Änderungsvorlage basiert, kann die angeforderte Änderung nicht vorgenommen werden, wenn Blockierungsereignisse in Change Calendar vorhanden sind, es sei denn, es werden zusätzliche Genehmigungen von bestimmten Genehmigern für Change-Freeze-Ereignisse erhalten.

Änderungsanforderung

Eine Änderungsanforderung ist eine Anforderung Change Manager zur Ausführung eines Automatisierungs-Runbooks, das eine oder mehrere Ressourcen in Ihren AWS oder lokalen Umgebungen aktualisiert. Ein Änderungsantrag wird mit einer Änderungsvorlage erstellt.

Wenn Sie eine Änderungsanforderung erstellen, müssen ein oder mehrere Genehmiger in Ihrer Organisation oder Ihrem Konto die Anforderung überprüfen und genehmigen. Ohne die erforderlichen Genehmigungen kann der Runbook-Workflow, der die angeforderten Änderungen anwendet, nicht ausgeführt werden.

Im System sind Änderungsanforderungen eine Art von OpsItem Eingabe. AWS Systems Manager OpsCenter Jedoch werden OpsItems des Typs `/aws/changerequest` nicht in OpsCenter angezeigt. Als OpsItems unterliegen Änderungsanforderungen den gleichen erzwungenen Kontingenten wie andere OpsItems-Typen.

Um eine Änderungsanforderung programmgesteuert zu erstellen, rufen Sie außerdem nicht die `CreateOpsItem`-API-Operation auf. Verwenden Sie anstelle die `StartChangeRequestExecution`-API-Operation. Anstatt sofort ausgeführt zu werden, muss die Änderungsanforderung genehmigt werden, und es dürfen keine blockierenden Ereignisse in Change Calendar vorliegen, um zu verhindern, dass der Workflow ausgeführt wird. Wenn Genehmigungen empfangen wurden und der Kalender nicht gesperrt ist (oder die Berechtigung erteilt wurde,

blockierende Kalenderereignisse zu umgehen), kann die `StartChangeRequestExecution`-Aktion abgeschlossen werden.

Runbook-Workflow

Ein Runbook-Workflow ist der Prozess der angeforderten Änderungen, die an den Zielressourcen in Ihrer Cloud oder On-Premises-Umgebung vorgenommen werden. Jede Änderungsanforderung bestimmt ein einziges Automation-Runbook, das zur Durchführung der angeforderten Änderung verwendet werden soll. Der Runbook-Workflow tritt auf, nachdem alle erforderlichen Genehmigungen erteilt wurden und keine Sperrereignisse in Change Calendar vorliegen. Wenn die Änderung für ein bestimmtes Datum und eine bestimmte Uhrzeit geplant wurde, beginnt der Runbook-Workflow erst nach der Planung, selbst wenn alle Genehmigungen eingegangen sind und der Kalender nicht blockiert ist.

Themen

- Einrichten von Change Manager
- Arbeiten mit Change Manager
- Prüfen und Protokollieren von Change Manager-Aktivitäten
- Fehlerbehebung für Change Manager

Einrichten von Change Manager

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Sie können ein Tool in verwenden Change Manager AWS Systems Manager, um Änderungen für eine gesamte Organisation, wie in konfiguriert AWS Organizations, oder für eine einzelne Organisation zu verwalten AWS-Konto.

Wenn Sie Change Manager mit einer Organisation verwenden, beginnen Sie mit dem Thema Einrichten von Change Manager für eine Organisation (Management-Konto)und fahren Sie dann mit Konfigurieren von Change Manager-Optionen und bewährten Methoden fort.

Wenn Sie Change Manager mit einem einzelnen Konto verwenden, fahren Sie direkt mit Konfigurieren von Change Manager-Optionen und bewährten Methoden fort.

> **Note**
>
> Wenn Sie mit Nutzung von Change Manager mit einem einzigen Konto beginnen, dieses Konto aber später zu einer Organisationseinheit hinzugefügt wird, für die Change Manager genehmigt ist, werden Ihre individuellen Kontoeinstellungen nicht berücksichtigt.

Themen

- Einrichten von Change Manager für eine Organisation (Management-Konto)
- Konfigurieren von Change Manager-Optionen und bewährten Methoden
- Konfigurieren von Rollen und Berechtigungen für Change Manager
- Steuern des Zugriffs auf Runbook-Workflows für automatische Genehmigung

Einrichten von Change Manager für eine Organisation (Management-Konto)

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Die Aufgaben in diesem Thema geltenChange Manager, wenn Sie ein Tool in oder AWS Systems Manager mit einer Organisation verwenden, die in eingerichtet ist AWS Organizations. Wenn Sie es Change Manager nur mit einem einzigen verwenden möchten AWS-Konto, fahren Sie mit dem Thema fortKonfigurieren von Change Manager-Optionen und bewährten Methoden.

Führen Sie die Aufgaben in diesem Abschnitt in einem aus AWS-Konto , der als Verwaltungskonto in Organizations dient. Weitere Informationen zum Verwaltungskonto und zu anderen Organizations-Konzepten finden Sie unter AWS Organizations -Terminologie und Konzepte.

Wenn Sie Organizations aktivieren und Ihr Konto als Verwaltungskonto angeben müssen, bevor Sie fortfahren, siehe Creating and managing an organization (Erstellen und Verwalten einer Organisation) im AWS Organizations -Benutzerhandbuch.

> **Note**
>
> Dieser Einrichtungsvorgang kann in den folgenden Fällen nicht ausgeführt werden AWS-Regionen:
>
> - Europa (Mailand) (eu-south-1)
> - Naher Osten (Bahrain) (me-south-1)
> - Afrika (Kapstadt) (af-south-1)
> - Asien-Pazifik (Hongkong) (ap-east-1)
>
> Stellen Sie sicher, dass Sie für dieses Verfahren in einer anderen Region in Ihrem Verwaltungskonto arbeiten.

Während des Setupvorgangs führen Sie die folgenden Hauptaufgaben in Quick Setup aus, einem Tool in AWS Systems Manager.

- Aufgabe 1: Registrieren eines delegierten Administrators für Ihre Organisation

  Die änderungsbezogenen Aufgaben, die mit Change Manager ausgeführt werden, werden in einem Ihrer Mitgliedskonten verwaltet, das Sie als delegiertes Administratorkonto angeben. Das delegierte Administratorkonto, das Sie für Change Manager registrieren wird zum delegierten Administratorkonto für alle Systems Manager-Vorgänge. (Möglicherweise haben Sie Administratorkonten für andere delegiert AWS-Services). Ihr delegiertes Administratorkonto für Change Manager, das nicht mit Ihrem Verwaltungskonto identisch ist, verwaltet Änderungsaktivitäten in Ihrer gesamten Organisation, einschließlich Änderungsvorlagen, Änderungsanforderungen und Genehmigungen für jede. Im delegierten Administratorkonto geben Sie auch andere Konfigurationsoptionen für Ihre Change Manager-Operationen an.

> **Important**
>
> Das delegierte Administratorkonto muss das einzige Mitglied der Organisationseinheit (OU) sein, der es in Organizations zugewiesen ist.

- Aufgabe 2: Definieren und Angeben von Runbook-Zugriffsrichtlinien für Änderungsanfordererrollen oder benutzerdefinierte Auftragsfunktionen, die Sie für Ihre Change Manager-Operationen verwenden möchten.

  Um Änderungsanforderungen in erstellen zu könnenChange Manager, müssen Benutzern in Ihren Mitgliedskonten AWS Identity and Access Management (IAM) -Berechtigungen erteilt werden, sodass sie nur auf die Automatisierungs-Runbooks und Änderungsvorlagen zugreifen können, die Sie ihnen zur Verfügung stellen.

  > **Note**
  >
  > Wenn ein Benutzer einen Änderungsantrag erstellt, wählt er zunächst eine Änderungsvorlage aus. Diese Änderungsvorlage stellt möglicherweise mehrere Runbooks zur Verfügung, der Benutzer kann jedoch nur ein Runbook für den jeweiligen Änderungsantrag auswählen. Änderungsvorlagen können auch so konfiguriert werden, dass Benutzer jedes verfügbare Runbook in ihre Anforderungen aufnehmen können.

  Um die erforderlichen Berechtigungen zu erteilen, verwendet Change Manager das Konzept von job functions (Auftragsfunktionen), die auch von IAM verwendet wird. Im Gegensatz zu den AWS -verwalteten Richtlinien für Auftragsfunktionen in IAM geben Sie sowohl die Namen Ihrer Change Manager-Auftragsfunktionen und die IAM-Berechtigungen für diese Auftragsfunktionen an.

  Wenn Sie eine Auftragsfunktion konfigurieren, empfiehlt es sich, eine benutzerdefinierte Richtlinie zu erstellen und nur die Berechtigungen bereitzustellen, die zum Ausführen von Änderungsverwaltungsaufgaben erforderlich sind. Sie können beispielsweise Berechtigungen angeben, die Benutzer basierend auf den von Ihnen definierten Auftragsfunktionen auf diesen bestimmten Satz von Runbooks beschränken.

  Sie können beispielsweise eine Auftragsfunktion mit dem Namen `DBAdmin` erstellen. Für diese Auftragsfunktion können Sie nur Berechtigungen erteilen, die für Runbooks erforderlich sind, die

sich auf Amazon DynamoDB-Datenbanken beziehen, z. B. `AWS-CreateDynamoDbBackup` und `AWSConfigRemediation-DeleteDynamoDbTable`.

Als weiteres Beispiel möchten Sie einigen Benutzern möglicherweise nur die Berechtigungen erteilen, die zum Arbeiten mit Runbooks im Zusammenhang mit Amazon Simple Storage Service (Amazon S3)-Buckets erforderlich sind, z. B. `AWS-ConfigureS3BucketLogging`und `AWSConfigRemediation-ConfigureS3BucketPublicAccessBlock`.

Der Konfigurationsprozess in Quick Setup für Change Manager stellt außerdem eine Reihe vollständiger Administratorberechtigungen für Systems Manager zur Verfügung, die Sie auf eine von Ihnen erstellte Administratorrolle anwenden können.

Jede Change Manager Quick Setup-Konfiguration, die Sie bereitstellen, erstellt eine Auftragsfunktion in Ihrem delegierten Administratorkonto mit Berechtigungen zum Ausführen von Change Manager-Vorlagen und Automation-Runbooks in den von Ihnen ausgewählten Organisationseinheiten. Sie können bis zu 15 Quick Setup-Konfigurationen für Change Manager erstellen.

- Aufgabe 3: Wählen Sie aus, welche Mitgliedskonten in Ihrer Organisation mit Change Manager verwendet werden sollen

  Sie können Change Manager mit allen Mitgliedskonten in allen Organisationseinheiten verwenden, die in Organizations eingerichtet sind, und in allen AWS-Regionen in denen sie arbeiten. Wenn Sie möchten, können Sie stattdessen Change Manager mit nur einigen Ihrer Organisationseinheiten verwenden.

> **Important**
>
> Bevor Sie mit diesem Verfahren beginnen, empfehlen wir dringend, die Schritte zu lesen, um die von Ihnen vorgenommenen Konfigurationsoptionen und die Berechtigungen zu verstehen, die Sie erteilen. Planen Sie insbesondere die benutzerdefinierten Auftragsfunktionen, die Sie erstellen, und die Berechtigungen, die Sie jeder Auftragsfunktion zuweisen. Dadurch wird sichergestellt, dass, wenn Sie später die von Ihnen erstellten Auftragsfunktionsrichtlinien an einzelne Benutzer, Benutzergruppen oder IAM-Rollen anhängen, ihnen nur die Berechtigungen erteilt werden, die Sie für diese beabsichtigen.
> Es hat sich bewährt, zunächst das delegierte Administratorkonto mit dem Anmeldenamen eines Administrators einzurichten. AWS-Konto Konfigurieren Sie dann Auftragsfunktionen

und deren Berechtigungen, nachdem Sie Änderungsvorlagen erstellt und die Runbooks identifiziert haben, die jedes einzelne verwendet.

Um Change Manager für die Verwendung mit einer Organisation einzurichten, führen Sie die folgende Aufgabe im Quick Setup-Bereich der Systems Manager Konsole aus.

Sie wiederholen diese Aufgabe für jede Auftragsfunktion, die Sie für Ihre Organisation erstellen möchten. Jede Auftragsfunktion, die Sie erstellen, kann Berechtigungen für einen anderen Satz von Organisationseinheiten haben.

So richten Sie eine Organisation für Change Manager im Organizations-Verwaltungskonto ein

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Change Manager-Registerkarte Create (Erstellen) aus.
4. Geben Sie für Delegiertes Administratorkonto die ID des AWS-Konto ein, das Sie zum Verwalten von Änderungsvorlagen, Änderungsanforderungen und Runbook-Workflows in Change Manager verwenden möchten.

   Wenn Sie zuvor ein delegiertes Administratorkonto für Systems Manager angegeben haben, wird seine ID bereits in diesem Feld gemeldet.

   > **Important**
   >
   > Das delegierte Administratorkonto muss das einzige Mitglied der Organisationseinheit (OU) sein, der es in Organizations zugewiesen ist.
   > Wenn das delegierte Administratorkonto, das Sie registrieren, später von dieser Rolle abgemeldet wird, entfernt das System seine Berechtigungen für die gleichzeitige Verwaltung von Systems Manager-Vorgängen. Denken Sie daran, dass es notwendig sein wird, dass Sie zum Quick Setup zurückkehren, ein anderes delegiertes Administratorkonto festlegen, und alle Auftragsfunktionen und -Berechtigungen erneut angeben.
   > Wenn Sie den Change Manager in einer Organisation verwenden, empfehlen wir, Änderungen immer über das delegierte Administratorkonto vorzunehmen. Obwohl Sie Änderungen von anderen Konten in der Organisation vornehmen, werden diese

Änderungen nicht im delegierten Administratorkonto gemeldet oder können nicht angezeigt werden.

5. Im Bereich Berechtigungen zum Anfordern und Vornehmen von Änderungen gehen Sie wie folgt vor.

> **Note**
>
> Jede von Ihnen erstellte Bereitstellungskonfiguration stellt die Berechtigungsrichtlinie für nur eine Auftragsfunktion bereit. Sie können zum Quick Setup zurückkehren, um weitere Auftragsfunktionen zu erstellen, wenn Sie Änderungsvorlagen zur Verwendung in Ihren Vorgängen erstellt haben.

So erstellen Sie eine Administratorrolle - Für eine Administratorauftragsfunktion, die IAM-Berechtigungen für alle AWS -Aktionen hat, gehen Sie wie folgt vor.

> **Important**
>
> Das Erteilen von vollständigen Administratorberechtigungen sollte sparsam und nur dann erfolgen, wenn für die Rollen der vollständige Zugriff auf Systems Manager erforderlich ist. Wichtige Informationen zu Sicherheitsüberlegungen für den Zugriff auf Systems Manager finden Sie unter Identity and Access Management für AWS Systems Manager und Bewährte Methoden für die Sicherheit für Systems Manager.

1. Für Auftragsfunktion geben Sie einen Namen zur Identifizierung dieser Rolle und ihrer Berechtigungen ein, z. B. **`My AWS Admin`**.
2. Für die Option Rolle und Berechtigungen wählen Sie Administratorberechtigungen.

So erstellen Sie andere Auftragsfunktionen - Gehen Sie wie folgt vor, um eine nicht-administrative Rolle zu erstellen:

1. Geben Sie für Auftragsfunktion einen Namen ein, um diese Rolle zu identifizieren und ihre Berechtigungen vorzuschlagen. Der von Ihnen gewählte Name sollte den Bereich der Runbooks repräsentieren, für die Sie Berechtigungen erteilen werden, z. B. `DBAdmin` oder `S3Admin`.

2. Für die Option Rolle und Berechtigungen wählen Sie Benutzerdefinierte Berechtigungen.
3. Geben Sie im Editor Berechtigungsrichtlinie die IAM-Berechtigungen im JSON-Format ein, die dieser Auftragsfunktion gewährt werden sollen.

> **Tip**
>
> Es wird empfohlen, dass Sie den IAM-Richtlinien-Editor verwenden, um Ihre Richtlinie zu erstellen und dann den Richtlinien-JSON-Code in das Feld Berechtigungsrichtlinie kopieren.

Beispielrichtlinie: DynamoDB-Datenbankverwaltung

Sie könnten zum Beispiel mit Richtlinieninhalten beginnen, die Berechtigungen für die Arbeit mit den Systems Manager-Dokumenten (SSM-Dokumenten) vorsehen, auf die die Auftragsfunktion Zugriff benötigt. Hier ist ein Beispiel für einen Richtlinieninhalt, der Zugriff auf alle AWS verwalteten Automation-Runbooks gewährt, die sich auf DynamoDB-Datenbanken beziehen, sowie auf zwei Änderungsvorlagen AWS-Konto `123456789012`, die im Beispiel in der Region USA Ost (Ohio) erstellt wurden (). `us-east-2`

Die Richtlinie enthält auch die Berechtigung für die StartChangeRequestExecution Operation, die für die Erstellung eines Änderungsantrags in Change Calendar erforderlich ist.

> **Note**
>
> Dieses Beispiel ist nicht umfassend. Für die Arbeit mit anderen AWS Ressourcen wie Datenbanken und Knoten sind möglicherweise zusätzliche Berechtigungen erforderlich.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "ssm:CreateDocument",
                "ssm:DescribeDocument",
                "ssm:DescribeDocumentParameters",
                "ssm:DescribeDocumentPermission",
                "ssm:GetDocument",
                "ssm:ListDocumentVersions",
                "ssm:ModifyDocumentPermission",
                "ssm:UpdateDocument",
                "ssm:UpdateDocumentDefaultVersion"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:*:document/AWS-CreateDynamoDbBackup",
                "arn:aws:ssm:us-east-1:*:document/AWS-AWS-
DeleteDynamoDbBackup",
                "arn:aws:ssm:us-east-1:*:document/AWS-
DeleteDynamoDbTableBackups",
                "arn:aws:ssm:us-east-1:*:document/AWSConfigRemediation-
DeleteDynamoDbTable",
                "arn:aws:ssm:us-east-1:*:document/AWSConfigRemediation-
EnableEncryptionOnDynamoDbTable",
                "arn:aws:ssm:us-east-1:*:document/AWSConfigRemediation-
EnablePITRForDynamoDbTable",
                "arn:aws:ssm:us-east-1:111122223333:document/
MyFirstDBChangeTemplate",
                "arn:aws:ssm:us-east-1:111122223333:document/
MySecondDBChangeTemplate"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ssm:ListDocuments",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ssm:StartChangeRequestExecution",
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/*",
                "arn:aws:ssm:us-east-1:111122223333:automation-execution/*"
            ]
        }
    ]
}
```

Weitere Informationen zu IAM-Richtlinien finden Sie unter Zugriffsverwaltung für AWS - Ressourcen und Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch

6. Im Bereich Targets wählen Sie aus, ob Sie der gesamten Organisation oder nur einigen Organisationseinheiten Berechtigungen für die Auftragsfunktion gewähren möchten, die Sie erstellen.

   Fahren Sie mit Schritt 9 fort, wenn Sie Ganze Organisation wählen.

   Fahren Sie mit Schritt 8 fort, wenn Sie Benutzerdefiniert wählen.

7. Aktivieren Sie im OUs Abschnitt Ziel die Kontrollkästchen der Organisationseinheiten, mit denen Sie die Option verwenden möchtenChange Manager.

8. Wählen Sie Erstellen aus.

Nachdem das System die Einrichtung von Change Manager für Ihre Organisation abgeschlossen hat, wird eine Zusammenfassung Ihrer Bereitstellungen angezeigt. Diese zusammenfassenden Informationen enthalten den Namen der Rolle, die für die von Ihnen konfigurierte Jobfunktion erstellt wurde. Beispiel, `AWS-QuickSetup-SSMChangeMgr-DBAdminInvocationRole`.

> **Note**
>
> Quick Setupverwendet AWS CloudFormation StackSets , um Ihre Konfigurationen bereitzustellen. Sie können auch Informationen zu einer abgeschlossenen Bereitstellungskonfiguration in der CloudFormation -Konsole einsehen. Weitere Informationen zu StackSets finden Sie unter Arbeiten mit AWS CloudFormation StackSets im AWS CloudFormation Benutzerhandbuch.

Im nächsten Schritt konfigurieren Sie zusätzliche Change Manager-Optionen. Sie können diese Aufgabe entweder in Ihrem delegierten Administratorkonto oder in einem beliebigen Konto in einer Organisationseinheit ausführen, das Sie für die Verwendung mit Change Manager zugelassen haben. Sie konfigurieren Optionen, wie z. B. die Auswahl einer Option für die Verwaltung der Benutzeridentität, die Festlegung, welche Benutzer Änderungsvorlagen und -Anfragen prüfen und genehmigen oder ablehnen können, und die Auswahl der Optionen zu bewährten Methoden, die für Ihr Unternehmen zulässig sein sollen. Weitere Informationen finden Sie unter Konfigurieren von Change Manager-Optionen und bewährten Methoden.

Konfigurieren von Change Manager-Optionen und bewährten Methoden

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Die Aufgaben in diesem Abschnitt müssen unabhängig davon ausgeführt werdenChange Manager, ob Sie ein Tool innerhalb AWS Systems Manager, innerhalb einer Organisation oder in einem einzelnen Tool verwenden AWS-Konto.

Wenn Sie Change Manager für eine Organisation verwenden, können Sie die folgenden Aufgaben entweder in Ihrem delegierten Administratorkonto oder in einem beliebigen Konto in einer Organisationseinheit durchführen, das Sie für die Verwendung mit Change Manager zugelassen haben.

Themen

- Aufgabe 1: Konfigurieren von Change Manager-Benutzeridentitätsverwaltung und Vorlagenprüfern
- Aufgabe 2: Konfigurieren von Change Manager Change-Freeze-Ereignisgenehmigern und bewährten Methoden
- Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen

Aufgabe 1: Konfigurieren von Change Manager-Benutzeridentitätsverwaltung und Vorlagenprüfern

Führen Sie die Aufgabe in diesem Verfahren beim ersten Zugriff auf Change Manager aus. Sie können diese Konfigurationseinstellungen später aktualisieren, indem Sie zu Change Manager zurückkehren und Bearbeiten in der Registerkarte Einstellungen auswählen.

Konfigurieren von Change Manager-Benutzeridentitätsverwaltung und Vorlagenprüfern

1. Melden Sie sich bei der an AWS-Managementkonsole.

   Wenn Sie Change Manager für eine Organisation verwenden, melden Sie sich mit Ihren Anmeldeinformationen für Ihr delegiertes Administratorkonto an. Das Benutzerkonto, das

Sie verwenden, muss über die erforderlichen AWS Identity and Access Management (IAM)-Berechtigungen zum Vornehmen von Aktualisierungen an Ihren Change Manager-Einstellungen verfügen.

2. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
3. Wählen Sie im Navigationsbereich Change Manager aus.
4. Führen Sie auf der Startseite des Dienstes je nach den verfügbaren Optionen einen der folgenden Schritte aus:

   - Wenn Sie Change Manager mit verwenden AWS Organizations , wählen Sie Delegiertes Konto einrichten.
   - Wenn Sie Change Manager mit einem einzigen verwenden AWS-Konto, wählen Sie Einrichten Change Manager.

     –oder–

     Klicken Sie auf Beispieländerungsanforderung erstellen, Überspringen und wählen Sie dann die Registerkarte Einstellungen.
5. Wählen Sie für Verwaltung der Benutzeridentität eine der folgenden Optionen.

   - AWS Identity and Access Management (IAM) — Identifizieren Sie die Benutzer, die Anfragen stellen und genehmigen und andere Aktionen ausführen, Change Manager indem Sie Ihre vorhandenen Benutzer, Gruppen und Rollen verwenden.
   - AWS IAM Identity Center (IAM Identity Center) — Erlauben Sie IAM Identity Center, Identitäten zu erstellen und zu verwalten, oder stellen Sie eine Verbindung zu Ihrer vorhandenen Identitätsquelle her, um die Benutzer zu identifizieren, die Aktionen in ausführen. Change Manager
6. Geben Sie im Abschnitt Template reviewer notification (Benachrichtigung für Vorlagenprüfer) die Amazon Simple Notification Service (Amazon SNS)-Themen an, die verwendet werden sollen, um Vorlagenprüfer darüber zu informieren, dass eine neue Änderungsvorlage oder Änderungsvorlagenversion zur Überprüfung bereit ist. Stellen Sie sicher, dass das von Ihnen ausgewählte Amazon SNS-Thema so konfiguriert ist, dass Benachrichtigungen an Ihre Vorlagenprüfer gesendet werden.

   Informationen zum Erstellen und Konfigurieren von Amazon SNS-Themen für Änderungsvorlagenprüferbenachrichtigungen finden Sie unter Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen.

1. Wählen Sie eine der folgenden Optionen aus, um das Amazon SNS-Thema für die Benachrichtigung der Vorlagenprüfer anzugeben:
   - Geben Sie einen SNS-Amazon-Ressourcenname (ARN) ein - Geben Sie für Thema-ARN einen ARN eines vorhandenen Amazon SNS Themas ein. Dieses Thema kann sich in jedem Konto Ihrer Organisation befinden.
   - Wählen Sie ein vorhandenes SNS-Thema - Wählen Sie für Target notification topic den ARN eines vorhandenen Amazon SNS-Themas in Ihrem aktuellen AWS-Konto. (Diese Option ist nicht verfügbar, wenn Sie in Ihrem aktuellen AWS-Konto und noch keine Amazon SNS SNS-Themen erstellt haben AWS-Region.)

   > **Note**
   >
   > Das von Ihnen ausgewählte Amazon SNS-Thema muss so konfiguriert werden, dass die gesendeten Benachrichtigungen und die Abonnenten, an die sie gesendet werden, festgelegt werden. Seine Zugriffsrichtlinie muss auch Systems Manager Berechtigungen gewähren, damit Change Manager Benachrichtigungen senden kann. Weitere Informationen finden Sie unter Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen.

2. Wählen Sie Add notification (Benachrichtigung hinzufügen) aus.

7. Wählen Sie im Abschnitt Änderungsvorlagenprüfer die Benutzer in Ihrer Organisation oder Ihrem Konto aus, um neue Änderungsvorlagen zu überprüfen oder Vorlagenversionen zu ändern, bevor sie in Ihren Vorgängen verwendet werden können.

   Änderungsvorlagenprüfer sind dafür verantwortlich, die Eignung und Sicherheit von Vorlagen zu überprüfen, die andere Benutzer zur Verwendung in Change Manager-Runbook-Workflows eingereicht haben.

   Wählen Sie die Änderungsvorlagenprüfer folgendermaßen aus:

   1. Wählen Sie Hinzufügen aus.
   2. Aktivieren Sie das Kontrollkästchen neben dem Namen aller Benutzer, Gruppen oder IAM-Rollen, die Sie als Änderungsvorlagenprüfer zuweisen möchten.
   3. Wählen Sie Add approvers (Hinzufügen von Genehmigern).

8. Wählen Sie Absenden aus.

Nachdem Sie diese erste Einrichtung abgeschlossen haben, konfigurieren Sie zusätzliche Change Manager-Einstellungen und bewährte Methoden, indem Sie die Schritte unterAufgabe 2: Konfigurieren von Change Manager Change-Freeze-Ereignisgenehmigern und bewährten Methoden befolgen.

Aufgabe 2: Konfigurieren von Change Manager Change-Freeze-Ereignisgenehmigern und bewährten Methoden

Nachdem Sie die Schritte unter Aufgabe 1: Konfigurieren von Change Manager-Benutzeridentitätsverwaltung und Vorlagenprüfern abgeschlossen haben, können Sie zusätzliche Änderungsanforderungsprüfer während Change–Freeze-Ereignissen bestimmen und angeben, welche verfügbaren bewährten Methoden Sie für Ihre Change Manager-Vorgänge zulassen wollen.

Ein Ereignis zum Einfrieren von Änderungen bedeutet, dass Einschränkungen im aktuellen Änderungskalender gelten (der Kalenderstatus AWS Systems Manager Change Calendar ist`CLOSED`). In diesen Fällen müssen zusätzlich zu den regulären Genehmigern für Änderungsanforderungen oder wenn die Änderungsanforderung mit einer Vorlage erstellt wurde, die automatische Genehmigungen zulässt, die Genehmiger des Änderungsstopps die Genehmigung für die Ausführung dieser Änderungsanforderung erteilen. Wenn dies nicht der Fall ist, wird die Änderung erst verarbeitet, wenn der Kalenderstatus wieder `OPEN` ist.

Konfigurieren von Change Manager Change-Freeze-Ereignisgenehmigern und bewährten Methoden

1. Wählen Sie im Navigationsbereich Change Manager aus.
2. Wählen Sie die Registerkarte Einstellungen und anschließend Bearbeiten.
3. Wählen Sie im Abschnitt Genehmiger für Change-Freeze-Ereignisse die Benutzer in Ihrer Organisation oder Ihrem Konto aus, die Änderungen genehmigen können, die ausgeführt werden, selbst wenn der verwendete Kalender in Change Calendar derzeit GESCHLOSSEN ist.

   > **Note**
   >
   > Um Change-Freeze-Überprüfungen zu erlauben, müssen Sie das Kontrollkästchen für die Option Änderungskalender auf eingeschränkte Änderungsereignisse prüfen in Bewährte Methoden aktivieren.

   Wählen Sie Genehmiger für Change–Freeze-Ereignisse aus, indem Sie die folgenden Schritte ausführen:

1. Wählen Sie Hinzufügen aus.
2. Aktivieren Sie das Kontrollkästchen neben dem Namen aller Benutzer, Gruppen oder IAM-Rollen, die Sie als Genehmiger für Change-Freeze-Ereignisse zuweisen möchten.
3. Wählen Sie Add approvers (Hinzufügen von Genehmigern).

4. Aktivieren Sie im Abschnitt Bewährte Methoden unten auf der Seite die bewährten Methoden, die Sie für jede der folgenden Optionen erzwingen möchten.

   - Option:Änderungskalender auf eingeschränkte Änderungsereignisse prüfen

     Um anzugeben, dass Change Manager einen Kalender in Change Calendar überprüft, um sicherzustellen, dass Änderungen nicht durch geplante Ereignisse blockiert werden, wählen Sie zunächst das Kontrollkästchen Enabled und wählen Sie dann den Kalender aus, um auf eingeschränkte Ereignisse der Änderungskalender-Liste zu prüfen.

     Mehr über Change Calendar erfahren Sie unter AWS Systems Manager Change Calendar.

   - Option: SNS-Thema für Genehmiger für geschlossene Ereignisse

     1. Wählen Sie eine der folgenden Optionen aus, um das Amazon Simple Notification Service (Amazon SNS)-Thema in Ihrem Konto anzugeben, das für das Senden von Benachrichtigungen an Genehmiger während der Change-Freeze-Ereignisse verwendet werden soll. (Beachten Sie, dass Sie Genehmiger auch im Abschnitt Genehmiger für Change-Freeze-Ereignisse über Bewährte Methoden angeben müssen.)
        - Geben Sie einen SNS-Amazon-Ressourcenname (ARN) ein - Geben Sie für Thema-ARN einen ARN eines vorhandenen Amazon SNS Themas ein. Dieses Thema kann sich in jedem Konto Ihrer Organisation befinden.
        - Wählen Sie ein vorhandenes SNS-Thema - Wählen Sie für Target notification topic den ARN eines vorhandenen Amazon SNS-Themas in Ihrem aktuellen AWS-Konto. (Diese Option ist nicht verfügbar, wenn Sie in Ihrem aktuellen AWS-Konto und noch keine Amazon SNS SNS-Themen erstellt haben AWS-Region.)

        > **Note**
        >
        > Das von Ihnen ausgewählte Amazon SNS-Thema muss so konfiguriert werden, dass die gesendeten Benachrichtigungen und die Abonnenten, an die sie gesendet werden, festgelegt werden. Seine Zugriffsrichtlinie muss auch Systems Manager Berechtigungen gewähren, damit Change Manager Benachrichtigungen senden

kann. Weitere Informationen finden Sie unter Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen.

2. Wählen Sie Add notification (Benachrichtigung hinzufügen) aus.

- Option: Überwachungen für alle Vorlagen erforderlich

  Wenn Sie sicherstellen möchten, dass alle Vorlagen für Ihre Organisation oder Ihr Konto einen CloudWatch Amazon-Alarm zur Überwachung Ihres Änderungsvorgangs angeben, aktivieren Sie das Kontrollkästchen Aktiviert.

- Option: Überprüfung und Genehmigung der Vorlage vor der Verwendung erforderlich

  Um sicherzustellen, dass keine Änderungsanforderungen erstellt und keine Runbook-Workflows ausgeführt werden, ohne auf einer Vorlage basieren zu müssen, die überprüft und genehmigt wurde, aktivieren Sie das Kontrollkästchen Enabled.

5. Wählen Sie Speichern.

Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen

Change ManagerÄnderung der Verfügbarkeit

AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Sie können ein Tool in konfigurieren Change Manager AWS Systems Manager, um Benachrichtigungen an ein Amazon Simple Notification Service (Amazon SNS) -Thema für Ereignisse im Zusammenhang mit Änderungsanforderungen und Änderungsvorlagen zu senden. Führen Sie die folgenden Aufgaben aus, um Benachrichtigungen für die Change Manager-Ereignisse zu erhalten, denen Sie ein Thema hinzufügen.

Themen

- Aufgabe 1: Erstellen und Abonnieren eines Amazon SNS-Themas
- Aufgabe 2: Aktualisieren der Amazon SNS-Zugriffsrichtlinie
- Aufgabe 3: (Optional) Aktualisieren Sie die Zugriffsrichtlinie AWS Key Management Service

Aufgabe 1: Erstellen und Abonnieren eines Amazon SNS-Themas

Zunächst müssen Sie ein Amazon SNS-Thema erstellen und abonnieren. Weitere Informationen finden Sie unter Erstellen eines Amazon-SNS-Themas und Abonnieren eines Amazon-SNS-Themas im Entwicklerhandbuch zu Amazon Simple Notification Service.

> **Note**
>
> Um Benachrichtigungen zu erhalten, müssen Sie den Amazon-Ressourcennamen (ARN) eines Amazon SNS-Themas angeben, das sich im selben AWS-Region und AWS-Konto wie das delegierte Administratorkonto befindet.

Aufgabe 2: Aktualisieren der Amazon SNS-Zugriffsrichtlinie

Gehen Sie wie folgt vor, um die Amazon SNS-Zugriffsrichtlinie zu aktualisieren, damit Systems Manager Change Manager-Benachrichtigungen für das Amazon SNS-Thema veröffentlichen kann, das Sie in Aufgabe 1 erstellt haben. Ohne Fertigstellung dieser Aufgabe hat Change Manager keine Berechtigung zum Senden von Benachrichtigungen für die Ereignisse, für die Sie das Thema hinzufügen.

1. Melden Sie sich bei https://console.aws.amazon.com/sns/v3/home an AWS-Managementkonsole und öffnen Sie die Amazon SNS SNS-Konsole.
2. Wählen Sie im Navigationsbereich Themen aus.
3. Wählen Sie das Thema aus, das Sie in Aufgabe 1 erstellt haben und klicken Sie dann auf Edit (Bearbeiten).
4. Erweitern Sie die Option Zugriffsrichtlinie.
5. Fügen Sie den folgenden `Sid` Block zur bestehenden Richtlinie hinzu, aktualisieren Sie ihn und ersetzen Sie jeden Block durch Ihre *`user input placeholder`* eigenen Informationen.

```
{
    "Sid": "Allow Change Manager to publish to this topic",
    "Effect": "Allow",
    "Principal": {
        "Service": "ssm.amazonaws.com"
    },
    "Action": "sns:Publish",
    "Resource": "arn:aws:sns:region:account-id:topic-name",
    "Condition": {
```

```
        "StringEquals": {
            "aws:SourceAccount": [
                "account-id"
            ]
        }
    }
}
```

Geben Sie diesen Block nach dem vorhandenen `Sid` Block ein und ersetzen Sie*region*,*account-id*, und *topic-name* durch die entsprechenden Werte für das von Ihnen erstellte Thema.

6. Wählen Sie Änderungen speichern aus.

Das System sendet jetzt Benachrichtigungen an das Amazon SNS-Thema, wenn der Ereignistyp auftritt, den Sie dem Thema hinzufügen.

> **Important**
>
> Wenn Sie das Amazon SNS SNS-Thema mit einem serverseitigen Verschlüsselungsschlüssel AWS Key Management Service (AWS KMS) konfiguriert haben, müssen Sie Aufgabe 3 abschließen.

Aufgabe 3: (Optional) Aktualisieren Sie die Zugriffsrichtlinie AWS Key Management Service

Wenn Sie die serverseitige Verschlüsselung AWS Key Management Service (AWS KMS) für Ihr Amazon SNS SNS-Thema aktiviert haben, müssen Sie auch die Zugriffsrichtlinie des Themas aktualisieren, das AWS KMS key Sie bei der Konfiguration des Themas ausgewählt haben. Gehen Sie wie folgt vor, um die Zugriffsrichtlinie zu aktualisieren, damit Systems Manager Change Manager-Genehmigungsbenachrichtigungen für das Amazon SNS-Thema veröffentlichen kann, das Sie in Aufgabe 1 erstellt haben.

1. Öffnen Sie die AWS KMS Konsole unter /kms. https://console.aws.amazon.com
2. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
3. Wählen Sie die ID des Kundenmasterschlüssels aus, den Sie bei der Erstellung des Themas ausgewählt haben.
4. Wählen Sie im Abschnitt Key policy (Schlüsselrichtlinie) die Option Switch to policy view (Zur Richtlinienansicht wechseln) aus.

5. Wählen Sie Bearbeiten aus.
6. Geben Sie den folgenden `Sid`-Block nach einem der vorhandenen `Sid`-Blöcke in die vorhandene Richtlinie ein. Ersetzen Sie jeden *`user input placeholder`* durch Ihre Informationen.

```
{
    "Sid": "Allow Change Manager to decrypt the key",
    "Effect": "Allow",
    "Principal": {
        "Service": "ssm.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "arn:aws:kms:region:account-id:key/key-id",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": [
                "account-id"
            ]
        }
    }
}
```

7. Geben Sie nun den folgenden `Sid`-Block nach einem der vorhandenen `Sid`-Blöcke in die Ressourcenrichtlinie ein, um zu verhindern, dass das Problem des dienstübergreifenden verwirrten Stellvertreters auftritt.

   Dieser Block verwendet die globalen Bedingungskontextschlüssel `aws:SourceArn` und `aws:SourceAccount`, um die Berechtigungen einzuschränken, die Systems Manager der Ressource einem anderen Dienst erteilt.

   Ersetzen Sie jeden *`user input placeholder`* durch Ihre Informationen.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "Configure confused deputy protection for AWS KMS keys
 used in Amazon SNS topic when called from Systems Manager",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": [
                "sns:Publish"
            ],
            "Resource": "arn:aws:sns:us-east-1:111122223333:topic-name",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:ssm:us-east-1:111122223333:*"
                },
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

8. Wählen Sie Änderungen speichern aus.

Konfigurieren von Rollen und Berechtigungen für Change Manager

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Standardmäßig hat Change Manager keine Berechtigung zur Ausführung von Aktionen auf Ihre Ressourcen. Sie müssen den Zugriff mithilfe einer AWS Identity and Access Management (IAM-) Servicerolle gewähren oder eine Rolle übernehmen. Diese Rolle ermöglicht es Change Manager, die in einer genehmigten Änderungsanforderung angegebenen Runbook-Workflows in Ihrem Namen sicher auszuführen. Die Rolle gewährt AWS Security Token Service (AWS STS) AssumeRoleVertrauen fürChange Manager.

Durch die Bereitstellung dieser Berechtigungen für eine Rolle, um im Namen von Benutzern in einer Organisation zu handeln, muss Benutzern dieses Array von Berechtigungen nicht selbst gewährt werden. Die durch die Berechtigungen zulässigen Aktionen sind nur auf genehmigte Vorgänge beschränkt.

Wenn Benutzer in Ihrem Konto oder Ihrer Organisation eine Änderungsanforderung erstellen, können sie diese Übernahmerolle auswählen, um die Änderungsvorgänge auszuführen.

Sie können eine neue Übernahmerolle für Change Manager erstellen oder eine vorhandene Rolle mit den erforderlichen Berechtigungen aktualisieren.

Wenn Sie eine Servicerolle für Change Manager erstellen müssen, führen Sie die folgenden Schritte aus.

Aufgaben

- Aufgabe 1: Erstellen einer Übernahmerollenrichtlinie für Change Manager
- Aufgabe 2: Erstellen einer Übernahmerolle für Change Manager
- Aufgabe 3: Anfügen der iam:PassRole-Richtlinie an andere Rollen
- Aufgabe 4: Hinzufügen von Inline-Richtlinien zu einer Rolle übernehmen, um andere aufzurufen AWS-Services
- Aufgabe 5: Konfigurieren des Benutzerzugriffs auf Change Manager

Aufgabe 1: Erstellen einer Übernahmerollenrichtlinie für Change Manager

Verwenden Sie das folgende Verfahren, um die Richtlinie zu erstellen, die Sie an Ihre Change Manager-Übernahmerolle anfügen.

Erstellen einer Übernahmerollenrichtlinie für Change Manager

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Richtlinien und dann Richtlinie erstellen.
3. Wählen Sie auf der Seite Create policy (Richtlinie erstellen) die Registerkarte JSON aus und ersetzen Sie den Standardinhalt durch folgenden, den Sie in den folgenden Schritten für Ihre eigenen Change Manager-Vorgänge ändern.

> **Note**
>
> Wenn Sie eine Richtlinie für ein einzelnes AWS-Konto Konto und nicht für eine Organisation mit mehreren Konten erstellen AWS-Regionen, können Sie den ersten Anweisungsblock weglassen. Die `iam:PassRole`-Berechtigung ist nicht erforderlich, wenn ein einziges Konto Change Manager verwendet.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/AWS-
SystemsManager-job-functionAdministrationRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ssm.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeDocument",
                "ssm:GetDocument",
                "ssm:StartChangeRequestExecution"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1::document/template-name",
                "arn:aws:ssm:us-east-1:111122223333:automation-execution/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:ListOpsItemEvents",
```

```
                "ssm:GetOpsItem",
                "ssm:ListDocuments",
                "ssm:DescribeOpsItems"
            ],
            "Resource": "*"
        }
    ]
}
```

4. Aktualisieren Sie für die `iam:PassRole` Aktion den `Resource` Wert so, dass er alle für Ihre Organisation definierten Jobfunktionen enthält, denen Sie Berechtigungen zum Initiieren von Runbook-Workflows gewähren möchten. ARNs
5. Ersetzen Sie die *`job-function`* Platzhalter *`region`* *`account-id`*`template-name`,*`delegated-admin-account-id`*,, und durch Werte für Ihre Change Manager Operationen.
6. Ändern Sie für die zweite `Resource`-Anweisung die Liste so, dass sie alle Änderungsvorlagen enthält, für die Sie Berechtigungen erteilen möchten. Alternativ können Sie `"Resource": "*"` angeben, um Berechtigungen für alle Änderungsvorlagen in Ihrer Organisation zu erteilen.
7. Wählen Sie Weiter: Tags aus.
8. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Richtlinie zu organisieren, zu verfolgen oder zu steuern.
9. Wählen Sie Weiter: Prüfen aus.
10. Geben Sie auf der Seite Review policy (Richtlinie überprüfen) im Feld Name einen Namen ein, wie z. B **`MyChangeManagerAssumeRole`**, und geben Sie anschließend eine optionale Beschreibung ein.
11. Klicken Sie auf Create policy (Richtlinie erstellen) und fahren Sie mit Aufgabe 2: Erstellen einer Übernahmerolle für Change Manager fort.

Aufgabe 2: Erstellen einer Übernahmerolle für Change Manager

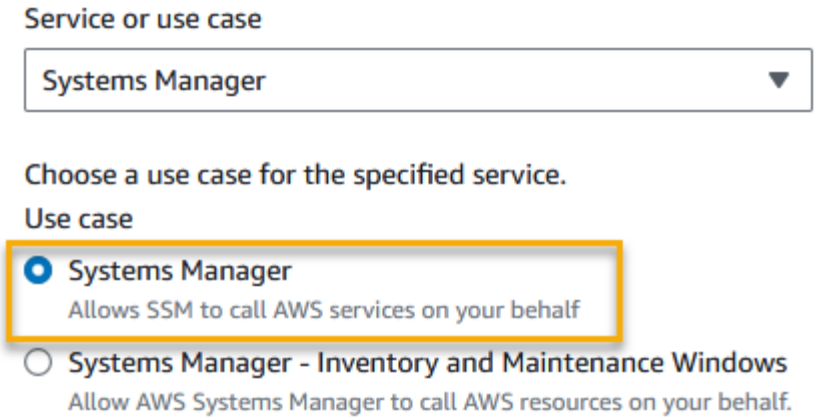
Führen Sie die folgenden Schritte zum Erstellen einer Change Manager-Übernahmerolle, ein Art von Servicerolle, für Change Manager aus.

Erstellen einer Übernahmerolle für Change Manager

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen und dann Rolle erstellen.

3. Wählen Sie für Select trusted entity (Vertrauenswürdige Entität auswählen) die folgenden Optionen:

   1. Wählen Sie unter Trusted entity type (Typ der vertrauenswürdigen Entität) die Option AWS - Service
   2. Für Anwendungsfälle für andere AWS-Services wählen Sie Systems Manager
   3. Wählen Sie Systems Manager, wie im folgenden Image gezeigt.



4. Wählen Sie Weiter aus.
5. Suchen Sie auf der Seite Attached permissions policy (Richtlinie für angehängte Berechtigungen) nach der Übernahmerollenrichtlinie, die Sie in Aufgabe 1: Erstellen einer Übernahmerollenrichtlinie für Change Manager erstellt haben, wie beispielsweise **`MyChangeManagerAssumeRole`**.
6. Aktivieren Sie das Kontrollkästchen neben dem Namen der Übernahmerollenrichtlinie und wählen Sie anschließend Next: Tags (Weiter: Tags) aus.
7. Geben Sie unter Role name (Rollenname) einen Namen für Ihr neues Instance-Profil ein, wie z. B. **`MyChangeManagerAssumeRole`**.
8. (Optional) Aktualisieren Sie für Description (Beschreibung) die Beschreibung für diese Instance-Rolle.
9. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Rolle zu organisieren, zu verfolgen oder zu steuern.
10. Wählen Sie Weiter: Prüfen aus.
11. (Optional) Fügen Sie für Tags ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Rolle zu organisieren, nachzuverfolgen oder zu steuern, und wählen Sie dann Rolle erstellen aus. Das System leitet Sie zur Seite Rollen zurück.
12. Wählen Sie Create role (Rolle erstellen) aus. Das System leitet Sie zur Seite Rollen zurück.

13. Wählen Sie auf der Seite Roles (Rollen) die gerade erstellte Rolle aus, um die Seite Summary (Übersicht) zu öffnen.

Aufgabe 3: Anfügen der **`iam:PassRole`**-Richtlinie an andere Rollen

Gehen Sie wie nachfolgend beschrieben vor, um die `iam:PassRole`-Richtlinie an ein IAM-Instance-Profil oder eine IAM-Servicerolle anzuhängen. (Der Systems Manager Manager-Dienst verwendet IAM-Instanzprofile, um mit EC2 Instanzen zu kommunizieren. Für nicht EC2 verwaltete Knoten in einer Hybrid- und Multi-Cloud-Umgebung wird stattdessen eine IAM-Servicerolle verwendet.)

Durch Anfügen der `iam:PassRole`-Richtlinie, kann der Change Manager-Service Übernahmerollenberechtigungen anderen Services oder Systems-Manager-Tools übergeben, wenn Runbook-Workflows ausgeführt werden.

Fügen Sie die **`iam:PassRole`**-Richtlinie an ein IAM-Instance-Profil oder eine Servicerolle wie folgt an

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen.
3. Suchen Sie nach der Change Manager-Übernahmerolle, die Sie erstellt haben, wie z. B **`MyChangeManagerAssumeRole`**, und wählen Sie seinen Namen aus.
4. Wählen Sie auf der Seite Summary (Zusammenfassung) für die gerade erstellte Rolle die Registerkarte Permissions (Berechtigungen) aus.
5. Wählen Sie Berechtigungen hinzufügen, eingebundene Richtlinie erstellen.
6. Wählen Sie auf der Seite Richtlinie erstellen die Registerkarte Visueller Editor aus.
7. Wählen Sie Service (Service) und anschließend die Option IAM aus.
8. Geben Sie im Textfeld Aktionen filtern die Option ein**`PassRole`**, und wählen Sie sie aus. PassRole
9. Erweitern Sie Resources (Ressourcen). Stellen Sie sicher, dass Specific ausgewählt ist und wählen Sie dann Add ARN aus.
10. Geben Sie im Feld Specify ARN for role (ARN für Rolle angeben) den ARN der IAM-Instance-Profilrolle oder der IAM-Servicerolle ein, an die Sie Übernahmerollenberechtigungen übergeben möchten. Das System füllt die Felder Account (Konto) und Role name with path (Rollenname mit Pfad) automatisch aus.
11. Wählen Sie Hinzufügen aus.

12. Wählen Sie Richtlinie prüfen.
13. Geben Sie für Name einen Namen ein, um diese Richtlinie zu identifizieren und wählen Sie dann Create poliy (Richtlinie erstellen) aus.

Weitere Informationen

- Erforderliche Instance-Berechtigungen für Systems Manager konfigurieren
- Die für Systems Manager in Hybrid- und Multi-Cloud-Umgebungen erforderliche IAM-Servicerolle erstellen

Aufgabe 4: Hinzufügen von Inline-Richtlinien zu einer Rolle übernehmen, um andere aufzurufen AWS-Services

Wenn eine Änderungsanforderung andere AWS-Services mithilfe der Rolle „Change ManagerÜbernehmen“ aufruft, muss die Rolle „Übernehmen“ so konfiguriert werden, dass sie berechtigt ist, diese Dienste aufzurufen. Diese Anforderung gilt für alle AWS Automations-Runbooks (AWS-*-Runbooks), die möglicherweise in einer Änderungsanforderung verwendet werden, wie z. B. die Runbooks`AWS-ConfigureS3BucketLogging`, und`AWS-CreateDynamoDBBackup`. `AWS-RestartEC2Instance` Diese Anforderung gilt auch für alle von Ihnen erstellten benutzerdefinierten Runbooks, die andere mithilfe AWS-Services von Aktionen aufrufen, die andere Dienste aufrufen. Wenn Sie unter anderem `aws:executeAwsApi`-, `aws:CreateStack`- oder `aws:copyImage`-Aktionen verwenden, dann müssen Sie die Servicerolle mit der Berechtigung zum Aufrufen solcher Services konfigurieren. Sie können Berechtigungen für andere AWS-Services aktivieren, indem Sie der IAM-Rolle eine eingebundene Richtlinie hinzufügen.

So fügen Sie einer angenommenen Rolle eine eingebunden Richtlinie hinzu, um andere AWS-Services (IAM-Konsole) aufzurufen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im Navigationsbereich Rollen.
3. Wählen Sie in der Liste den Namen der Übernahmerolle aus, die Sie aktualisieren möchten, z. B `MyChangeManagerAssumeRole`.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie Add permissions, Create inline policy (Berechtigungen hinzufügen, eingebundene Richtlinie erstellen).

6. Wählen Sie den Tab JSON.
7. Geben Sie ein JSON-Richtliniendokument für das ein AWS-Services , das Sie aufrufen möchten. Nachfolgend sind zwei Beispiele für JSON-Richtliniendokumente aufgeführt.

   Amazon-S3-**`PutObject`** und **`GetObject`**-Beispiel

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

   Amazon EC2 **`CreateSnapshot`** und **`DescribeSnapShots`** Beispiel

   JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":"ec2:DescribeSnapshots",
         "Resource":"*"
      }
```

```
    ]
}
```

Details zur IAM-Richtliniensprache und finden Sie in der IAM JSON Policy Reference im IAM-Benutzerhandbuch.

8. Wählen Sie, wenn Sie fertig sind, Review policy (Richtlinie überprüfen) aus. Die Richtlinienvalidierung meldet mögliche Syntaxfehler.
9. Für Name geben Sie einen Namen zur Identifizierung der Richtlinie ein, die Sie erstellen. Überprüfen Sie unter Summary die Richtlinienzusammenfassung, um die Berechtigungen einzusehen, die von Ihrer Richtlinie gewährt werden. Wählen Sie dann Create policy aus, um Ihre Eingaben zu speichern.
10. Nachdem Sie eine Inline-Richtlinie erstellt haben, wird sie automatisch in Ihre Rolle eingebettet.

Aufgabe 5: Konfigurieren des Benutzerzugriffs auf Change Manager

Wenn Ihrem Benutzer, Ihrer Gruppe oder Rolle Administratorrechte zugewiesen sind, haben Sie Zugriff auf Change Manager. Wenn Sie nicht über Administratorberechtigungen verfügen, muss ein Administrator die von `AmazonSSMFullAccess` verwaltete Richtlinie oder eine Richtlinie, die vergleichbare Berechtigungen bereitstellt, Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle zuweisen.

Konfigurieren Sie mit den folgenden Schritten einen Benutzer zur Verwendung von Change Manager. Der ausgewählte Benutzer verfügt über die Berechtigung zum Konfigurieren und Ausführen von Change Manager.

Abhängig von der Identitätsanwendung, die Sie in Ihrer Organisation verwenden, können Sie eine der drei verfügbaren Optionen zum Konfigurieren des Benutzerzugriffs auswählen. Weisen Sie beim Konfigurieren des Benutzerzugriffs Folgendes zu oder fügen Sie Folgendes hinzu:

1. Weisen Sie die `AmazonSSMFullAccess`-Richtlinie oder eine vergleichbare Richtlinie zu, die Zugriff auf Systems Manager gewährt.
2. Weisen Sie die `iam:PassRole`-Richtlinie zu.
3. Fügen Sie den ARN für die von Change Manager übernommene Rolle hinzu, die Sie am Ende von Aufgabe 2: Erstellen einer Übernahmerolle für Change Manager kopiert haben.

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

  Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center -Benutzerhandbuch.
- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

  Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anleitung unter Eine Rolle für einen externen Identitätsanbieter (Verbund) erstellen im IAM-Benutzerhandbuch.
- IAM-Benutzer:
  - Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Befolgen Sie die Anleitung unter Eine Rolle für einen IAM-Benutzer erstellen im IAM-Benutzerhandbuch.
  - (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter Hinzufügen von Berechtigungen zu einem Benutzer (Konsole) im IAM-Benutzerhandbuch.

Sie haben die Konfiguration der erforderlichen Rollen für Change Manager abgeschlossen. Sie können jetzt die Change Manager-Übernahmerolle-ARN in Ihren Change Manager-Vorgängen verwenden.

Steuern des Zugriffs auf Runbook-Workflows für automatische Genehmigung

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

In jeder Änderungsvorlage, die für Ihre Organisation oder Ihr Konto erstellt wurde, können Sie angeben, ob Änderungsanforderungen, die mit dieser Vorlage erstellt wurden, als automatisch genehmigte Änderungsanforderungen ausgeführt werden können. Dies bedeutet, dass sie automatisch ohne Überprüfungsschritt ausgeführt werden (mit Ausnahme von Change–Freeze-Ereignissen).

Möglicherweise möchten Sie jedoch verhindern, dass bestimmte Benutzer, Gruppen oder AWS Identity and Access Management (IAM-) Rollen automatisch genehmigte Änderungsanforderungen

ausführen, auch wenn eine Änderungsvorlage dies zulässt. Sie können dies durch die Verwendung des `ssm:AutoApprove`-Bedingungsschlüssel für den `StartChangeRequestExecution`-Vorgang in einer IAM-Richtlinie tun, die der Benutzer-, Gruppen- oder IAM-Rolle zugewiesen ist.

Sie können die folgende Richtlinie als Inline-Richtlinie hinzufügen, wobei die Bedingung als `false` angegeben wird, um zu verhindern, dass Benutzer automatisch genehmigungsfähige Änderungsanforderungen ausführen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
            {
            "Effect": "Allow",
            "Action": "ssm:StartChangeRequestExecution",
            "Resource": "*",
            "Condition": {
                "BoolIfExists": {
                    "ssm:AutoApprove": "false"
                }
            }
        }
    ]
}
```

Informationen zum Festlegen von Inline-Richtlinien finden Sie unter Inline-Richtlinien und Hinzufügen und Entfernen von IAM-Identitätsberechtigungen im IAM-Benutzerhandbuch.

Weitere Informationen über Bedingungsschlüssel für Systems Manager finden Sie unter Condition keys for Systems Manager (Bedingungsschlüssel für Systems Manager).

Arbeiten mit Change Manager

> **Change ManagerVerfügbarkeit ändern**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt

nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Mit Change Manager einem Tool in AWS-Konto können Benutzer in AWS Systems Manager Ihrer gesamten Organisation oder in einem einzigen Tool Aufgaben im Zusammenhang mit Änderungen ausführen, für die ihnen die erforderlichen Berechtigungen erteilt wurden. Change ManagerZu den Aufgaben gehören die folgenden:

- Änderungsvorlagen erstellen, prüfen und genehmigen oder ablehnen.

  Eine Änderungsvorlage ist eine Sammlung von Konfigurationseinstellungen in Change Manager, die beispielsweise erforderliche Genehmigungen, verfügbare Runbooks und Benachrichtigungsoptionen für Änderungsanforderungen definiert.
- Änderungsanforderungen erstellen, prüfen und genehmigen oder ablehnen.

  Eine Änderungsanforderung ist eine Anforderung Change Manager zur Ausführung eines Automatisierungs-Runbooks, das eine oder mehrere Ressourcen in Ihren AWS oder lokalen Umgebungen aktualisiert. Ein Änderungsantrag wird mit einer Änderungsvorlage erstellt.
- Geben Sie an, welche Benutzer in Ihrer Organisation oder Ihrem Konto zu Prüfern für Änderungsvorlagen und Änderungsanforderungen gemacht werden können.
- Bearbeiten Sie Konfigurationseinstellungen, z. B. die Verwaltung von Benutzeridentitäten in Change Manager und welche der verfügbaren bewährten Methoden-Optionen in Ihrem Change Manager erzwungen werden. Weitere Informationen zum Konfigurieren dieser Einstellungen finden Sie unter Konfigurieren von Change Manager-Optionen und bewährten Methoden.

Themen

- Arbeiten mit Änderungsvorlagen
- Verwenden von Änderungsanforderungen
- Überprüfen von Details, Aufgaben und Zeitplänen für Änderungsanforderungen (Konsole)
- Aggregierte Anzahl von Änderungsaufträgen anzeigen (Befehlszeile)

Arbeiten mit Änderungsvorlagen

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Eine Änderungsvorlage ist eine Sammlung von Konfigurationseinstellungen in Change Manager, die beispielsweise erforderliche Genehmigungen, verfügbare Runbooks und Benachrichtigungsoptionen für Änderungsanforderungen definiert.

> **Note**
>
> AWS bietet ein Beispiel für eine Hello World-Änderungsvorlage, die Sie zum Ausprobieren verwenden könnenChange Manager, ein Tool in AWS Systems Manager. Sie erstellen jedoch Ihre eigenen Änderungsvorlagen, um die Änderungen zu definieren, die Sie an den Ressourcen in Ihrer Organisation oder Ihrem Konto zulassen möchten.

Die Änderungen, die bei der Ausführung eines Runbook-Workflows vorgenommen werden, basieren auf dem Inhalt eines Automation-Runbooks. In jede von Ihnen erstellte Änderungsvorlage können Sie ein oder mehrere Automation-Runbooks aufnehmen, aus denen der Benutzer, der eine Änderungsanforderung stellt, auswählen kann, um sie während der Aktualisierung auszuführen. Sie können auch Änderungsvorlagen erstellen, mit denen Anforderer ein beliebiges Automation-Runbook für den Änderungsantrag auswählen können.

Um eine Änderungsvorlage zu erstellen, können Sie die Builder-Option in der Konsolenseite Vorlage erstellen verwenden, um eine Änderungsvorlage zu erstellen. Alternativ können Sie mit der Editor-Option JSON- oder YAML-Inhalte mit der gewünschten Konfiguration für Ihren Runbook-Workflow manuell erstellen. Sie können auch ein Befehlszeilentool verwenden, um eine Änderungsvorlage zu erstellen, wobei JSON-Inhalt für die Änderungsvorlage in einer externen Datei gespeichert ist.

Themen

- Testen Sie die Vorlage für AWS verwaltete Hello World Änderungen

- Erstellen von Änderungsvorlagen
- Überprüfen und Genehmigen oder Ablehnen von Änderungsvorlagen
- Löschen von Änderungsvorlagen

Testen Sie die Vorlage für AWS verwaltete **`Hello World`** Änderungen

> Change ManagerÄnderung der Verfügbarkeit
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestehende Kunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Sie können die Beispieländerungsvorlage `AWS-HelloWorldChangeTemplate` verwenden, die das Beispiel-Automation-Runbook `AWS-HelloWorld` verwendet, um den Prüf- und Genehmigungsprozess zu testen, nachdem Sie die Einrichtung von Change Manager abgeschlossen haben, einem Tool in AWS Systems Manager. Diese Vorlage dient zum Testen oder Überprüfen der konfigurierten Berechtigungen, Genehmigungszuweisungen und des Genehmigungsprozesses. Die Genehmigung zur Verwendung dieser Änderungsvorlage in Ihrer Organisation oder Ihrem Konto wurde bereits von AWS bereitgestellt. Jeder Änderungsantrag, der auf dieser Änderungsvorlage basiert, muss jedoch weiterhin von Prüfern in Ihrer Organisation oder Ihrem Konto genehmigt werden.

Anstatt Änderungen an einer Ressource vorzunehmen, besteht das Ergebnis des mit dieser Vorlage verknüpften Runbook-Workflows darin, eine Meldung in der Ausgabe eines Automatisierungsschritts zu drucken.

Bevor Sie beginnen

Überprüfen Sie zu Beginn, ob Sie die folgenden Aufgaben ausgeführt haben:

- Wenn Sie Änderungen in einer Organisation verwalten AWS Organizations möchten, führen Sie die unter beschriebenen Aufgaben zur Einrichtung der Organisation durchEinrichten von Change Manager für eine Organisation (Management-Konto).
- Konfigurieren Sie Change Manager für Ihr delegiertes Administratorkonto oder ein einzelnes Konto, wie unter Konfigurieren von Change Manager-Optionen und bewährten Methoden beschrieben.

> **Note**
>
> Wenn Sie in Ihren Change Manager-Einstellungen die Bewährte–Methoden-Option Überwachen für alle Vorlagen erforderlich aktiviert haben, schalten Sie sie vorübergehend aus, während Sie die Änderungsvorlage Hello World testen.

Um die AWS verwaltete Hello World-Änderungsvorlage auszuprobieren

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Manager aus.
3. Wählen Sie Create request (Erstellen einer Anfrage).
4. Wählen Sie die Änderungsvorlage mit dem Namen `AWS-HelloWorldChangeTemplate` und wählen Sie danach Weiter.
5. Geben Sie für Name einen Namen für die Änderungsanforderung ein, mit der ihre Funktion leicht zu erkennen ist, z. B. **`MyChangeRequestTest`**.
6. Weitere Informationen zu den weiteren Schritten zum Erstellen der Änderungsanforderung finden Sie unter Erstellen von Änderungsanforderungen.

Nächste Schritte

Weitere Informationen zum Genehmigen von Änderungsanforderungen finden Sie unter Überprüfen und Genehmigen oder Ablehnen von Änderungsanforderungen.

Um den Status und die Ergebnisse Ihres Änderungsantrags anzuzeigen, wählen Sie den Namen Ihres Änderungsantrags auf der Registerkarte Anforderungen in Change Manager.

Erstellen von Änderungsvorlagen

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt

nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Eine Änderungsvorlage ist eine Sammlung von Konfigurationseinstellungen in Change Manager, die beispielsweise erforderliche Genehmigungen, verfügbare Runbooks und Benachrichtigungsoptionen für Änderungsanforderungen definiert.

Sie können Änderungsvorlagen für Ihre Vorgänge in Change Manager, einem Tool in AWS Systems Manager, über die Konsole erstellen, die Builder- und Editor-Optionen enthält, oder mit Befehlszeilentools.

Themen

- Über Genehmigungen in Ihren Änderungsvorlagen
- Erstellen von Änderungsvorlagen mit Builder
- Erstellen von Änderungsvorlagen mit dem Editor
- Erstellen von Änderungsvorlagen mit Befehlszeilenwerkzeugen

Über Genehmigungen in Ihren Änderungsvorlagen

Change ManagerÄnderung der Verfügbarkeit

AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestehende Kunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Für jede von Ihnen erstellte Änderungsvorlage können Sie bis zu fünf Genehmigungsebenen für daraus erstellte Änderungsanfragen angeben. Für jede dieser Ebenen können Sie bis zu fünf potenzielle Genehmiger benennen. Ein Genehmiger ist nicht auf einen einzelnen Benutzer beschränkt. Sie können auch eine IAM-Gruppe oder IAM-Rolle als einzelne Genehmiger angeben. Für IAM-Gruppen und IAM-Rollen können ein oder mehrere Benutzer, die zu der Gruppe oder Rolle gehören, Genehmigungen für den Erhalt der Gesamtzahl der Genehmigungen erteilen, die für eine Änderungsanforderung erforderlich sind. Sie können auch mehr Genehmiger angeben, als Ihre Änderungsvorlage erfordert.

Change Manager unterstützt zwei Hauptansätze für Genehmigungen: Genehmigungen pro Ebene und Genehmigungen pro Zeile. In manchen Situationen ist auch eine Kombination der beiden Typen möglich. Wir empfehlen, in Ihren Change Manager-Operationen nur Genehmigungen pro Ebene zu verwenden.

Per-level approvals

Empfohlen. Ab dem 23. Januar 2023 unterstützt Change Manager Genehmigungen pro Ebene. In diesem Modell geben Sie zunächst für jede Genehmigungsebene in Ihrer Änderungsvorlage an, wie viele Genehmigungen für diese Ebene erforderlich sind. Anschließend legen Sie mindestens so viele Genehmiger für die Ebene fest und können weitere Genehmiger angeben. Allerdings muss nur die von Ihnen festgelegte Anzahl von Genehmigern pro Ebene die Änderungsanfrage genehmigen. Sie können zum Beispiel fünf Genehmiger angeben, aber nur drei Genehmigungen verlangen.

Beispiele für diesen Genehmigungstyp in Konsolenansicht und JSON finden Sie unter the section called “Beispiel für eine Genehmigungskonfiguration pro Ebene”.

Per-line approvals

Unterstützt aus Gründen der Abwärtskompatibilität. Die ursprüngliche Version von Change Manager hat nur Genehmigungen pro Zeile unterstützt. In diesem Modell wird jeder für eine Genehmigungsebene angegebene Genehmiger als Genehmigungszeile dargestellt. Jeder Genehmiger musste eine Änderungsanfrage genehmigen, damit es auf dieser Ebene genehmigt werden konnte. Vor dem 23. Januar 2023 war dies das einzige unterstützte Modell für Genehmigungen. Änderungsvorlagen, die vor diesem Datum erstellt wurden, unterstützen weiterhin Genehmigungen pro Zeile, aber wir empfehlen, stattdessen Genehmigungen pro Ebene zu verwenden.

Beispiele für diesen Genehmigungstyp in Konsolenansicht und JSON finden Sie unter the section called “Beispiel für eine Genehmigungskonfiguration pro Zeile”.

Combined per-line and per-level approvals

Nicht empfohlen. In der Konsole unterstützt die Registerkarte Builder nicht mehr das Hinzufügen von Genehmigungen pro Zeile. In einigen Fällen kann es jedoch vorkommen, dass Sie in einer Änderungsvorlage sowohl Genehmigungen pro Zeile als auch pro Ebene erhalten. Dies kann vorkommen, wenn Sie eine Änderungsvorlage aktualisieren, die vor dem 23. Januar 2023 erstellt wurde, oder wenn Sie eine Änderungsvorlage erstellen oder aktualisieren, indem Sie ihren YAML-Inhalt manuell bearbeiten,

Beispiele für diesen Genehmigungstyp in Konsolenansicht und JSON finden Sie unter the section called “Beispiel für eine kombinierte Genehmigungskonfiguration pro Ebene und pro Zeile”.

> **Important**
>
> Es ist zwar möglich, eine Änderungsvorlage zu erstellen, die Genehmigungen pro Zeile und pro Ebene kombiniert, diese Konfiguration ist jedoch nicht empfohlen oder erforderlich. Die Genehmigungsart, die mehr Genehmigungen erfordert (Genehmigungen pro Zeile oder pro Ebene), hat Vorrang. Zum Beispiel:
>
> - Wenn eine Änderungsvorlage drei Genehmigungen pro Ebene, aber fünf Genehmigungen pro Zeile angibt, sind fünf Genehmigungen erforderlich.
> - Wenn eine Änderungsvorlage vier Genehmigungen pro Ebene, aber zwei Genehmigungen pro Zeile vorsieht, sind vier Genehmigungen erforderlich.

Sie können eine Ebene erstellen, die sowohl Genehmigungen pro Zeile als auch pro Ebene enthält, indem Sie den YAML- oder JSON-Inhalt manuell bearbeiten. Anschließend werden auf der Registerkarte Builder Steuerelemente zum Festlegen der erforderlichen Anzahl von Genehmigungen sowohl für die Ebene als auch für einzelne Zeilen angezeigt. Neue Ebenen, die Sie mithilfe der Konsole hinzufügen, unterstützen jedoch weiterhin nur Genehmigungskonfigurationen pro Ebene.

Benachrichtigungen und Ablehnungen von Änderungsanfragen

Amazon-SNS-Benachrichtigungen

Wenn eine Änderungsanfrage mit Ihrer Änderungsvorlage erstellt wird, werden Benachrichtigungen an Abonnenten des Amazon Simple Notification Service (Amazon SNS)-Themas gesendet, das für Genehmigungsbenachrichtigungen auf dieser Ebene vorgesehen ist. Sie können das Benachrichtigungsthema in der Änderungsvorlage angeben oder dem Benutzer, der die Änderungsanfrage erstellt, erlauben, eines anzugeben.

Nachdem die Mindestanzahl erforderlicher Genehmigungen auf einer Ebene empfangen wurde, werden Benachrichtigungen an Genehmiger gesendet, die das Amazon-SNS-Thema für die nächste Ebene abonniert haben, und so weiter.

> **Important**
>
> Stellen Sie sicher, dass die von Ihnen gemeinsam benannten IAM-Rollen, -Gruppen und -Benutzer über ausreichend Genehmigungen verfügen, um die von Ihnen angegebene Anzahl von Genehmigungen zu erfüllen. Wenn Sie beispielsweise nur eine einzelne IAM-Gruppe mit drei Benutzern als Genehmiger festlegen, können Sie nicht festlegen, dass auf dieser Ebene fünf Genehmigungen obligatorisch sind, sondern nur drei oder weniger.

Ablehnungen von Änderungsanfragen

Unabhängig davon, wie viele Genehmigungsebenen und Genehmiger Sie angeben, ist nur eine Ablehnung einer Änderungsanfrage erforderlich, um zu verhindern, dass der Runbook-Workflow für diese Anfrage ausgeführt wird.

Beispiele für Change Manager-Genehmigungsarten

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestehende Kunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Die folgenden Beispiele veranschaulichen die Konsolenansicht und den JSON-Inhalt für die drei Arten von Genehmigungstypen in Change Manager.

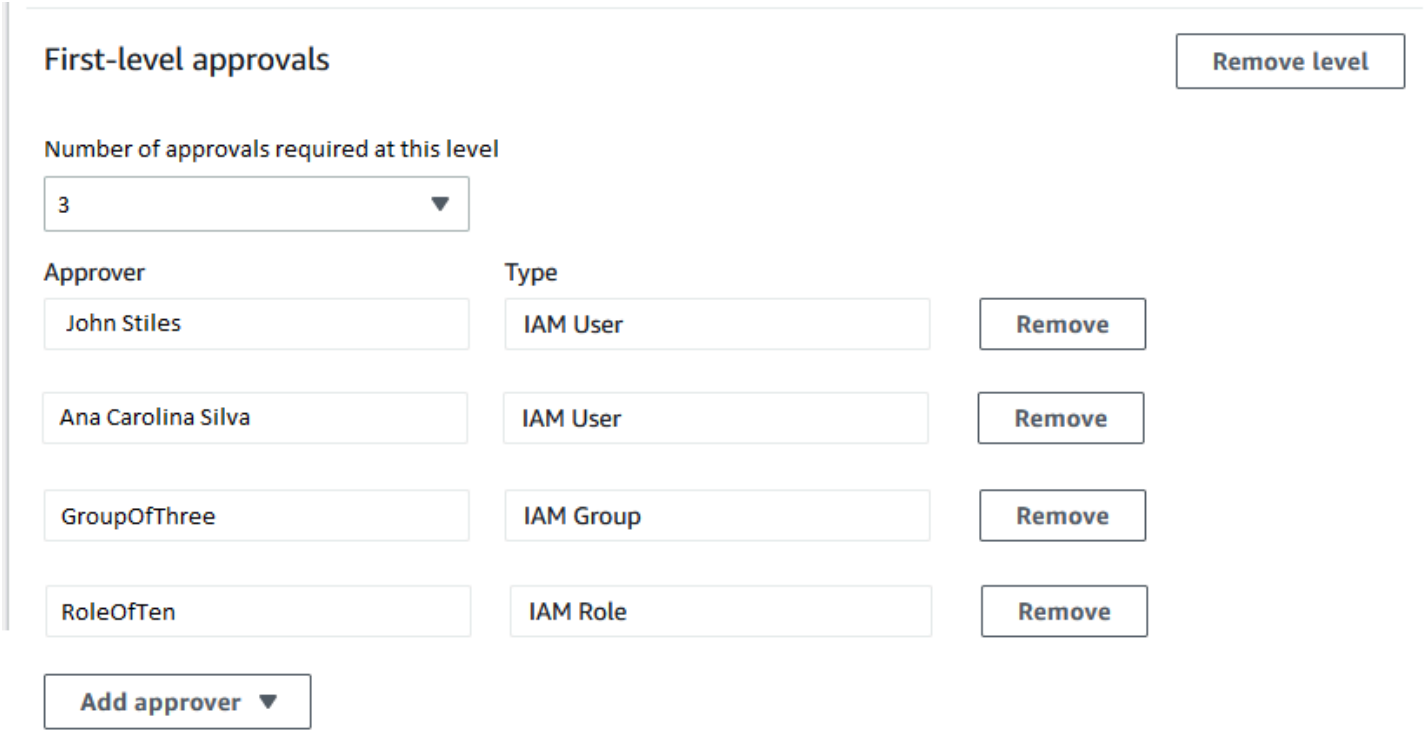
Themen

- Beispiel für eine Genehmigungskonfiguration pro Ebene
- Beispiel für eine Genehmigungskonfiguration pro Zeile
- Beispiel für eine kombinierte Genehmigungskonfiguration pro Ebene und pro Zeile

Beispiel für eine Genehmigungskonfiguration pro Ebene

Bei der im folgenden Image gezeigten Einrichtung der Genehmigungsebene pro Ebene sind drei Genehmigungen erforderlich. Diese Genehmigungen können aus einer beliebigen Kombination von IAM-Benutzern, Gruppen und Rollen stammen, die als Genehmiger angegeben sind. Zu den angegebenen Genehmigern gehören zwei IAM-Benutzer (John Stiles und Ana Carolina Silva), eine Benutzergruppe mit drei Mitgliedern (`GroupOfThree`) und eine Benutzerrolle, die zehn Benutzer repräsentiert (`RoleOfTen`).

Wenn alle drei Benutzer in der `GroupOfThree`-Gruppe die Änderungsanfrage genehmigen, wird sie für diese Ebene genehmigt. Es ist nicht erforderlich, eine Genehmigung von jedem Benutzer, Gruppe oder Rolle zu erhalten. Die Mindestanzahl an Genehmigungen kann von einer beliebigen Kombination festgelegter Genehmiger stammen. Wir empfehlen, für Ihre Change Manager-Operationen nur Genehmigungen pro Ebene zu verwenden.



Das folgende Beispiel veranschaulicht einen Teil des YAML-Codes für diese Konfiguration.

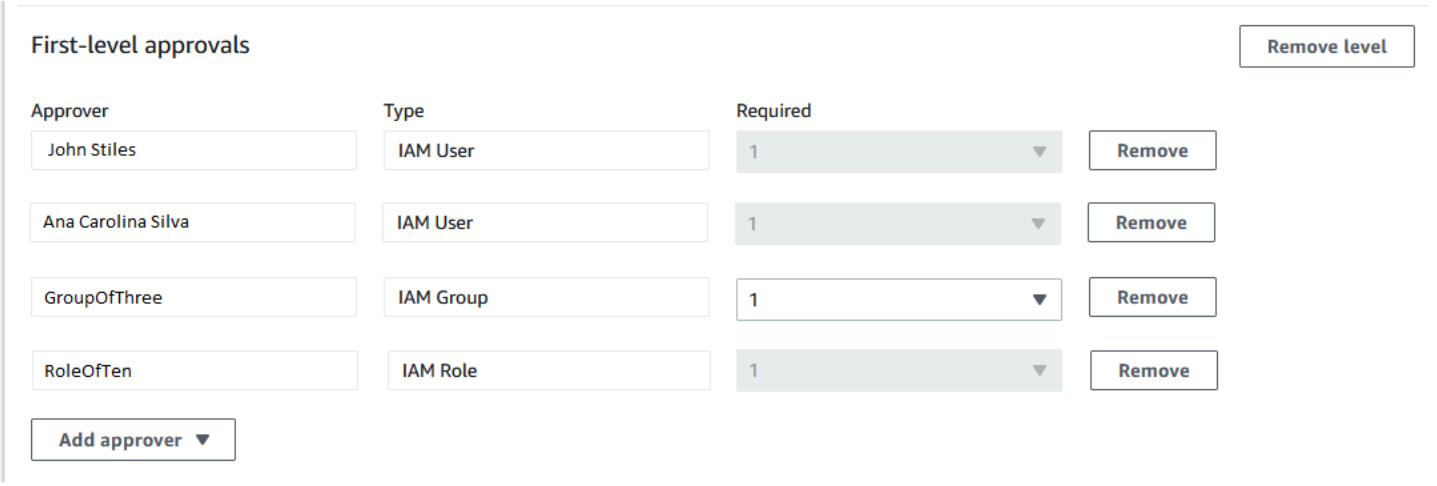
> **Note**
>
> Diese Version des YAML-Codes enthält eine zusätzliche Eingabe, `MinRequiredApprovals` (mit einem großen Anfangsbuchstaben `M`). Der Wert für diese Eingabe gibt an, wie viele Genehmigungen von allen verfügbaren Prüfern erforderlich sind. Beachten Sie

auch, dass der Wert `minRequiredApprovals` (in Kleinbuchstaben `m`) für jeden Genehmiger in der `Approvers`-Liste `0` (Null) ist. Dies zeigt an, dass der Genehmiger zu den Gesamtgenehmigungen beitragen kann, aber nicht dazu verpflichtet ist.

```
schemaVersion: "0.3"
emergencyChange: false
autoApprovable: false
mainSteps:
  - name: ApproveAction1
    action: aws:approve
    timeoutSeconds: 604800
    inputs:
      Message: Please approve this change request
      MinRequiredApprovals: 3
      EnhancedApprovals:
        Approvers:
          - approver: John Stiles
            type: IamUser
            minRequiredApprovals: 0
          - approver: Ana Carolina Silva
            type: IamUser
            minRequiredApprovals: 0
          - approver: GroupOfThree
            type: IamGroup
            minRequiredApprovals: 0
          - approver: RoleOfTen
            type: IamRole
            minRequiredApprovals: 0
templateInformation: >
  #### What is the purpose of this change?
    //truncated
```

Beispiel für eine Genehmigungskonfiguration pro Zeile

In der Konfiguration der Genehmigungsebene, die im folgenden Image dargestellt ist, werden vier Genehmiger angegeben. Dazu gehören zwei IAM-Benutzer (John Stiles und Ana Carolina Silva), eine Benutzergruppe mit drei Mitgliedern (`GroupOfThree`) und eine Benutzerrolle, die zehn Benutzer repräsentiert (`RoleOfTen`). Aus Gründen der Abwärtskompatibilität werden Genehmigungen pro Zeile unterstützt, jedoch nicht empfohlen.



Damit die Änderungsanfrage in dieser Genehmigungskonfiguration pro Zeile genehmigt werden kann, muss sie von allen genehmigenden Zeilen genehmigt werden:: John Stiles, Ana Carolina Silva, einem Mitglied der `GroupOfThree`-Gruppe und einem Mitglied der `RoleOfTen`-Rolle.

Das folgende Beispiel veranschaulicht einen Teil des YAML-Codes für diese Konfiguration.

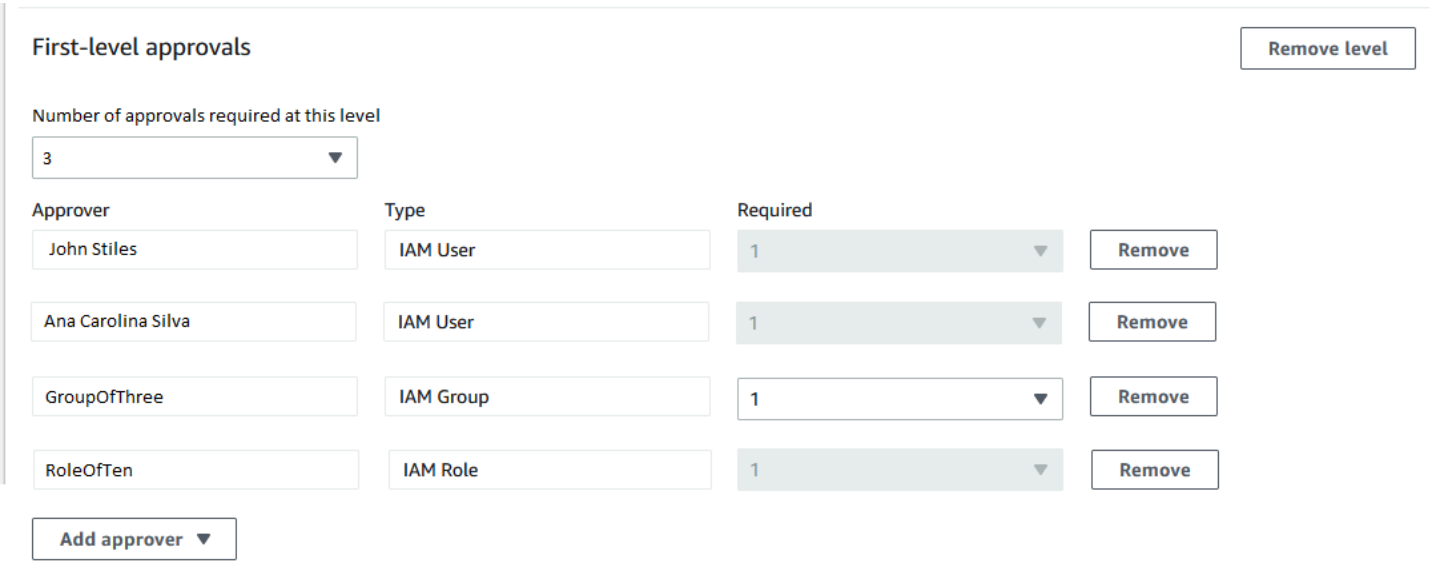
> **Note**
>
> Beachten Sie, dass der Wert für jeden `minRequiredApprovals`-Genehmiger 1 beträgt. Dies bedeutet, dass von jedem Genehmiger eine Genehmigung erforderlich ist.

```
schemaVersion: "0.3"
emergencyChange: false
autoApprovable: false
mainSteps:
  - name: ApproveAction1
    action: aws:approve
    timeoutSeconds: 10000
    inputs:
      Message: Please approve this change request
      EnhancedApprovals:
        Approvers:
          - approver: John Stiles
            type: IamUser
            minRequiredApprovals: 1
          - approver: Ana Carolina Silva
            type: IamUser
            minRequiredApprovals: 1
```

```
          - approver: GroupOfThree
            type: IamGroup
            minRequiredApprovals: 1
          - approver: RoleOfTen
            type: IamRole
            minRequiredApprovals: 1
executableRunBooks:
  - name: AWS-HelloWorld
    version: $DEFAULT
templateInformation: >
  #### What is the purpose of this change?
    //truncated
```

Beispiel für eine kombinierte Genehmigungskonfiguration pro Ebene und pro Zeile

Im folgenden Image werden bei der kombinierten Genehmigungskonfiguration pro Ebene und pro Zeile drei Genehmigungen für die Ebene angegeben, aber vier Genehmigungen für die Genehmigungen der einzelnen Positionen. Welcher Genehmigungstyp mehr Genehmigungen erfordert, hat Vorrang vor dem anderen, sodass für diese Konfiguration vier Genehmigungen erforderlich sind. Eine kombinierte Genehmigung pro Ebene und pro Linie wird nicht empfohlen.




```
schemaVersion: "0.3"
emergencyChange: false
autoApprovable: false
mainSteps:
  - name: ApproveAction1
    action: aws:approve
```

```
    timeoutSeconds: 604800
    inputs:
      Message: Please approve this change request
      MinRequiredApprovals: 3
      EnhancedApprovals:
        Approvers:
          - approver: John Stiles
            type: IamUser
            minRequiredApprovals: 1
          - approver: Ana Carolina Silva
            type: IamUser
            minRequiredApprovals: 1
          - approver: GroupOfThree
            type: IamGroup
            minRequiredApprovals: 1
          - approver: RoleOfTen
            type: IamRole
            minRequiredApprovals: 1
templateInformation: >
  #### What is the purpose of this change?
    //truncated
```

Themen

- Erstellen von Änderungsvorlagen mit Builder
- Erstellen von Änderungsvorlagen mit dem Editor
- Erstellen von Änderungsvorlagen mit Befehlszeilenwerkzeugen

Erstellen von Änderungsvorlagen mit Builder

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Mit dem Builder für Änderungsvorlagen in Change Manager, einem Tool in AWS Systems Manager, können Sie den in Ihrer Änderungsvorlage definierten Runbook-Workflow konfigurieren, ohne JSON-

oder YAML-Syntax verwenden zu müssen. Nachdem Sie Ihre Optionen festgelegt haben, konvertiert das System Ihre Eingabe in das YAML-Format, das Systems Manager zum Ausführen von Runbook-Workflows verwenden kann.

So erstellen Sie eine Änderungsvorlage mit Builder

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Manager aus.
3. Wählen Sie Create template (Vorlage erstellen) aus.
4. Geben Sie für Name einen Namen für die Vorlage ein, mit der ihre Funktion leicht zu erkennen ist, z. B. **`UpdateEC2LinuxAMI`**.
5. Gehen Sie im Abschnitt Details zur Änderungsvorlage wie folgt vor:

   - Geben Sie für Beschreibung eine kurze Erklärung ein, wie und wann die von Ihnen erstellte Änderungsvorlage verwendet werden soll.

     Mit dieser Beschreibung können Benutzer, die Änderungsanforderungen erstellen, feststellen, ob sie die richtige Änderungsvorlage verwenden. Es hilft denjenigen, die Änderungsanforderungen überprüfen, zu verstehen, ob die Anforderung genehmigt werden soll.
   - Geben Sie für Änderungsvorlagentyp an, ob Sie eine Standard- oder eine Notfalländerungsvorlage erstellen.

     Eine Vorlage für Notfalländerungen wird für Situationen verwendet, in denen eine Änderung auch dann vorgenommen werden muss, wenn die Änderungen ansonsten durch ein Ereignis im verwendeten Kalender blockiert werden AWS Systems Manager Change Calendar. Änderungsanforderungen, die aus einer Notfalländerungsvorlage erstellt wurden, müssen immer noch von den dafür vorgesehenen Genehmigern genehmigt werden, aber die angeforderten Änderungen können auch dann ausgeführt werden, wenn der Kalender gesperrt ist.
   - Geben Sie für Runbook-Optionen die Runbooks an, aus denen Benutzer beim Erstellen einer Änderungsanforderung auswählen können. Sie können ein einzelnes Runbook oder mehrere Runbooks hinzufügen. Alternativ können Sie Anforderern erlauben, anzugeben, welches Runbook verwendet werden soll. In jedem dieser Fälle kann nur ein Runbook in der Änderungsanforderung aufgenommen werden.

- Wählen Sie für Runbook die Namen der Runbooks und die Versionen dieser Runbooks aus, aus denen Benutzer für ihre Änderungsanforderungen auswählen können. Unabhängig davon, wie viele Runbooks Sie der Änderungsvorlage hinzufügen, kann pro Änderungsanforderung nur eines ausgewählt werden.

  Sie geben kein Runbook an, wenn Sie Jedes Runbook kann verwendet werden vorher bereits gewählt haben.

  > **Tip**
  >
  > Wählen Sie ein Runbook und eine Runbook-Version aus und wählen Sie dann View (Anzeigen), um den Inhalt des Runbooks in der Oberfläche von Systems Manager Documents zu prüfen.

6. Geben Sie im Abschnitt Vorlageninformationen mit Markdown Informationen für Benutzer ein, die Änderungsanforderungen von dieser Änderungsvorlage erstellen. Wir haben eine Reihe von Fragen bereitgestellt, die Sie für Benutzer, die Änderungsanforderungen erstellen, einfügen können, oder Sie können stattdessen andere Informationen und Fragen hinzufügen.

   > **Note**
   >
   > Markdown ist eine Markup-Sprache, die es Ihnen ermöglicht, Dokumente und einzelne Schritte innerhalb des Dokuments mit Beschreibungen im Wiki-Stil zu versehen. Weitere Informationen zur Verwendung von Markdown finden Sie unter Verwenden von Markdown in AWS.

   Wir empfehlen, Benutzern Fragen zur Beantwortung ihrer Änderungsanforderungen zur Verfügung zu stellen, damit Genehmiger entscheiden können, ob sie jede Änderungsanforderung erteilen möchten oder nicht, z. B. das Auflisten aller manuellen Schritte, die für die Ausführung als Teil der Änderung erforderlich sind, und ein Rollback-Plan.

   > **Tip**
   >
   > Wechseln Sie zwischen Vorschau ausblenden und Vorschau anzeigen, um zu sehen, wie der Inhalt während der Erstellung aussieht.

7. Im Abschnitt Change request approvals (Genehmigungen für Änderungsanträge) gehen Sie wie folgt vor:

   - (Optional) Wenn Sie zulassen möchten, dass Änderungsanforderungen, die aus dieser Änderungsvorlage erstellt wurden, automatisch ausgeführt werden, ohne von Genehmigern geprüft zu werden (mit Ausnahme von Change–Freeze-Ereignissen), wählen Sie Aktivieren der automatischen Genehmigung (Enable auto-approval).

     > **Note**
     >
     > Durch Aktivieren von automatischen Genehmigungen in einer Änderungsvorlage erhalten Benutzer die Option zur Umgehung von Überprüfern. Sie können weiterhin auswählen, ob Prüfer beim Erstellen einer Änderungsanforderung angegeben werden sollen. Daher müssen Sie in der Änderungsvorlage weiterhin Prüferoptionen angeben.

     > **Important**
     >
     > Wenn Sie die automatische Genehmigung für eine Änderungsvorlage aktivieren, können Benutzer Änderungsanforderungen mithilfe dieser Vorlage übermitteln, die vor der Ausführung nicht von Prüfern überprüft werden müssen (mit Ausnahme von Change-Freeze-Genehmigern). Wenn Sie einen bestimmten Benutzer, eine Gruppe oder IAM-Rolle daran hindern möchten, automatische Genehmigungsanforderungen zu senden, können Sie eine Bedingung in einer IAM-Richtlinie zu diesem Zweck verwenden. Weitere Informationen finden Sie unter Steuern des Zugriffs auf Runbook-Workflows für automatische Genehmigung.

   - Wählen Sie für Anzahl der auf dieser Ebene erforderlichen Genehmigungen die Anzahl der Genehmigungen aus, die aus dieser Änderungsvorlage erstellte Änderungsanfragen für diese Ebene erhalten müssen.
   - Um obligatorische Genehmiger der ersten Ebene hinzuzufügen, wählen Sie Genehmiger hinzufügen und wählen Sie eine der folgenden Optionen:
     - In der Vorlage angegebene Genehmiger - Wählen Sie einen oder mehrere Benutzer, Gruppen oder AWS Identity and Access Management -(IAM)-Rollen Ihres Kontos aus, um Änderungsanforderungen zu genehmigen, die mit dieser Änderungsvorlage erstellt wurden. Alle Änderungsanforderungen, die mit dieser Vorlage erstellt werden, müssen von jedem von Ihnen angegebenen Genehmiger geprüft und genehmigt werden.

- Request specified approvers (Angegebene Genehmiger anfordern) – Der Benutzer, der die Änderungsanforderung stellt, gibt Prüfer zum Zeitpunkt der Anforderung an und kann aus einer Liste von Benutzern in Ihrem Konto wählen.

  Die Nummer, die Sie im Feld Erforderlich eingeben, legt fest, wie viele Prüfer von einer Änderungsanforderung angegeben werden müssen, die diese Änderungsvorlage verwendet.

> **Important**
>
> Vor dem 23. Januar 2023 konnten auf der Registerkarte Builder nur Genehmigungen pro Zeile angegeben werden. Neue Änderungsvorlagen und neue Ebenen, die Sie mithilfe der Registerkarte Builder zu vorhandenen Änderungsvorlagen hinzufügen, unterstützen nur Genehmigungen pro Ebene. Wir empfehlen, in Ihren Change Manager-Operationen nur Genehmigungen pro Ebene zu verwenden.
> Weitere Informationen finden Sie unter Über Genehmigungen in Ihren Änderungsvorlagen.

- Gehen Sie für SNS-Thema zur Benachrichtigung von Genehmiger wie folgt vor:

  1. Wählen Sie eine der folgenden Optionen, um das Amazon Simple Notification Service (Amazon SNS)-Thema in Ihrem Konto anzugeben, das für das Senden von Benachrichtigungen an die Genehmiger verwendet werden soll, wenn eine Änderungsanforderung zur Überprüfung bereit ist:

     - Geben Sie einen SNS-Amazon-Ressourcenname (ARN) ein - Geben Sie für Thema-ARN einen ARN eines vorhandenen Amazon SNS Themas ein. Dieses Thema kann sich in jedem Konto Ihrer Organisation befinden.
     - Wählen Sie ein vorhandenes SNS-Thema - Wählen Sie für Target notification topic den ARN eines vorhandenen Amazon SNS-Themas in Ihrem aktuellen AWS-Konto. (Diese Option ist nicht verfügbar, wenn Sie in Ihrem aktuellen AWS-Konto und noch keine Amazon SNS SNS-Themen erstellt haben AWS-Region.)
     - SNS-Thema angeben, wenn die Änderungsanforderung erstellt wird - Der Benutzer, der eine Änderungsanforderung erstellt, kann das Amazon SNS-Thema angeben, das für Benachrichtigungen verwendet werden soll.

> **Note**
>
> Das von Ihnen ausgewählte Amazon SNS-Thema muss so konfiguriert werden, dass die gesendeten Benachrichtigungen und die Abonnenten, an die sie gesendet

werden, festgelegt werden. Seine Zugriffsrichtlinie muss auch Systems Manager Berechtigungen gewähren, damit Change Manager Benachrichtigungen senden kann. Weitere Informationen finden Sie unter Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen.

2. Wählen Sie Add notification (Benachrichtigung hinzufügen) aus.

8. (Optional) Um eine zusätzliche Ebene von Genehmigern hinzuzufügen, wählen Sie Add approval level (Genehmigungsebene hinzufügen) und wählen Sie zwischen vorlagenspezifischen Genehmigern und angeforderten Genehmigern für diese Ebene. Wählen Sie dann ein SNS-Thema aus, um diese Genehmiger zu benachrichtigen.

   Nachdem alle Genehmigungen von Genehmiger der ersten Ebene eingegangen sind, werden Genehmiger der zweiten Ebene benachrichtigt usw.

   Sie können maximal fünf Genehmigungsebenen in jeder Vorlage hinzufügen. So könnten Sie beispielsweise für die erste Stufe die Genehmigung von Benutzern in technischen Rollen und für die zweite Stufe die Genehmigung des Managers verlangen.

9. Geben Sie im Abschnitt Überwachung für den zu überwachenden CloudWatch Alarm den Namen eines CloudWatch Amazon-Alarms im aktuellen Konto ein, um den Fortschritt der Runbook-Workflows zu überwachen, die auf dieser Vorlage basieren.

   > **Tip**
   >
   > Um einen neuen Alarm zu erstellen oder die Einstellungen eines Alarms, den Sie angeben möchten, zu überprüfen, wählen Sie Die CloudWatch Amazon-Konsole öffnen. Informationen zum Arbeiten mit CloudWatch Alarmen finden Sie unter Verwenden von CloudWatch Alarmen im CloudWatch Amazon-Benutzerhandbuch.

10. Führen Sie im Abschnitt Notifications (Benachrichtigungen) folgende Schritte aus:

    1. Wählen Sie eine der folgenden Optionen aus, um das Amazon SNS-Thema in Ihrem Konto anzugeben, das zum Senden von Benachrichtigungen über Änderungsanforderungen verwendet werden soll, die mit dieser Änderungsvorlage erstellt werden:

       - Geben Sie einen SNS-Amazon-Ressourcenname (ARN) ein - Geben Sie für Thema-ARN einen ARN eines vorhandenen Amazon SNS Themas ein. Dieses Thema kann sich in jedem Konto Ihrer Organisation befinden.

- Wählen Sie ein vorhandenes SNS-Thema - Wählen Sie für Target notification topic den ARN eines vorhandenen Amazon SNS-Themas in Ihrem aktuellen AWS-Konto. (Diese Option ist nicht verfügbar, wenn Sie in Ihrem aktuellen AWS-Konto und noch keine Amazon SNS SNS-Themen erstellt haben AWS-Region.)

> **Note**
>
> Das von Ihnen ausgewählte Amazon SNS-Thema muss so konfiguriert werden, dass die gesendeten Benachrichtigungen und die Abonnenten, an die sie gesendet werden, festgelegt werden. Seine Zugriffsrichtlinie muss auch Systems Manager Berechtigungen gewähren, damit Change Manager Benachrichtigungen senden kann. Weitere Informationen finden Sie unter Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen.

2. Wählen Sie Add notification (Benachrichtigung hinzufügen) aus.

11. (Optional) Wenden Sie im Abschnitt „Tags“ ein oder mehrere name/value Tag-Schlüsselpaare auf die Änderungsvorlage an.

    Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Sie können beispielsweise eine Änderungsvorlage mit Tags versehen, um den Änderungstyp und die Umgebung, in der sie ausgeführt wird, zu identifizieren. In diesem Fall könnten Sie die folgenden name/value Schlüsselpaare angeben:

    - `Key=TaskType,Value=InstanceRepair`
    - `Key=Environment,Value=Production`

12. Klicken Sie auf Save and preview (speichern und Vorschau ansehen).

13. Überprüfen Sie die Details der Änderungsvorlage, die Sie gerade erstellen.

    Wenn Sie die Änderungsvorlage ändern möchten, bevor Sie sie zur Überprüfung einreichen, wählen Sie Actions (Aktionen).

    Wenn Sie mit dem Inhalt der Änderungsvorlage zufrieden sind, klicken Sie auf Submit for review (Zur Überprüfung einreichen). Die Benutzer in Ihrer Organisation oder Ihrem Konto, die als Vorlagenprüfer in der Registerkarte Einstellungen in Change Manager angegeben sind, werden benachrichtigt, dass eine neue Änderungsvorlage zur Überprüfung aussteht.

Wenn ein Amazon SNS-Thema für Änderungsvorlagen angegeben wurde, werden Benachrichtigungen gesendet, wenn die Änderungsvorlage abgelehnt oder genehmigt wird. Wenn Sie keine Benachrichtigungen zu dieser Änderungsvorlage erhalten, können Sie zu Change Manager später zurückkehren, um ihren Status zu überprüfen.

Erstellen von Änderungsvorlagen mit dem Editor

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Verwenden Sie die Schritte in diesem Thema, um eine Änderungsvorlage in Change Manager, einem Tool in AWS Systems Manager, zu konfigurieren, indem Sie JSON oder YAML eingeben, anstatt die Konsolensteuerung zu verwenden.

Erstellen einer Änderungsvorlage mit dem Editor

1. Wählen Sie im Navigationsbereich Change Manager aus.
2. Wählen Sie Create template (Vorlage erstellen) aus.
3. Geben Sie für Name einen Namen für die Vorlage ein, mit der ihre Funktion leicht zu erkennen ist, z. B. **`RestartEC2LinuxInstance`**.
4. Wählen Sie über Change template details (Vorlagendetails ändern) Editor.
5. Wählen Sie im Abschnitt Document Editor (Dokumenteneditor) die Option Edit (Bearbeiten) und geben Sie dann den JSON- oder YAML-Inhalt für Ihre Änderungsvorlage ein.

   Im Folgenden wird ein -Beispiel gezeigt.

   > **Note**
   >
   > Der Parameter `minRequiredApprovals` wird verwendet, um anzugeben, wie viele Prüfer auf einer bestimmten Ebene eine Änderungsanforderung genehmigen müssen, die mit dieser Vorlage erstellt wird.

Dieses Beispiel zeigt zwei Genehmigungsebenen. Sie können bis zu fünf Genehmigungsebenen angeben, aber nur eine Ebene ist erforderlich.
In der ersten Ebene muss der spezifische Benutzer „John-Doe“ jeden Änderungsantrag genehmigen. Danach müssen drei beliebige Mitglieder der IAM-Rolle `Admin` die Änderungsanforderung genehmigen.
Weitere Informationen zum Genehmigen von Änderungsvorlagen finden Sie unter Über Genehmigungen in Ihren Änderungsvorlagen.

YAML

```
description: >-
  This change template demonstrates the feature set available for creating
  change templates for Change Manager. This template starts a Runbook workflow
  for the Automation runbook called AWS-HelloWorld.
templateInformation: >
  ### Document Name: HelloWorldChangeTemplate

  ## What does this document do?

  This change template demonstrates the feature set available for creating
  change templates for Change Manager. This template starts a Runbook workflow
  for the Automation runbook called AWS-HelloWorld.

  ## Input Parameters

  * ApproverSnsTopicArn: (Required) Amazon Simple Notification Service ARN for
  approvers.

  * Approver: (Required) The name of the approver to send this request to.

  * ApproverType: (Required) The type of reviewer.
    * Allowed Values: IamUser, IamGroup, IamRole, SSOGroup, SSOUser

  ## Output Parameters

  This document has no outputs
schemaVersion: '0.3'
parameters:
  ApproverSnsTopicArn:
    type: String
    description: Amazon Simple Notification Service ARN for approvers.
```

```
  Approver:
    type: String
    description: IAM approver
  ApproverType:
    type: String
    description: >-
      Approver types for the request. Allowed values include IamUser, IamGroup,
      IamRole, SSOGroup, and SSOUser.
executableRunBooks:
  - name: AWS-HelloWorld
    version: '1'
emergencyChange: false
autoApprovable: false
mainSteps:
  - name: ApproveAction1
    action: 'aws:approve'
    timeoutSeconds: 3600
    inputs:
      Message: >-
        A sample change request has been submitted for your review in Change
        Manager. You can approve or reject this request.
      EnhancedApprovals:
        NotificationArn: '{{ ApproverSnsTopicArn }}'
        Approvers:
          - approver: John-Doe
            type: IamUser
            minRequiredApprovals: 1
  - name: ApproveAction2
    action: 'aws:approve'
    timeoutSeconds: 3600
    inputs:
      Message: >-
        A sample change request has been submitted for your review in Change
        Manager. You can approve or reject this request.
      EnhancedApprovals:
        NotificationArn: '{{ ApproverSnsTopicArn }}'
        Approvers:
          - approver: Admin
            type: IamRole
            minRequiredApprovals: 3
```

JSON

```
{
   "description": "This change template demonstrates the feature set available
 for creating
  change templates for Change Manager. This template starts a Runbook workflow
  for the Automation runbook called AWS-HelloWorld",
   "templateInformation": "### Document Name: HelloWorldChangeTemplate\n\n
    ## What does this document do?\n
    This change template demonstrates the feature set available for creating
 change templates for Change Manager.
    This template starts a Runbook workflow for the Automation runbook called
 AWS-HelloWorld.\n\n
    ## Input Parameters\n* ApproverSnsTopicArn: (Required) Amazon Simple
 Notification Service ARN for approvers.\n
    * Approver: (Required) The name of the approver to send this request to.\n
    * ApproverType: (Required) The type of reviewer.  * Allowed Values: IamUser,
 IamGroup, IamRole, SSOGroup, SSOUser\n\n
    ## Output Parameters\nThis document has no outputs\n",
   "schemaVersion": "0.3",
   "parameters": {
      "ApproverSnsTopicArn": {
         "type": "String",
         "description": "Amazon Simple Notification Service ARN for approvers."
      },
      "Approver": {
         "type": "String",
         "description": "IAM approver"
      },
      "ApproverType": {
         "type": "String",
         "description": "Approver types for the request. Allowed values include
 IamUser, IamGroup, IamRole, SSOGroup, and SSOUser."
      }
   },
   "executableRunBooks": [
      {
         "name": "AWS-HelloWorld",
         "version": "1"
      }
   ],
   "emergencyChange": false,
   "autoApprovable": false,
```

```
  "mainSteps": [
     {
        "name": "ApproveAction1",
        "action": "aws:approve",
        "timeoutSeconds": 3600,
        "inputs": {
           "Message": "A sample change request has been submitted for your
review in Change Manager. You can approve or reject this request.",
           "EnhancedApprovals": {
              "NotificationArn": "{{ ApproverSnsTopicArn }}",
              "Approvers": [
                 {
                    "approver": "John-Doe",
                    "type": "IamUser",
                    "minRequiredApprovals": 1
                 }
              ]
           }
        }
     },
       {
        "name": "ApproveAction2",
        "action": "aws:approve",
        "timeoutSeconds": 3600,
        "inputs": {
           "Message": "A sample change request has been submitted for your
review in Change Manager. You can approve or reject this request.",
           "EnhancedApprovals": {
              "NotificationArn": "{{ ApproverSnsTopicArn }}",
              "Approvers": [
                 {
                    "approver": "Admin",
                    "type": "IamRole",
                    "minRequiredApprovals": 3
                 }
              ]
           }
        }
     }
  ]
}
```

6. Klicken Sie auf Save and preview (speichern und Vorschau ansehen).

7. Überprüfen Sie die Details der Änderungsvorlage, die Sie gerade erstellen.

   Wenn Sie die Änderungsvorlage ändern möchten, bevor Sie sie zur Überprüfung einreichen, wählen Sie Actions (Aktionen).

   Wenn Sie mit dem Inhalt der Änderungsvorlage zufrieden sind, klicken Sie auf Submit for review (Zur Überprüfung einreichen). Die Benutzer in Ihrer Organisation oder Ihrem Konto, die als Vorlagenprüfer in der Registerkarte Einstellungen in Change Manager angegeben sind, werden benachrichtigt, dass eine neue Änderungsvorlage zur Überprüfung aussteht.

   Wenn ein Amazon Simple Notification Service (Amazon SNS)-Thema für Änderungsvorlagen angegeben wurde, werden Benachrichtigungen gesendet, wenn die Änderungsvorlage abgelehnt oder genehmigt wird. Wenn Sie keine Benachrichtigungen zu dieser Änderungsvorlage erhalten, können Sie zu Change Manager später zurückkehren, um ihren Status zu überprüfen.

Erstellen von Änderungsvorlagen mit Befehlszeilenwerkzeugen

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestehende Kunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Die folgenden Verfahren beschreiben, wie Sie das AWS Command Line Interface (AWS CLI) (unter LinuxmacOS, oderWindows Server) verwenden oder AWS Tools for Windows PowerShell eine Änderungsanforderung in Change Manager einem Tool in erstellen AWS Systems Manager.

Erstellen einer Änderungsvorlage:

1. Installieren und konfigurieren Sie das AWS CLI oder das AWS -Tools für PowerShell, falls Sie das noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.

2. Erstellen Sie eine JSON-Datei auf Ihrem lokalen Computer und geben Sie Ihr beispielsweise einen Namen wie `MyChangeTemplate.json`. Kopieren Sie anschließend den folgenden Inhalt in die Datei:

> **Note**
>
> Änderungsvorlagen verwenden eine Version von Schema 0.3, die nicht die gleiche Unterstützung wie für Automation-Runbooks enthält.

Im Folgenden wird ein -Beispiel gezeigt.

> **Note**
>
> Der Parameter `minRequiredApprovals` wird verwendet, um anzugeben, wie viele Prüfer auf einer bestimmten Ebene eine Änderungsanforderung genehmigen müssen, die mit dieser Vorlage erstellt wird.
> Dieses Beispiel zeigt zwei Genehmigungsebenen. Sie können bis zu fünf Genehmigungsebenen angeben, aber nur eine Ebene ist erforderlich.
> In der ersten Ebene muss der spezifische Benutzer „John-Doe“ jeden Änderungsantrag genehmigen. Danach müssen drei beliebige Mitglieder der IAM-Rolle `Admin` die Änderungsanforderung genehmigen.
> Weitere Informationen zum Genehmigen von Änderungsvorlagen finden Sie unter Über Genehmigungen in Ihren Änderungsvorlagen.

```
{
   "description": "This change template demonstrates the feature set available for
 creating
  change templates for Change Manager. This template starts a Runbook workflow
  for the Automation runbook called AWS-HelloWorld",
   "templateInformation": "### Document Name: HelloWorldChangeTemplate\n\n
    ## What does this document do?\n
    This change template demonstrates the feature set available for creating change
 templates for Change Manager.
    This template starts a Runbook workflow for the Automation runbook called AWS-
HelloWorld.\n\n
    ## Input Parameters\n* ApproverSnsTopicArn: (Required) Amazon Simple
 Notification Service ARN for approvers.\n
```

```
   * Approver: (Required) The name of the approver to send this request to.\n
   * ApproverType: (Required) The type of reviewer.  * Allowed Values: IamUser,
IamGroup, IamRole, SSOGroup, SSOUser\n\n
   ## Output Parameters\nThis document has no outputs\n",
  "schemaVersion": "0.3",
  "parameters": {
     "ApproverSnsTopicArn": {
        "type": "String",
        "description": "Amazon Simple Notification Service ARN for approvers."
     },
     "Approver": {
        "type": "String",
        "description": "IAM approver"
     },
     "ApproverType": {
        "type": "String",
        "description": "Approver types for the request. Allowed values include
IamUser, IamGroup, IamRole, SSOGroup, and SSOUser."
     }
  },
  "executableRunBooks": [
     {
        "name": "AWS-HelloWorld",
        "version": "1"
     }
  ],
  "emergencyChange": false,
  "autoApprovable": false,
  "mainSteps": [
     {
        "name": "ApproveAction1",
        "action": "aws:approve",
        "timeoutSeconds": 3600,
        "inputs": {
           "Message": "A sample change request has been submitted for your review
in Change Manager. You can approve or reject this request.",
           "EnhancedApprovals": {
              "NotificationArn": "{{ ApproverSnsTopicArn }}",
              "Approvers": [
                 {
                    "approver": "John-Doe",
                    "type": "IamUser",
                    "minRequiredApprovals": 1
                 }
```

```
                ]
            }
         }
      },
        {
         "name": "ApproveAction2",
         "action": "aws:approve",
         "timeoutSeconds": 3600,
         "inputs": {
            "Message": "A sample change request has been submitted for your review
 in Change Manager. You can approve or reject this request.",
            "EnhancedApprovals": {
               "NotificationArn": "{{ ApproverSnsTopicArn }}",
               "Approvers": [
                  {
                     "approver": "Admin",
                     "type": "IamRole",
                     "minRequiredApprovals": 3
                  }
               ]
            }
         }
      }
   ]
}
```

3. Führen Sie den folgenden Befehl aus, um die Änderungsvorlage zu erstellen.

   Linux & macOS

   ```
   aws ssm create-document \
       --name MyChangeTemplate \
       --document-format JSON \
       --document-type Automation.ChangeTemplate \
       --content file://MyChangeTemplate.json \
       --tags Key=tag-key,Value=tag-value
   ```

   Windows

   ```
   aws ssm create-document ^
       --name MyChangeTemplate ^
       --document-format JSON ^
       --document-type Automation.ChangeTemplate ^
   ```

```
    --content file://MyChangeTemplate.json ^
    --tags Key=tag-key,Value=tag-value
```

PowerShell

```
$json = Get-Content -Path "C:\path\to\file\MyChangeTemplate.json" | Out-String
New-SSMDocument `
    -Content $json `
    -Name "MyChangeTemplate" `
    -DocumentType "Automation.ChangeTemplate" `
    -Tags "Key=tag-key,Value=tag-value"
```

Weitere Informationen zu anderen Optionen, die Sie angeben können, finden Sie unter create-document.

Das System gibt unter anderem folgende Informationen zurück

```
{
   "DocumentDescription":{
      "CreatedDate":1.585061751738E9,
      "DefaultVersion":"1",
      "Description":"Use this template to update an EC2 Linux AMI. Requires one
      approver specified in the template and an approver specified in the
 request.",
      "DocumentFormat":"JSON",
      "DocumentType":"Automation",
      "DocumentVersion":"1",
      "Hash":"0d3d879b3ca072e03c12638d0255ebd004d2c65bd318f8354fcde820dEXAMPLE",
      "HashType":"Sha256",
      "LatestVersion":"1",
      "Name":"MyChangeTemplate",
      "Owner":"123456789012",
      "Parameters":[
         {
            "DefaultValue":"",
            "Description":"Level one approvers",
            "Name":"LevelOneApprovers",
            "Type":"String"
         },
         {
            "DefaultValue":"",
```

```
            "Description":"Level one approver type",
            "Name":"LevelOneApproverType",
            "Type":"String"
         },
   "cloudWatchMonitors": {
      "monitors": [
         "my-cloudwatch-alarm"
      ]
   }
      ],
      "PlatformTypes":[
         "Windows",
         "Linux"
      ],
      "SchemaVersion":"0.3",
      "Status":"Creating",
      "Tags":[

      ]
   }
}
```

Die Benutzer in Ihrer Organisation oder Ihrem Konto, die als Vorlagenprüfer in der Registerkarte Einstellungen in Change Manager angegeben sind, werden benachrichtigt, dass eine neue Änderungsvorlage zur Überprüfung aussteht.

Wenn ein Amazon Simple Notification Service (Amazon SNS)-Thema für Änderungsvorlagen angegeben wurde, werden Benachrichtigungen gesendet, wenn die Änderungsvorlage abgelehnt oder genehmigt wird. Wenn Sie keine Benachrichtigungen zu dieser Änderungsvorlage erhalten, können Sie zu Change Manager später zurückkehren, um ihren Status zu überprüfen.

Überprüfen und Genehmigen oder Ablehnen von Änderungsvorlagen

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestehende Kunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Wenn Sie als Überprüfer für Änderungsvorlagen in Change Manager, einem Tool in AWS Systems Manager, angegeben sind, werden Sie benachrichtigt, wenn eine neue Änderungsvorlage oder eine neue Version einer Änderungsvorlage auf Ihre Überprüfung wartet. Ein Amazon Simple Notification Service (Amazon SNS)-Thema sendet die Benachrichtigungen.

> **Note**
>
> Diese Funktionalität hängt davon ab, ob Ihr Konto so konfiguriert wurde, dass ein Amazon SNS-Thema verwendet wird, um Benachrichtigungen zur Überprüfung von Änderungsvorlagen zu senden. Informationen zum Festlegen eines Themas für die Benachrichtigung eines Vorlagenprüfers finden Sie unter Aufgabe 1: Konfigurieren von Change Manager-Benutzeridentitätsverwaltung und Vorlagenprüfern.

Um die Änderungsvorlage zu überprüfen, folgen Sie dem Link in Ihrer Benachrichtigung, melden Sie sich bei der AWS-Managementkonsole an und folgen Sie den Schritten in diesem Verfahren.

Überprüfen und Genehmigen oder Ablehnen einer Änderungsvorlage

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Manager aus.
3. Wählen Sie im Abschnitt Change templates (Änderungsvorlagen) unten auf der Registerkarte Overview (Übersicht) die Nummer in Pending review (ausstehende Prüfung).
4. Suchen Sie in der Liste Change templates (Änderungsvorlagen) den Namen der zu überprüfenden Änderungsvorlage und wählen Sie ihn aus.
5. Überprüfen Sie auf der Übersichtsseite den vorgeschlagenen Inhalt der Änderungsvorlage und führen Sie einen der folgenden Schritte aus:

   - Um die Änderungsvorlage zu genehmigen, wodurch sie in Änderungsanforderungen verwendet werden kann, wählen Sie Approve (Genehmigen).
   - Um die Änderungsvorlage abzulehnen, wodurch ihre Verwendung in Änderungsanforderungen verhindert wird, wählen Sie Reject (Ablehnen).

Löschen von Änderungsvorlagen

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

In diesem Thema wird beschrieben, wie Sie Vorlagen löschen, die Sie in Change Manager, einem Tool in Systems Manager, erstellt haben. Wenn Sie Change Manager für eine Organisation verwenden, wird dieses Verfahren in Ihrem delegierten Administratorkonto durchgeführt.

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Manager aus.
3. Wählen Sie die Registerkarte Templates (Vorlagen) aus.
4. Wählen Sie den Namen der zu löschenden Vorlage aus.
5. Wählen Sie Actions, Delete template (Aktionen, Vorlage löschen).
6. Geben Sie in das Bestätigungsfeld das Wort **DELETE** ein und wählen Sie dann Delete (Löschen).

Verwenden von Änderungsanforderungen

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestehende Kunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Eine Änderungsanforderung ist eine Anforderung Change Manager zur Ausführung eines Automatisierungs-Runbooks, das eine oder mehrere Ressourcen in Ihren AWS oder lokalen Umgebungen aktualisiert. Ein Änderungsantrag wird mit einer Änderungsvorlage erstellt.

Wenn Sie eine Änderungsanforderung in Change Manager erstellen, einem Tool in AWS Systems Manager, muss mindestens ein Genehmiger in Ihrer Organisation oder Ihrem Konto die Anforderung überprüfen und genehmigen. Ohne die erforderlichen Genehmigungen kann der Runbook-Workflow, der die angeforderten Änderungen vornimmt, nicht ausgeführt werden.

Themen

- Erstellen von Änderungsanforderungen
- Überprüfen und Genehmigen oder Ablehnen von Änderungsanforderungen

Erstellen von Änderungsanforderungen

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Wenn Sie eine Änderungsanforderung in Change Manager erstellen, einem Tool in AWS Systems Manager, führt die von Ihnen ausgewählte Änderungsvorlage in der Regel folgende Aktionen aus:

- Bestimmt Genehmiger für die Änderungsanforderung oder gibt an, wie viele Genehmigungen erforderlich sind
- Gibt das Amazon Simple Notification Service (Amazon SNS)-Thema an, das verwendet werden soll, um Genehmiger über Ihre Änderungsanforderung zu benachrichtigen
- Spezifiziert einen CloudWatch Amazon-Alarm zur Überwachung des Runbook-Workflows für die Änderungsanforderung
- Identifiziert, aus welchen Automation-Runbooks Sie wählen können, um die angeforderte Änderung vorzunehmen

In einigen Fällen kann eine Änderungsvorlage konfiguriert werden, sodass Sie Ihr eigenes Automation-Runbook angeben und angeben, wer die Anforderung überprüfen und genehmigen soll.

> **Important**
>
> Wenn Sie den Change Manager in einer Organisation verwenden, empfehlen wir, Änderungen immer über das delegierte Administratorkonto vorzunehmen. Obwohl Sie Änderungen von anderen Konten in der Organisation vornehmen, werden diese Änderungen nicht im delegierten Administratorkonto gemeldet oder können nicht angezeigt werden.

Themen

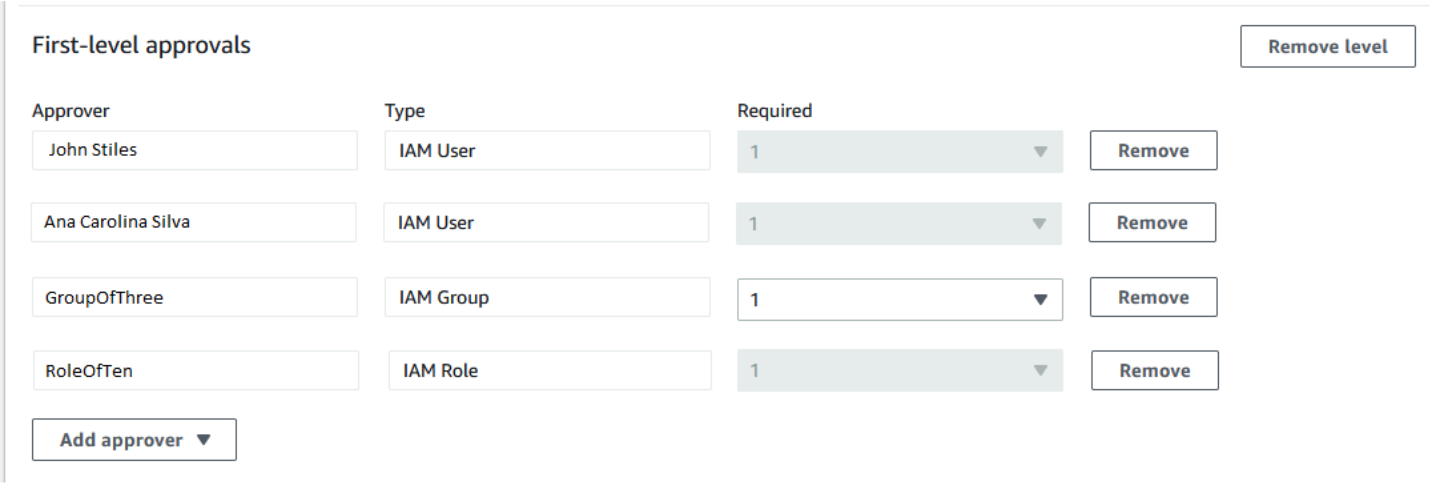
- Über die Genehmigung von Änderungsanfragen
- Erstellen von Änderungsanforderungen (Konsole)
- Erstellen von Änderungsanforderungen (AWS CLI)

Über die Genehmigung von Änderungsanfragen

Abhängig von den in einer Änderungsvorlage festgelegten Anforderungen können Änderungsanfragen, die Sie auf dieser Grundlage erstellen, Genehmigungen von bis zu fünf Ebenen erfordern, bevor der Runbook-Workflow für die Anfrage ausgeführt werden kann. Für jede dieser Ebenen kann der Ersteller der Vorlage bis zu fünf potenzielle Genehmiger angeben. Ein Genehmiger ist nicht auf einen einzelnen Benutzer beschränkt. Ein Genehmiger in diesem Sinne kann auch eine IAM-Gruppe oder IAM-Rolle sein. Für IAM-Gruppen und IAM-Rollen können ein oder mehrere Benutzer, die zu der Gruppe oder Rolle gehören, Genehmigungen für den Erhalt der Gesamtzahl der Genehmigungen erteilen, die für eine Änderungsanforderung erforderlich sind. Ersteller von Vorlagen können auch mehr Genehmiger angeben, als die Änderungsvorlage erfordert.

Ursprüngliche Genehmigungsworkflows und aktualisierte and/or Genehmigungen

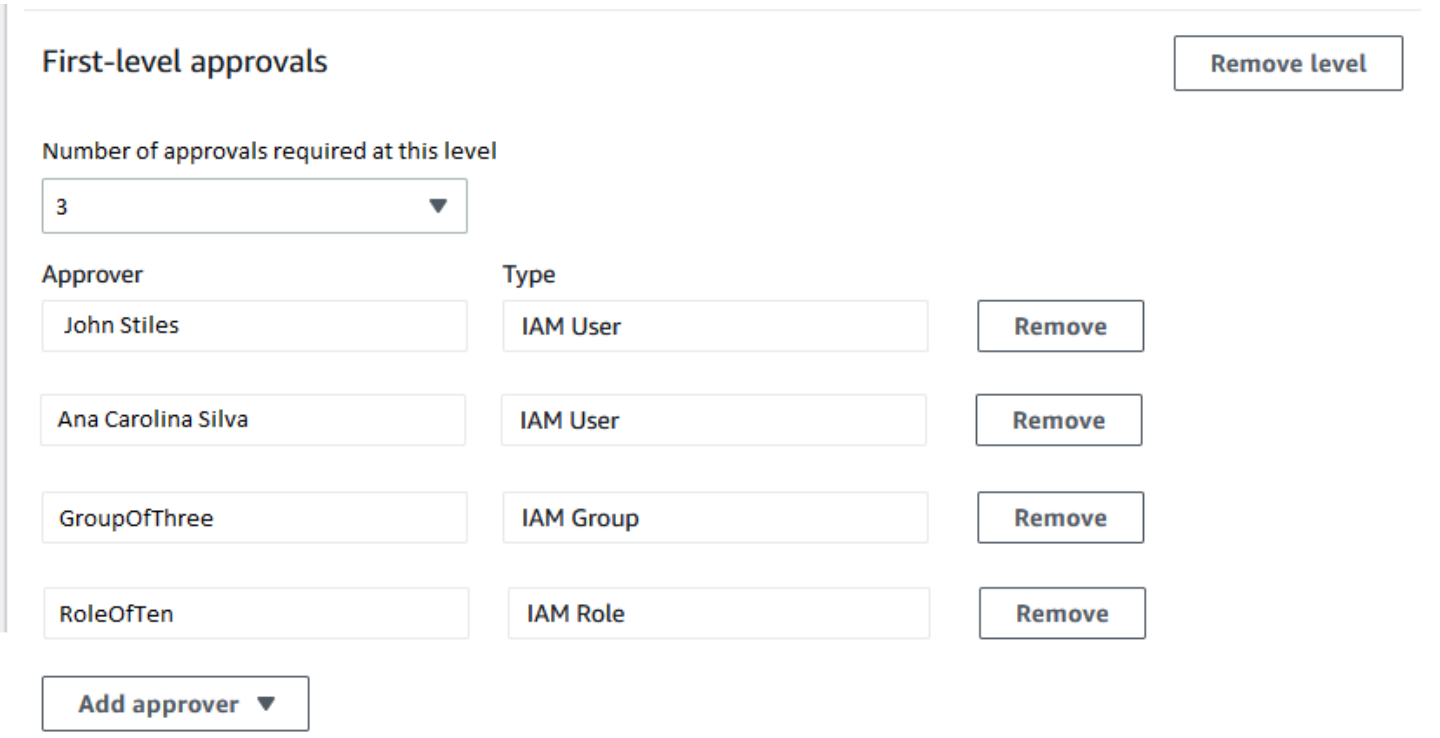
Bei Verwendung von Änderungsvorlagen, die vor dem 23. Januar 2023 erstellt wurden, muss eine Genehmigung von jedem angegebenen Genehmiger eingeholt werden, damit die Änderungsanfrage auf dieser Ebene genehmigt werden kann. Beispielsweise sind in der Einrichtung der Genehmigungsebene, die im folgenden Image angezeigt werden, vier Genehmiger angegeben. Zu den angegebenen Genehmigern gehören zwei Benutzer (John Stiles und Ana Carolina Silva), eine Benutzergruppe mit drei Mitgliedern (GroupOfThree) und eine Benutzerrolle, die zehn Benutzern () entspricht. RoleOfTen



Damit die Änderungsanfrage auf dieser Ebene genehmigt werden kann, muss sie von John Stiles, Ana Carolina Silva, einem Mitglied der `GroupOfThree`-Gruppe und einem Mitglied der `RoleOfTen`-Rolle genehmigt werden.

Unter Verwendung von Änderungsvorlagen, die am oder nach dem 23. Januar 2023 erstellt wurden, können Vorlagenersteller für jede Genehmigungsebene eine Gesamtzahl der erforderlichen Genehmigungen angeben. Diese Genehmigungen können aus einer beliebigen Kombination von Benutzern, Gruppen und Rollen stammen, die als Genehmiger angegeben sind. Eine Änderungsvorlage könnte nur eine Genehmigung für eine Ebene erfordern, aber beispielsweise zwei einzelne Benutzer, zwei Gruppen und eine Rolle als potenzielle Genehmiger angeben.

Beispielsweise sind in der Einrichtung der Genehmigungsebene, die im folgenden Image angezeigt werden, pro Ebene drei Genehmigungen erforderlich. Zu den angegebenen Genehmigern gehören zwei Benutzer (John Stiles und Ana Carolina Silva), eine Benutzergruppe mit drei Mitgliedern (`GroupOfThree`) und eine Benutzerrolle, die zehn Benutzer repräsentiert (`RoleOfTen`).



Wenn alle drei Benutzer in der `GroupOfThree`-Gruppe die Änderungsanfrage genehmigen, wird er für diese Ebene genehmigt. Es ist nicht erforderlich, eine Genehmigung von jedem Benutzer, Gruppe oder Rolle zu erhalten. Die Mindestanzahl an Genehmigungen kann von einer beliebigen Kombination potenzieller Genehmiger stammen.

Wenn Ihre Änderungsanfrage erstellt wird, werden Benachrichtigungen an die Abonnenten des Amazon-SNS-Themas gesendet, das für Genehmigungsbenachrichtigungen auf dieser Ebene angegeben wurde. Der Ersteller der Änderungsvorlage hat möglicherweise das zu verwendende Benachrichtigungsthema angegeben oder Ihnen erlaubt, eines anzugeben.

Nachdem die Mindestanzahl erforderlicher Genehmigungen auf einer Ebene eingegangen ist, werden Benachrichtigungen an Genehmiger gesendet, die das Amazon-SNS-Thema für die nächste Ebene abonniert haben, und so weiter.

Unabhängig davon, wie viele Genehmigungsebenen und Genehmiger angegeben sind, ist nur eine Ablehnung einer Änderungsanfrage erforderlich, um zu verhindern, dass der Runbook-Workflow für diese Anforderung ausgeführt wird.

Erstellen von Änderungsanforderungen (Konsole)

Im Folgenden wird beschrieben, wie Sie eine Änderungsanforderung mit Hilfe der Systems Manager-Konsole erstellen.

So erstellen Sie eine Änderungsanforderung (Konsole)

1. Öffnen Sie die Konsole unter AWS Systems Manager . https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Change Manager aus.
3. Wählen Sie Create request (Erstellen einer Anfrage).
4. Suchen Sie nach einer Änderungsvorlage, die Sie für diese Änderungsanforderung verwenden möchten, und wählen Sie sie aus.
5. Wählen Sie Weiter aus.
6. Geben Sie für Name einen Namen für die Änderungsanforderung ein, mit der ihre Funktion leicht zu erkennen ist, z. B. **`UpdateEC2LinuxAMI-us-east-2`**.
7. Wählen Sie für Runbook das Runbook aus, das Sie für die gewünschte Änderung verwenden möchten.

   > **Note**
   >
   > Wenn die Option zum Auswählen eines Runbooks nicht verfügbar ist, hat der Autor der Änderungsvorlage angegeben, welches Runbook verwendet werden muss.

8. Verwenden Sie für Change request information (Informationen zu Änderungsanforderung) Markdown, um zusätzliche Informationen zur Änderungsanforderung bereitzustellen, damit Prüfer entscheiden können, ob sie die Änderungsanforderung genehmigen oder ablehnen möchten. Der Autor der Vorlage, die Sie verwenden, hat möglicherweise Anweisungen oder Fragen zur Beantwortung bereitgestellt.

   > **Note**
   >
   > Markdown ist eine Markup-Sprache, die es Ihnen ermöglicht, Dokumente und einzelne Schritte innerhalb des Dokuments mit Beschreibungen im Wiki-Stil zu versehen. Weitere Informationen zur Verwendung von Markdown finden Sie unter Verwenden von Markdown in AWS.

9. Wählen Sie im Abschnitt Workflow start time (Workflow-Startzeit) eine der folgenden Optionen:

   - Run the operation at a scheduled time (Ausführen des Vorgangs zu einem geplanten Zeitpunkt) - Geben Sie für Requested start time (Gewünschte Startzeit) das Datum und die Uhrzeit ein, die Sie für die Ausführung des Runbook-Workflows für diesen Auftrag

vorschlagen. Geben Sie bei Estimated end time (Geschätzte Endzeit) das Datum und die Uhrzeit ein, zu der der Runbook-Workflow voraussichtlich abgeschlossen sein wird. (Diese Zeit ist nur eine Schätzung, die Sie Prüfern zur Verfügung stellen.)

> **Tip**
>
> Wählen Sie View Change Calendar (Änderungskalender anzeigen), um zu prüfen, ob für die von Ihnen angegebene Zeit Sperrungen vorliegen.

- Führen Sie den Vorgang so schnell wie möglich nach der Genehmigung aus. - Wenn die Änderungsanforderung genehmigt wird, wird der Runbook-Workflow ausgeführt, sobald ein Zeitraum nicht eingeschränkt ist, in dem Änderungen vorgenommen werden können.

10. Im Abschnitt Change request approvals (Genehmigungen für Änderungsanträge) gehen Sie wie folgt vor:

    1. Wenn Approval type (Genehmigungstyp)-Optionen angezeigt werden, wählen Sie eine der folgenden Optionen:

       - Automatic approval (Automatische Genehmigung) - Die ausgewählte Änderungsvorlage ist so konfiguriert, dass Änderungsanforderungen automatisch ausgeführt werden können - ohne Prüfung durch Genehmiger. Fahren Sie fort mit Schritt 11.

         > **Note**
         >
         > Die in den IAM-Richtlinien angegebenen Berechtigungen, die Ihre Verwendung von Systems Manager regeln, dürfen Sie nicht daran hindern, Änderungsanforderungen zur automatischen Genehmigung zu übermitteln, damit sie automatisch ausgeführt werden können.

       - Specify approvers (Angeben von Genehmigern) - Sie müssen einen oder mehrere Benutzer, Gruppen oder IAM-Rollen hinzufügen, um diese Änderungsanforderung zu überprüfen und zu genehmigen.

         > **Note**
         >
         > Sie können Überprüfer auch dann angeben, wenn die in den IAM-Richtlinien, die Ihre Verwendung von Systems Manager regeln, festgelegten Berechtigungen

die Ausführung von Änderungsanforderungen mit automatischer Genehmigung erlauben.

2. Wählen Sie Genehmiger hinzufügen und wählen Sie dann einen oder mehrere Benutzer, Gruppen oder AWS Identity and Access Management (IAM-) Rollen aus der Liste der verfügbaren Prüfer aus.

> **Note**
>
> Möglicherweise sind bereits ein oder mehrere Genehmiger angegeben. Dies bedeutet, dass obligatorische Genehmiger bereits in der von Ihnen ausgewählten Änderungsvorlage angegeben sind. Diese Genehmiger können nicht aus der Anforderung entfernt werden. Wenn die Schaltfläche Genehmiger hinzufügen nicht verfügbar ist, lässt die von Ihnen ausgewählte Vorlage nicht zu, dass zusätzliche Prüfer zu Anfragen hinzugefügt werden.

Weitere Informationen zum Genehmigen von Änderungsanfragen finden Sie unter Über die Genehmigung von Änderungsanfragen.

3. Wählen Sie unter SNS topic to notify approvers (SNS-Thema zur Benachrichtigung von Genehmigern) eine der folgenden Optionen, um das Amazon SNS-Thema in Ihrem Konto anzugeben, das zum Senden von Benachrichtigungen an die Genehmigenden verwendet werden soll, die Sie zu dieser Änderungsanforderung hinzufügen.

> **Note**
>
> Wenn die Option zum Angeben eines Amazon SNS-Themas nicht verfügbar ist, gibt die von Ihnen ausgewählte Änderungsvorlage bereits das zu verwendende Amazon SNS-Thema an.

- Geben Sie einen SNS-Amazon-Ressourcenname (ARN) ein - Geben Sie für Thema-ARN einen ARN eines vorhandenen Amazon SNS Themas ein. Dieses Thema kann sich in jedem Konto Ihrer Organisation befinden.
- Wählen Sie ein vorhandenes SNS-Thema - Wählen Sie für Target notification topic den ARN eines vorhandenen Amazon SNS-Themas in Ihrem aktuellen Konto. (Diese Option

ist nicht verfügbar, wenn Sie in Ihrem aktuellen AWS-Konto und noch keine Amazon SNS SNS-Themen erstellt haben AWS-Region.)

> **Note**
>
> Das von Ihnen ausgewählte Amazon SNS-Thema muss so konfiguriert werden, dass die gesendeten Benachrichtigungen und die Abonnenten, an die sie gesendet werden, festgelegt werden. Seine Zugriffsrichtlinie muss auch Systems Manager Berechtigungen gewähren, damit Change Manager Benachrichtigungen senden kann. Weitere Informationen finden Sie unter Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen.

4. Wählen Sie Add notification (Benachrichtigung hinzufügen) aus.

11. Wählen Sie Weiter aus.
12. Wählen Sie für IAM role (IAM-Rolle) eine IAM-Rolle in Ihrem aktuellen Konto aus, die über die erforderlichen Berechtigungen zur Ausführung der Runbooks verfügt, die für diese Änderungsanforderung angegeben sind.

    Diese Rolle wird auch als Dienstrolle bezeichnet oder Übernahmerolle für Automation. Weitere Informationen über diese Rolle finden Sie unter Einrichten der Automatisierung.
13. Wählen Sie im Abschnitt Deployment location (Standort der Bereitstellung) eine der folgenden Optionen:

> **Note**
>
> Wenn Sie AWS-Konto nur Change Manager mit einer einzigen Person und nicht mit einer Organisation AWS Organizations, in der Sie eingerichtet sind, verwenden, müssen Sie keinen Bereitstellungsort angeben.

- Apply change to this account (Änderung auf dieses Konto anwenden)— Der Runbook-Workflow wird nur im aktuellen Konto ausgeführt. Für eine Organisation bedeutet dies das delegierte Administratorkonto.
- Änderung auf mehrere Organisationseinheiten anwenden (OUs) — Gehen Sie wie folgt vor:
  1. Geben Sie für Konten und Organisationseinheiten (OUs) die ID eines Mitgliedskontos in Ihrer Organisation im Format **`123456789012`** oder die ID einer Organisationseinheit im Format ein**`o-o96EXAMPLE`**.

2. (Optional) Geben Sie für Execution role name (Name der Ausführungsrolle) den Namen der IAM-Rolle im Zielkonto oder Organisationseinheit ein, die über die erforderlichen Berechtigungen zum Ausführen der Runbooks verfügt, die für diese Änderungsanforderung angegeben sind. Alle Konten in einer von Ihnen angegebenen Organisationseinheit sollten für diese Rolle denselben Namen verwenden.
3. (Optional) Wählen Sie Add another target location (Weiteres Ziel hinzufügen) für jedes zusätzliche Konto oder jede Organisationseinheit, die Sie angeben möchten, und wiederholen Sie die Schritte a und b.
4. Wählen Sie für Target die Region aus AWS-Region, in der die Änderung vorgenommen werden soll, z. B. `Ohio (us-east-2)` für die Region USA Ost (Ohio).
5. Erweitern Sie den Reiter Rate control (Ratenregelung).

   Geben Sie für Concurrency (Gleichzeitigkeit) eine Zahl ein, und wählen Sie dann aus der Liste aus, ob dies die Anzahl oder den Prozentsatz der Konten darstellt, in denen der Runbook-Workflow gleichzeitig ausgeführt werden kann.

   Geben Sie für Error threshold (Schwellenwert-Fehler) eine Zahl ein und wählen Sie dann aus der Liste aus, ob dies die Anzahl oder den Prozentsatz der Konten darstellt, bei denen der Runbook-Workflow fehlschlagen kann, bevor der Vorgang beendet wird.

14. Gehen Sie im Abschnitt Deployment targets (Bereitstellungsziele) wie folgt vor:

    1. Wählen Sie eine der folgenden Optionen:

       - Single resource (Einzelne Ressource) - Die Änderung soll nur für eine Ressource vorgenommen werden. Zum Beispiel, ein einzelner Knoten oder ein einzelnes Amazon Machine Image (AMI), je nach der in den Runbooks für diesen Änderungsauftrag definierten Operation.
       - Multiple resources (Mehrere Ressourcen) - Wählen Sie für Parameter die verfügbaren Parameter aus den Runbooks für diesen Änderungsauftrag aus. Diese Auswahl spiegelt den Typ der Ressource wider, die aktualisiert wird.

         Wenn das Runbook für diese Änderungsanforderung beispielsweise `AWS-RetartEC2Instance` ist, können Sie `InstanceId` wählen und dann festlegen, welche Instances aktualisiert werden, indem Sie eine der folgenden Optionen auswählen:

         - Specify tags (Tags angeben) - Geben Sie ein Schlüssel-Wert-Paar ein, mit dem alle zu aktualisierenden Ressourcen getaggt werden.

- Choose a resource group (Eine Ressourcengruppe auswählen) - Wählen Sie den Namen der Ressourcengruppe aus, zu der alle zu aktualisierenden Ressourcen gehören.
- Specify parameter values (Parameterwerte angeben) - Identifizieren Sie die zu aktualisierenden Ressourcen im Abschnitt Runbook parameters (Runbook-Parameter).
- Target all instances (Alle Instances anvisieren) – Nehmen Sie die Änderung für alle verwalteten Knoten an den Zielorten vor.

2. Wenn Sie Multiple resources (Mehrere Ressourcen) wählen, erweitern Sie Rate control (Ratenregelung).

   Geben Sie für Concurrency (Gleichzeitigkeit) eine Zahl ein und wählen Sie dann aus der Liste aus, ob dies die Anzahl oder den Prozentsatz der Ziele darstellt, die der Runbook-Workflow gleichzeitig aktualisieren kann.

   Geben Sie für Error threshold (Schwellenwert-Fehler) eine Zahl ein und wählen Sie dann aus der Liste aus, ob dies die Anzahl oder den Prozentsatz der Ziele darstellt, bei denen die Aktualisierung fehlschlagen kann, bevor der Vorgang beendet wird.

15. Wenn Sie Specify parameter values (Parameterwerte angeben) gewählt haben, um mehrere Ressourcen im vorherigen Schritt zu aktualisieren: Geben Sie im Abschnitt Runbook parameters (Runbook-Parameter die Werte für die erforderlichen Eingabeparameter an. Die Parameterwerte, die Sie angeben müssen, basieren auf dem Inhalt der Automation-Runbooks, die der ausgewählten Änderungsvorlage zugeordnet sind.

    Wenn die Änderungsvorlage beispielsweise das `AWS-RetartEC2Instance` Runbook verwendet, müssen Sie eine oder mehrere Instanzen IDs für den InstanceIdParameter eingeben. Alternativ können Sie Show interactive instance picker (Interaktive Instance-Auswahl anzeigen) wählen und nachher die verfügbaren Instance einzeln auswählen.

16. Wählen Sie Weiter aus.

17. Überprüfen Sie auf der Seite Review and submit (Überprüfen und Einreichen) die Ressourcen und Optionen, die Sie für diesen Änderungsantrag angegeben haben.

    Wählen Sie die Schaltfläche Bearbeiten für jeden Abschnitt, an dem Sie Änderungen vornehmen möchten.

    Wenn Sie mit den Details zur Änderungsanforderung zufrieden sind, klicken Sie auf Submit for approval (Zur Genehmigung einreichen).

Wenn in der Änderungsvorlage, die Sie für die Anfrage ausgewählt haben, ein Amazon SNS-Thema angegeben wurde, werden Benachrichtigungen gesendet, wenn die Anfrage abgelehnt oder genehmigt wird. Wenn Sie keine Benachrichtigungen für die Anfrage erhalten, können Sie zu Change Manager zurückkehren, um den Status Ihrer Anfrage zu überprüfen.

Erstellen von Änderungsanforderungen (AWS CLI)

Sie können mithilfe von AWS Command Line Interface (AWS CLI) eine Änderungsanforderung erstellen, indem Sie Optionen und Parameter für die Änderungsanforderung in einer JSON-Datei angeben und die `--cli-input-json` Option verwenden, um sie in Ihren Befehl aufzunehmen.

So erstellen Sie eine Änderungsanforderung (AWS CLI)

1. Installieren und konfigurieren Sie das AWS CLI oder das AWS -Tools für PowerShell, falls Sie das noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.

2. Erstellen Sie eine JSON-Datei auf Ihrem lokalen Computer und geben Sie Ihr beispielsweise einen Namen wie `MyChangeRequest.json`. Fügen Sie der Datei anschließend den folgenden Inhalt ein:

   *`placeholders`*Ersetzen Sie sie durch Werte für Ihre Änderungsanforderung.

   > **Note**
   >
   > Dieser Beispiel-JSON-Code erstellt eine Änderungsanforderung mithilfe der `AWS-HelloWorldChangeTemplate`-Änderungsvorlage und dem `AWS-HelloWorld`-Runbook. Damit Sie dieses Beispiel für Ihre eigenen Änderungsanfragen anpassen können, finden Sie unter StartChangeRequestExecution in der AWS Systems Manager -API-Referenz Informationen zu allen verfügbaren Parametern
   > Weitere Informationen zum Genehmigen von Änderungsanfragen finden Sie unter Über die Genehmigung von Änderungsanfragen.

```
{
    "ChangeRequestName": "MyChangeRequest",
    "DocumentName": "AWS-HelloWorldChangeTemplate",
    "DocumentVersion": "$DEFAULT",
```

```
    "ScheduledTime": "2021-12-30T03:00:00",
    "ScheduledEndTime": "2021-12-30T03:05:00",
    "Tags": [
        {
            "Key": "Purpose",
            "Value": "Testing"
        }
    ],
    "Parameters": {
        "Approver": [
            "JohnDoe"
        ],
        "ApproverType": [
            "IamUser"
        ],
        "ApproverSnsTopicArn": [
            "arn:aws:sns:us-east-2:123456789012:MyNotificationTopic"
        ]
    },
    "Runbooks": [
        {
            "DocumentName": "AWS-HelloWorld",
            "DocumentVersion": "1",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Parameters": {
                "AutomationAssumeRole": [
                    "arn:aws:iam::123456789012:role/MyChangeManagerAssumeRole"
                ]
            }
        }
    ],
    "ChangeDetails": "### Document Name: HelloWorldChangeTemplate\n\n## What does
 this document do?\nThis change template demonstrates the feature set available
 for creating change templates for Change Manager. This template starts a Runbook
 workflow for the Automation document called AWS-HelloWorld.\n\n## Input Parameters
\n* ApproverSnsTopicArn: (Required) Amazon Simple Notification Service ARN for
 approvers.\n* Approver: (Required) The name of the approver to send this request
 to.\n* ApproverType: (Required) The type of reviewer.\n  * Allowed Values:
 IamUser, IamGroup, IamRole, SSOGroup, SSOUser\n\n## Output Parameters\nThis
 document has no outputs \n"
}
```

3. Führen Sie in dem Verzeichnis, in dem Sie die JSON-Datei erstellt haben, den folgenden Befehl aus.

```
aws ssm start-change-request-execution --cli-input-json file://MyChangeRequest.json
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "AutomationExecutionId": "b3c1357a-5756-4839-8617-2d2a4EXAMPLE"
}
```

Überprüfen und Genehmigen oder Ablehnen von Änderungsanforderungen

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestehende Kunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Wenn Sie als Prüfer für eine Änderungsanforderung in Change Manager, einem Tool in AWS Systems Manager, angegeben sind, werden Sie über ein Amazon Simple Notification Service (Amazon SNS)-Thema benachrichtigt, wenn eine neue Änderungsanfrage auf Ihre Überprüfung wartet.

> **Note**
>
> Diese Funktion hängt davon ab, ob ein Amazon SNS in der Änderungsvorlage für das Senden von Überprüfungsbenachrichtigungen angegeben wurde. Weitere Informationen finden Sie unter Konfigurieren von Amazon SNS-Themen für Change Manager-Benachrichtigungen.

Um die Änderungsanfrage zu überprüfen, können Sie dem Link in Ihrer Benachrichtigung folgen oder sich AWS-Managementkonsole direkt bei der anmelden und die Schritte in diesem Verfahren befolgen.

> **Note**
>
> Wenn ein Amazon SNS-Thema für Prüfer in einer Änderungsvorlage zugewiesen ist, werden Benachrichtigungen an die Abonnenten des Themas gesendet, wenn sich die Änderungsanforderung ändert.
> Weitere Informationen zum Genehmigen von Änderungsanfragen finden Sie unter Über die Genehmigung von Änderungsanfragen.

Überprüfen und Genehmigen oder Ablehnen von Änderungsanforderungen (Konsole)

Die folgenden Verfahren beschreiben, wie Sie die Systems-Manager-Konsole verwenden, um Änderungsanforderungen zu überprüfen und zu genehmigen oder abzulehnen.

Überprüfen und Genehmigen oder Ablehnen eines einzigen Änderungsantrags

1. Öffnen Sie den Link in der E-Mail-Benachrichtigung, die Sie erhalten haben, und melden Sie sich bei der an AWS-Managementkonsole, wodurch Sie zur Änderungsanfrage weitergeleitet werden, die Sie überprüfen können.
2. Überprüfen Sie auf der Übersichtsseite den vorgeschlagenen Inhalt der Änderungsanforderung.

   Um die Änderungsanforderung zu genehmigen, wählen Sie Approve (Genehmigen). Geben Sie im Dialogfeld alle Kommentare ein, die Sie für diese Genehmigung hinzufügen möchten, und wählen Sie dann Approve (Genehmigen). Der durch diese Anforderung dargestellte Runbook-Workflow beginnt entweder nach der Planung oder sobald Änderungen nicht durch Einschränkungen blockiert sind.

   –oder–

   Um die Änderungsanforderung abzulehnen, wählen Sie Reject (Ablehnen). Geben Sie im Dialogfeld alle Kommentare ein, die Sie für diese Ablehnung hinzufügen möchten, und wählen Sie dann Reject (Ablehnen).

Überprüfen und Genehmigen oder Ablehnen von Änderungsanträgen auf einmal

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Manager aus.
3. Wählen Sie die Registerkarte Approvals (Genehmigungen) aus.
4. (Optional) Überprüfen Sie die Details der Anfragen, für die Ihre Genehmigung aussteht, indem Sie den Namen jeder Anfrage auswählen und dann zur Registerkarte Approvals (Genehmigungen) zurückkehren.
5. Aktivieren Sie das Kontrollkästchen jeder Änderungsanforderung, die Sie genehmigen möchten.

   –oder–

   Aktivieren Sie das Kontrollkästchen jeder Änderungsanforderung, die Sie ablehnen möchten.
6. Geben Sie im Dialogfeld alle Kommentare ein, die Sie für diese Genehmigung oder Ablehnung hinzufügen möchten.
7. Je nachdem, ob Sie die ausgewählten Änderungsanträge genehmigen oder ablehnen, wählen Sie Approve (Genehmigen) oder Reject (Ablehnen) aus.

Überprüfen und Genehmigen oder Ablehnen einer Änderungsanforderung (Befehlszeile)

Das folgende Verfahren beschreibt, wie Sie die AWS Command Line Interface (AWS CLI) (unter Linux, oderWindows Server) verwendenmacOS, um eine Änderungsanforderung zu überprüfen und zu genehmigen oder abzulehnen.

Überprüfen und Genehmigen oder Ablehnen eines Änderungsantrags

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Erstellen Sie auf Ihrem lokalen Computer eine JSON-Datei, die die Parameter für Ihren AWS CLI Aufruf angibt.

```
{
  "OpsItemFilters":
  [
```

```
    {
      "Key": "OpsItemType",
      "Values": ["/aws/changerequest"],
      "Operator": "Equal"
    }
  ],
  "MaxResults": number
}
```

Sie können die Ergebnisse für einen bestimmten Genehmiger filtern, indem Sie den Amazon Resource Name (ARN) des Genehmigers in der JSON-Datei angeben. Ein Beispiel.

```
{
  "OpsItemFilters":
  [
    {
      "Key": "OpsItemType",
      "Values": ["/aws/changerequest"],
      "Operator": "Equal"
    },
    {
      "Key": "ChangeRequestByApproverArn",
      "Values": ["arn:aws:iam::account-id:user/user-name"],
      "Operator": "Equal"
    }
  ],
  "MaxResults": number
}
```

3. Führen Sie den folgenden Befehl aus, um die maximale Anzahl von Änderungsanforderungen anzuzeigen, die Sie in der JSON-Datei angegeben haben.

   Linux & macOS

   ```
   aws ssm describe-ops-items \
   --cli-input-json file://filename.json
   ```

   Windows

   ```
   aws ssm describe-ops-items ^
   --cli-input-json file://filename.json
   ```

4. Führen Sie den folgenden Befehl aus, um eine Änderungsanforderung zu genehmigen oder abzulehnen.

   Linux & macOS

   ```
   aws ssm send-automation-signal \
       --automation-execution-id ID \
       --signal-type Approve_or_Reject \
       --payload Comment="message"
   ```

   Windows

   ```
   aws ssm send-automation-signal ^
   --automation-execution-id ID ^
       --signal-type Approve_or_Reject ^
       --payload Comment="message"
   ```

   Wenn in der Änderungsvorlage, die Sie für die Anfrage ausgewählt haben, ein Amazon SNS-Thema angegeben wurde, werden Benachrichtigungen gesendet, wenn die Anfrage abgelehnt oder genehmigt wird. Wenn Sie keine Benachrichtigungen für die Anfrage erhalten, können Sie zu Change Manager zurückkehren, um den Status Ihrer Anfrage zu überprüfen. Informationen zu anderen Optionen bei Verwendung dieses Befehls finden Sie unter send-automation-signal im AWS Systems Manager -Abschnitt der AWS CLI -Befehlsreferenz.

Überprüfen von Details, Aufgaben und Zeitplänen für Änderungsanforderungen (Konsole)

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestehende Kunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Im Dashboard von Change Manager, ein Tool in AWS Systems Manager, können Sie Informationen zu einem Änderungsauftrag anzeigen, einschließlich der Aufträge, für die bereits Änderungen bearbeitet wurden. Diese Details enthalten einen Link zur Automatisierungs-Operation, mit der die

Runbooks ausgeführt werden, die die Änderung vornehmen. Eine Automation-Ausführungs-ID wird generiert, wenn die Anforderung erstellt wird. Der Prozess wird jedoch erst ausgeführt, wenn alle Genehmigungen erteilt wurden und keine Einschränkungen vorhanden sind, um die Änderung zu blockieren.

Überprüfung von Details, Aufgaben und Zeitplänen für Änderungsanträge

1. Wählen Sie im Navigationsbereich Change Manager aus.
2. Wählen Sie die Registerkarte Requests (Anforderungen).
3. Suchen Sie im Abschnitt Change requests (Änderungsanforderungen) nach der Änderungsanforderung, die Sie überprüfen möchten.

   Sie können die Option Create date range (Datumsbereich erstellen) verwenden, um die Ergebnisse auf einen bestimmten Zeitraum zu beschränken.

   Sie können Anforderungen nach folgenden Eigenschaften filtern:

   - `Status`
   - `Request ID`
   - `Approver`
   - `Requester`

   Führen Sie beispielsweise die folgenden Schritte aus, um Details zu allen Änderungsanforderungen anzuzeigen, die in den letzten 24 Stunden erfolgreich abgeschlossen wurden:

   1. Wählen Sie für Create date range (Datumsbereich erstellen) 1d.
   2. Wählen Sie im Suchfeld Status, aus CompletedWithSuccess.
   3. Wählen Sie in den Ergebnissen den Namen der erfolgreich abgeschlossenen Änderungsanforderung aus, für die die Ergebnisse überprüft werden sollen.
4. Zeigen Sie Informationen zur Änderungsanforderung auf den folgenden Registerkarten an:

   - Request details (Details anfordern) - Zeigt grundlegende Details zur Änderungsanforderung an, einschließlich des Anforderers, der Änderungsvorlage und der Automation-Runbooks, die für die Änderung ausgewählt wurden. Sie können auch einem Link zu den Details des Automatisierungsvorgangs folgen und Informationen zu allen in der Anfrage angegebenen Runbook-Parametern, den dem Änderungsantrag zugewiesenen CloudWatch Amazon-

Alarmen sowie zu den Genehmigungen und Kommentaren, die für die Anfrage bereitgestellt wurden, einsehen.

- Task (Aufgabe) - Zeigt Informationen über die Aufgabe in der Änderung an, einschließlich des Aufgabenstatus für abgeschlossene Änderungsanforderungen, der Zielressourcen, der Schritte in den zugeordneten Automation-Runbooks sowie Details zum Parallelitäts- und Fehlerschwellenwert.
- Timeline (Zeitplan)- Zeigt eine Zusammenfassung aller Ereignisse an, die mit dem Änderungsauftrag verknüpft sind, nach Datum und Uhrzeit. Die Zusammenfassung gibt an, wann die Änderungsanforderung erstellt wurde, welche Aktionen von den zugewiesenen Genehmigern durchgeführt wurden, wann die genehmigten Änderungsanforderungen ausgeführt werden sollen, wie der Workflow des Runbooks aussieht und welche Statusänderungen für den gesamten Änderungsprozess und die einzelnen Schritte im Runbook vorgenommen wurden.
- Associated events (Zugeordnete Ereignisse) – Zeigen Sie überprüfbare Details zu Änderungsanfragen an, die in AWS CloudTrail Lake aufgezeichnet wurden. Zu den Details gehören die ausgeführten API-Aktionen, die für diese Aktionen enthaltenen Anforderungsparameter, das Benutzerkonto, das die Aktion ausgeführt hat, die während des Prozesses aktualisierten Ressourcen und mehr.

  Wenn Sie die CloudTrail Lake-Ereignisverfolgung aktivieren, erstellt CloudTrail Lake einen Ereignisdatenspeicher für Ereignisse im Zusammenhang mit Ihren Änderungsanforderungen. Die Ereignisdetails sind für das Konto oder die Organisation verfügbar, für die die Änderungsanfrage ausgeführt wurde. Sie können die CloudTrail Lake-Ereignisverfolgung von jeder Änderungsanfrage in Ihrem Konto oder Ihrer Organisation aus aktivieren. Informationen zur Aktivierung der CloudTrail Lake-Integration und zum Erstellen eines Ereignisdatenspeichers finden Sie unterÜberwachung der Ereignisse Ihrer Änderungsanfragen.

  > **Note**
  >
  > Die Nutzung von CloudTrail Lake ist kostenpflichtig. Weitere Details finden Sie unter AWS CloudTrail -Preise.

Aggregierte Anzahl von Änderungsaufträgen anzeigen (Befehlszeile)

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Mithilfe der GetOpsSummaryAPI-Operation können Sie die aggregierte Anzahl von Änderungsanforderungen in Change Manager einem Tool in AWS Systems Manager anzeigen. Dieser API-Vorgang kann Zählungen für ein einzelnes Konto AWS-Konto in einem einzigen AWS-Region oder für mehrere Konten und mehrere Regionen zurückgeben.

> **Note**
>
> Wenn Sie die aggregierte Anzahl von Änderungsanträgen für mehrere AWS-Konten und mehrere anzeigen möchten AWS-Regionen, müssen Sie eine Ressourcendatensynchronisierung einrichten und konfigurieren. Weitere Informationen finden Sie unter Erstellen einer Resource Data Sync für Inventory.

Das folgende Verfahren beschreibt, wie Sie AWS Command Line Interface (AWS CLI) (unter Linux, oderWindows Server) verwendenmacOS, um die aggregierte Anzahl von Änderungsanforderungen anzuzeigen.

Anzeigen der aggregierten Anzahl von Änderungsanforderungen

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie einen der folgenden Befehle aus.

   Single account and Region (Einzelnes Konto und Region)

Dieser Befehl gibt die Anzahl aller Änderungsanforderungen für die AWS-Konto und AWS-Region für die Ihre AWS CLI Sitzung konfiguriert ist, zurück.

Linux & macOS

```
aws ssm get-ops-summary \
--filters Key=AWS:OpsItem.OpsItemType,Values="/aws/changerequests",Type=Equal \
--aggregators AggregatorType=count,AttributeName=Status,TypeName=AWS:OpsItem
```

Windows

```
aws ssm get-ops-summary ^
--filters Key=AWS:OpsItem.OpsItemType,Values="/aws/changerequests",Type=Equal ^
--aggregators AggregatorType=count,AttributeName=Status,TypeName=AWS:OpsItem
```

Das Aufruf gibt unter anderem folgende Informationen zurück.

```
{
    "Entities": [
        {
            "Data": {
                "AWS:OpsItem": {
                    "Content": [
                        {
                            "Count": "38",
                            "Status": "Open"
                        }
                    ]
                }
            }
        }
    ]
}
```

Mehrere Konten, and/or Regionen

Dieser Befehl gibt die Anzahl aller Änderungsanforderungen für die AWS-Konten und, die in der Ressourcendatensynchronisierung AWS-Regionen angegeben sind, zurück.

Linux & macOS

```
aws ssm get-ops-summary \
    --sync-name resource_data_sync_name \
    --filters Key=AWS:OpsItem.OpsItemType,Values="/aws/
changerequests",Type=Equal \
    --aggregators AggregatorType=count,AttributeName=Status,TypeName=AWS:OpsItem
```

Windows

```
aws ssm get-ops-summary ^
    --sync-name resource_data_sync_name ^
    --filters Key=AWS:OpsItem.OpsItemType,Values="/aws/
changerequests",Type=Equal ^
    --aggregators AggregatorType=count,AttributeName=Status,TypeName=AWS:OpsItem
```

Das Aufruf gibt unter anderem folgende Informationen zurück.

```
{
    "Entities": [
        {
            "Data": {
                "AWS:OpsItem": {
                    "Content": [
                        {
                            "Count": "43",
                            "Status": "Open"
                        },
                        {
                            "Count": "2",
                            "Status": "Resolved"
                        }
                    ]
                }
            }
        }
    ]
}
```

Multiple accounts and a specific Region (Mehrere Konten und eine spezifische Region

Dieser Befehl gibt eine Anzahl aller Änderungsanforderungen für die AWS-Konten aus, die in der Ressourcendatensynchronisation angegeben sind. Er gibt jedoch nur Daten aus der Region aus, die im Befehl angegeben ist.

Linux & macOS

```
aws ssm get-ops-summary \
    --sync-name resource_data_sync_name \
    --filters Key=AWS:OpsItem.SourceRegion,Values='Region',Type=Equal
 Key=AWS:OpsItem.OpsItemType,Values="/aws/changerequests",Type=Equal \
    --aggregators AggregatorType=count,AttributeName=Status,TypeName=AWS:OpsItem
```

Windows

```
aws ssm get-ops-summary ^
    --sync-name resource_data_sync_name ^
    --filters Key=AWS:OpsItem.SourceRegion,Values='Region',Type=Equal
 Key=AWS:OpsItem.OpsItemType,Values="/aws/changerequests",Type=Equal ^
    --aggregators AggregatorType=count,AttributeName=Status,TypeName=AWS:OpsItem
```

Multiple accounts and Regions with output grouped by Region (Mehrere Konten und Regionen mit Ausgabe gruppiert nach Region)

Dieser Befehl gibt die Anzahl aller Änderungsanforderungen für AWS-Konten und zurück, die in der Ressourcendatensynchronisierung AWS-Regionen angegeben wurden. Die Ausgabe zeigt die Zählinformationen pro Region an.

Linux & macOS

```
aws ssm get-ops-summary \
    --sync-name resource_data_sync_name \
    --filters Key=AWS:OpsItem.OpsItemType,Values="/aws/
changerequests",Type=Equal \
    --aggregators
 '[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"Status","Aggregat
[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"SourceRegion"}]}]'
```

Windows

```
aws ssm get-ops-summary ^
    --sync-name resource_data_sync_name ^
    --filters Key=AWS:OpsItem.OpsItemType,Values="/aws/
changerequests",Type=Equal ^
    --aggregators
 '[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"Status","Aggregat
[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"SourceRegion"}]}]'
```

Das Aufruf gibt unter anderem folgende Informationen zurück.

```
{
        "Entities": [
            {
                "Data": {
                    "AWS:OpsItem": {
                        "Content": [
                            {
                                "Count": "38",
                                "SourceRegion": "us-east-1",
                                "Status": "Open"
                            },
                            {
                                "Count": "4",
                                "SourceRegion": "us-east-2",
                                "Status": "Open"
                            },
                            {
                                "Count": "1",
                                "SourceRegion": "us-west-1",
                                "Status": "Open"
                            },
                            {
                                "Count": "2",
                                "SourceRegion": "us-east-2",
                                "Status": "Resolved"
                            }
                        ]
                    }
                }
```

```
            }
        ]
    }
```

Multiple accounts and Regions with output grouped by accounts and Regions (Mehrere Konten und Regionen mit Ausgabe gruppiert nach Konten und Regionen)

Dieser Befehl gibt die Anzahl aller Änderungsanforderungen für AWS-Konten und zurück, die in der Ressourcendatensynchronisierung AWS-Regionen angegeben wurden. Die Ausgabe gruppiert die Zählinformationen nach Konten und Regionen.

Linux & macOS

```
aws ssm get-ops-summary \
    --sync-name resource_data_sync_name \
    --filters Key=AWS:OpsItem.OpsItemType,Values="/aws/
changerequests",Type=Equal \
    --aggregators
 '[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"Status","Aggregat
[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"SourceAccountId","A
[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"SourceRegion"}]}]}]
```

Windows

```
aws ssm get-ops-summary ^
    --sync-name resource_data_sync_name ^
    --filters Key=AWS:OpsItem.OpsItemType,Values="/aws/
changerequests",Type=Equal ^
    --aggregators
 '[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"Status","Aggregat
[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"SourceAccountId","A
[{"AggregatorType":"count","TypeName":"AWS:OpsItem","AttributeName":"SourceRegion"}]}]}]
```

Das Aufruf gibt unter anderem folgende Informationen zurück.

```
{
    "Entities": [
        {
            "Data": {
                "AWS:OpsItem": {
```

```
                    "Content": [
                        {
                            "Count": "38",
                            "SourceAccountId": "123456789012",
                            "SourceRegion": "us-east-1",
                            "Status": "Open"
                        },
                        {
                            "Count": "4",
                            "SourceAccountId": "111122223333",
                            "SourceRegion": "us-east-2",
                            "Status": "Open"
                        },
                        {
                            "Count": "1",
                            "SourceAccountId": "111122223333",
                            "SourceRegion": "us-west-1",
                            "Status": "Open"
                        },
                        {
                            "Count": "2",
                            "SourceAccountId": "444455556666",
                            "SourceRegion": "us-east-2",
                            "Status": "Resolved"
                        },
                        {
                            "Count": "1",
                            "SourceAccountId": "222222222222",
                            "SourceRegion": "us-east-1",
                            "Status": "Open"
                        }
                    ]
                }
            }
        }
    ]
}
```

Prüfen und Protokollieren von Change Manager-Aktivitäten

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Sie können Aktivitäten inChange Manager, einem Tool in, überprüfen AWS Systems Manager, indem Sie Amazon CloudWatch und AWS CloudTrail Alarme verwenden.

Weitere Informationen zu den Prüfungs- und Protokollierungsoptionen für Systems Manager finden Sie unter Einloggen und Überwachen AWS Systems Manager.

Change ManagerAktivitäten mithilfe von CloudWatch Alarmen prüfen

Sie können einen CloudWatch Alarm konfigurieren und einer Änderungsvorlage zuweisen. Wenn im Alarm definierte Bedingungen erfüllt sind, werden die für den Alarm angegebenen Aktionen ausgeführt. In der Alarmkonfiguration können Sie ein Amazon Simple Notification Service (Amazon SNS)-Thema angeben, um zu benachrichtigen, wenn eine Alarmbedingung erfüllt ist.

Weitere Informationen zum Erstellen einer Change Manager-Vorlage finden Sie unter Arbeiten mit Änderungsvorlagen.

Informationen zum Erstellen von CloudWatch Alarmen finden Sie unter Verwenden von CloudWatch Alarmen im CloudWatch Amazon-Benutzerhandbuch.

Überprüfen Sie Change Manager die Aktivität mithilfe von CloudTrail

CloudTrail erfasst API-Aufrufe, die in der Systems Manager-Konsole, dem AWS Command Line Interface (AWS CLI) und dem Systems Manager SDK getätigt wurden. Sie können die Informationen in der CloudTrail Konsole oder in einem Amazon Simple Storage Service (Amazon S3) -Bucket anzeigen, wo sie gespeichert sind. Ein Bucket wird für alle CloudTrail Protokolle Ihres Kontos verwendet.

Protokolle von Change Manager-Aktionen zeigen die Erstellung von Änderungsvorlagendokumenten, Genehmigungen und Ablehnungen von Änderungsvorlagen und Änderungsanforderungen, von

Automation-Runbooks generierte Aktivitäten und vieles mehr. Weitere Informationen zum Anzeigen und Verwenden von CloudTrail Protokollen der Systems Manager Manager-Aktivitäten finden Sie unterProtokollierung von AWS Systems Manager-API-Aufrufen mit AWS CloudTrail.

Fehlerbehebung für Change Manager

> **Change ManagerVerfügbarkeit ändern**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestehende Kunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Im Folgenden finden Sie Informationen zur Behandlung von Problemen mit Change Manager, einem Tool in AWS Systems Manager.

Themen

- Fehler „Gruppe {GUID} nicht gefunden“ bei der Genehmigung von Änderungsanforderungen bei der Verwendung von Active Directory (Gruppen)

Fehler „Gruppe *{GUID}* nicht gefunden“ bei der Genehmigung von Änderungsanforderungen bei der Verwendung von Active Directory (Gruppen)

Problem: Wenn AWS IAM Identity Center (IAM Identity Center) für die Benutzeridentitätsverwaltung verwendet wird, Change Manager erhält ein Mitglied einer Active Directory-Gruppe, dem Genehmigungsberechtigungen erteilt wurden, die Fehlermeldung „Nicht autorisiert“ oder „Gruppe nicht gefunden“.

- Lösung: Wenn Sie Active Directory-Gruppen in IAM Identity Center für den Zugriff auf auswählen AWS-Managementkonsole, plant das System eine regelmäßige Synchronisierung, bei der Informationen aus diesen Active Directory-Gruppen in IAM Identity Center kopiert werden. Dieser Vorgang muss abgeschlossen werden, bevor Benutzer, die über die Active Directory-Gruppenmitgliedschaft autorisiert sind, eine Anforderung erfolgreich genehmigen können. Weitere Informationen finden Sie unter Mit Ihrem Microsoft-AD-Verzeichnis verbinden im AWS IAM Identity Center -Benutzerhandbuch.

AWS Systems Manager-Documents

Ein AWS Systems Manager Dokument (SSM-Dokument) definiert die Aktionen, die Systems Manager auf Ihren verwalteten Instanzen ausführt. Systems Manager umfasst mehr als 100 vorkonfigurierter Dokumente, die Sie verwenden können, indem Sie zur Laufzeit Parameter angeben. Vorkonfigurierte Dokumente finden Sie in der Systems-Manager-Dokumentenkonsole, indem Sie die Registerkarte Owned by Amazon (Eigentum von Amazon) auswählen. Alternativ können Sie beim Aufrufen des API-Vorgangs `Owner` für den Filter `ListDocuments` Amazon angeben. Dokumente verwenden JavaScript Object Notation (JSON) oder YAML und enthalten Schritte und Parameter, die Sie angeben.

Aus Sicherheitsgründen unterstützen SSM-Dokumente seit dem 14. Juli 2025 die Interpolation von Umgebungsvariablen bei der Verarbeitung von Parametern. Dieses Feature, das in Schemaversion 2.2 und in SSM Agent-Version 3.3.2746.0 oder höher verfügbar ist, trägt dazu bei, Angriffe mit Befehlseinschleusung zu verhindern.

Um mit SSM-Dokumenten zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich die Option Dokumente aus.

> **Important**
>
> In Systems Manager ist ein zu Amazon gehörendes SSM-Dokument ein Dokument, das von Amazon Web Services selbst erstellt und verwaltet wird. Dokumente, die Amazon gehören, enthalten ein Präfix wie `AWS-*` im Dokumentnamen. Als Eigentümer des Dokuments gilt Amazon und nicht als ein bestimmtes Benutzerkonto innerhalb AWS. Diese Dokumente sind öffentlich zugänglich und können von allen verwendet werden.

Wie kann meine Organisation vom Documents-Tool profitieren?

Documents, ein Tool in AWS Systems Manager, bietet folgende Vorteile:

- Kategorien von Dokumenten

  Zur einfacheren Suche nach den benötigten Dokumenten wählen Sie je nach Typ des gesuchten Dokuments eine Kategorie aus. Um die Suche zu erweitern, können Sie mehrere Kategorien desselben Dokumenttyps auswählen. Die Auswahl von Kategorien verschiedener Dokumenttypen wird nicht unterstützt. Kategorien werden nur für Dokumente im Besitz von Amazon unterstützt.

- Dokumentversionen

  Sie können unterschiedliche Versionen von Dokumenten erstellen und speichern. Anschließend können Sie eine Standardversion für jedes Dokument angeben. Die Standardversion eines Dokuments kann auf eine neuere Version aktualisiert und wieder auf eine ältere Version zurückgesetzt werden. Wenn Sie den Inhalt eines Dokuments ändern, inkrementiert Systems Manager automatisch die Versionsnummer des Dokuments. Sie können jede Version eines Dokuments abrufen oder verwenden, indem Sie die Dokumentversion in der Konsole, AWS Command Line Interface (AWS CLI) -Befehlen oder API-Aufrufen angeben.

- Anpassen von Dokumenten an die eigenen Bedürfnisse

  Wenn Sie die Schritte und Aktionen in einem Dokument anpassen möchten, können Sie Ihre eigenen Dokumente erstellen. Das System speichert das Dokument zusammen mit Ihrem Namen, AWS-Konto in dem AWS-Region Sie es erstellt haben. Weitere Informationen zum Erstellen eines SSM-Dokuments finden Sie unter Erstellen von SSM-Dokumentinhalten.

- Markieren von Dokumenten

  Sie können Ihre Dokumente markieren, um sie später anhand der zugewiesenen Tags schnell zu identifizieren. Beispielsweise können Sie Dokumente für bestimmte Umgebungen, Abteilungen, Benutzer, Gruppen oder Zeiträume markieren. Sie können den Zugriff auf Dokumente auch einschränken, indem Sie eine AWS Identity and Access Management (IAM-) Richtlinie erstellen, die festlegt, auf welche Tags ein Benutzer oder eine Gruppe zugreifen kann.

- Freigeben von Dokumenten

  Sie können Ihre Dokumente öffentlich zugänglich machen oder für bestimmte AWS-Konten in derselben AWS-Region freigeben. Das Teilen von Dokumenten zwischen Konten kann nützlich sein, wenn Sie beispielsweise möchten, dass alle Amazon Elastic Compute Cloud (Amazon EC2) -Instances, die Sie Kunden oder Mitarbeitern bereitstellen, dieselbe Konfiguration haben. Möglicherweise möchten Sie nicht nur Anwendungen oder Patches auf den Instances auf dem neuesten Stand halten, sondern auch bestimmte Aktivitäten von Kunden-Instances beschränken. Oder Sie möchten sicherstellen, dass Instances, die von Mitarbeiterkonten organisationsweit genutzt werden, auf bestimmte interne Ressourcen zugreifen können. Weitere Informationen finden Sie unter Freigeben von SSM-Dokumenten.

Wer sollte Documents verwenden?

- Jeder AWS Kunde, der Systems Manager Manager-Tools verwenden möchte, um seine betriebliche Effizienz in großem Umfang zu verbessern, Fehler im Zusammenhang mit manuellen Eingriffen zu reduzieren und die Zeit bis zur Lösung häufiger Probleme zu verkürzen.
- Infrastrukturexperten, die Bereitstellungs- und Konfigurationsaufgaben automatisieren möchten.
- Administratoren, die häufig auftretende Probleme zuverlässig lösen, die Effizienz bei der Fehlerbehebung verbessern und die Anzahl sich wiederholender Vorgänge reduzieren möchten.
- Benutzer, die eine Aufgabe automatisieren möchten, die sie normalerweise manuell ausführen.

Welche Typen von SSM-Dokumenten gibt es?

Die folgende Tabelle beschreibt die verschiedenen Arten von SSM-Dokumenten und ihre jeweilige Nutzung.

Type (Typ)	Verwendet mit	Details
ApplicationConfiguration ApplicationConfigurationSchema	AWS AppConfig	AWS AppConfig, ein Tool in AWS Systems Manager, ermöglicht es Ihnen, Anwendungskonfigurationen zu erstellen, zu verwalten und schnell bereitzustellen. Sie können Konfigurationsdaten in einem SSM-Dokument speichern, indem Sie ein Dokument erstellen, das den Dokumenttyp `ApplicationConfiguration` verwendet. Weitere Informationen finden Sie unter Freeform configurations (Freiform-Konfigurationen) im AWS AppConfig Benutzerhandbuch.

Type (Typ)	Verwendet mit	Details
		Wenn Sie eine Konfiguration in einem SSM-Dokument erstellen, müssen Sie ein entsprechendes JSON-Schema angeben. Das Schema verwendet den `ApplicationConfigurationSchema` -Dokumenttyp und definiert wie ein Regelsatz die zulässigen Eigenschaften für jede Anwendungskonfigurationseinstellung. Weitere Informationen finden Sie unter About validators (Informationen zu Validatoren) im AWS AppConfig -Benutzerhandbuch.

Type (Typ)	Verwendet mit	Details
Automation-Runbook	Automation State Manager Maintenance Windows	Verwenden Sie Automation-Runbooks, wenn Sie allgemeine Wartungs- und Bereitstellungsaufgaben durchführen, wie z. B. das Erstellen oder Aktualisieren eines Amazon Machine Image (AMI). State Manager verwendet Automation-Runbooks, um eine Konfiguration anzuwenden. Diese Aktionen können während des Lebenszyklus einer Instance jederzeit für ein oder mehrere Ziele ausgeführt werden. Maintenance Windows verwendet Automation-Runbooks, um basierend auf dem angegebenen Zeitplan allgemeine Wartungs- und Bereitstellungsaufgaben durchzuführen. Alle Automation-Runbooks, die für Linux-basierte Betriebssysteme unterstützt werden, werden auch auf EC2 Instances für unterstützt. macOS

Type (Typ)	Verwendet mit	Details
Kalenderdokument ändern	Change Calendar	Change Calendar, ein Tool in AWS Systems Manager, verwendet den `ChangeCalendar` Dokumenttyp. Ein Change Calendar-Dokument speichert einen Kalendereintrag und zugehörige Ereignisse, die es ermöglichen oder verhindern können, dass Automation-Aktionen Ihre Umgebung verändern. In Change Calendar speichert ein Dokument iCalendar 2.0-Daten im Klartextformat. Change Calendarwird auf EC2 Instanzen für nicht unterstützt ztmacOS.

Type (Typ)	Verwendet mit	Details
AWS CloudFormation Vorlage	AWS CloudFormation	AWS CloudFormation Vorlagen beschreiben die Ressourcen, die Sie in Ihren CloudFormation Stacks bereitstellen möchten. Durch das Speichern von CloudFormation Vorlagen als Systems Manager-D okumente können Sie von den Dokumentfunktionen von Systems Manager profitieren. Dazu gehören das Erstellen und Vergleichen mehrerer Versionen Ihrer Vorlage und das Freigeben Ihrer Vorlage für andere Konten in derselben AWS-Region. Sie können CloudFormation Vorlagen und Stapel mithilfe Application Manager eines Tools in Systems Manager erstellen und bearbeite n. Weitere Informationen finden Sie unter Arbeiten mit CloudFormation-Vorlagen und Stacks in Application Manager.

Type (Typ)	Verwendet mit	Details
Befehlsdokument	Run Command State Manager Maintenance Windows	Run Command, ein Tool in AWS Systems Manager, verwendet Befehlsdokumente, um Befehle auszuführen. State Manager, ein Tool in AWS Systems Manager, verwendet Befehlsdokumente, um eine Konfiguration anzuwenden. Diese Aktionen können zu einem beliebige n Zeitpunkt im Lebenszyklus einer Instance auf einem oder mehreren Zielen ausgeführ t werden. Maintenance Windows, ein Tool in AWS Systems Manager, verwendet Befehlsdokumente, um eine Konfiguration auf der Grundlage des angegebenen Zeitplans anzuwenden. Die meisten Befehlsdo kumente werden unter allen Linux- und Windows Server-Betriebssystemen von Systems Manager unterstüt zt. Die folgenden Befehlsdo kumente werden auf EC2 Instanzen für unterstüt ztmacOS: • `AWS-ConfigureAWSPa ckage` • `AWS-RunPatchBaseli ne`

Type (Typ)	Verwendet mit	Details
		• `AWS-RunPatchBaselineAssociation` • `AWS-RunShellScript`
AWS Config Vorlage für das Konformitätspaket	AWS Config	AWS Config Vorlagen für Konformitätspakete sind Dokumente im YAML-Format, die zur Erstellung von Konformitätspaketen verwendet werden. Sie enthalten die Liste der AWS Config verwalteten oder benutzerdefinierten Regeln und Abhilfemaßnahmen. Weitere Informationen finden Sie unter Konformitätspakete.
Paketdokument	Distributor	In Distributor, einem Tool in AWS Systems Manager, wird ein Paket durch ein SSM-Dokument dargestellt. Ein Paketdokument enthält angefügte ZIP-Archivdateien mit Software oder Ressourcen zur Installation auf verwalteten Instances. Durch die Erstellung eines Pakets in Distributor wird das Paketdokument erstellt. Distributor wird unter Oracle Linux und macOS verwalteten Instances nicht unterstützt.

Type (Typ)	Verwendet mit	Details
Richtliniendokument	State Manager	Inventory, ein Tool in AWS Systems Manager, verwendet das `AWS-GatherSoftwareInventory` Richtlinendokument mit einer State Manager Verknüpfung, um Inventardaten von verwalteten Instanzen zu sammeln. Beim Erstellen eigener SSM-Dokumente sind Automation-Runbooks und Command-Dokumente die bevorzugte Methode zum Durchsetzen einer Richtlinie auf einer verwalteten Instance. Systems Manager Inventory und `AWS-GatherSoftwareInventory` -Richtliniendokument werden auf allen Betriebssystemen unterstützt, die von Systems Manager unterstützt werden.

Type (Typ)	Verwendet mit	Details
Vorlage für die Analyse nach einem Vorfall	Incident Manager-Analyse nach einem Vorfall	Incident Manager verwendet die Vorlage für die Analyse nach einem Vorfall, um eine Analyse zu erstellen, die auf bewährten Methoden für das AWS Betriebsmanagement basiert. Erstellen Sie mithilfe der Vorlage eine Analyse, mit der Ihr Team Verbesserungen für die Reaktion auf Vorfälle ermitteln kann.

Type (Typ)	Verwendet mit	Details
Sitzungsdokument	Session Manager	Session Manager, ein Tool in AWS Systems Manager, bestimmt anhand von Sitzungsdokumenten, welcher Sitzungstyp gestartet werden soll, z. B. eine Portweiterleitungssitzung, eine Sitzung zur Ausführung eines interaktiven Befehls oder eine Sitzung zur Erstellung eines SSH-Tunnels. Befehlsdokumente werden unter allen Linux- und Windows Server-Betriebssystemen von Systems Manager unterstützt. Die folgenden Befehlsdokumente werden auf EC2 Instanzen für macOS unterstützt: • `AWS-PasswordReset` • `AWS-StartInteractiveCommand` • `AWS-StartPortForwardingSession` • `AWS-StartPortForwardingSessionToSocket` • `AWS-StartSSHSession`

SSM-Dokumentkontingente

Informationen zu SSM-Dokumentkontingenten finden Sie unter Systems Manager Service Quotas in der Allgemeine Amazon Web Services-Referenz.

Themen

- Dokument-Komponenten
- Erstellen von SSM-Dokumentinhalten
- Arbeiten mit Dokumenten
- Beheben von Fehlern mit der Parameterverarbeitung

Dokument-Komponenten

Dieser Bereich enthält Informationen zu den Komponenten, aus denen sich SSM-Dokumente zusammensetzen.

Inhalt

- Schemata, Features und Beispiele
- Datenelemente und Parameter
- Referenz für Befehlsdokument-Plugins

Schemata, Features und Beispiele

AWS Systems Manager (SSM) -Dokumente verwenden die folgenden Schemaversionen.

- Dokumente des Typs `Command` können die Schema-Versionen 1.2, 2.0 und 2.2 verwenden. Wenn Sie Schema 1.2-Dokumente verwenden, empfehlen wir, dass Sie Dokumente erstellen, die Schema-Version 2.2 verwenden.
- Dokumente des Typs `Policy` müssen Schema-Version 2.0 oder höher verwenden.
- Dokumente des Typs `Automation` müssen Schema-Version 0.3 verwenden.
- Dokumente des Typs `Session` müssen Schema-Version 1.0 verwenden.
- Sie können Dokumente im JSON- oder YAML-Format erstellen.

Weitere Informationen zu `Session`-Dokumentschemas finden Sie unter Schema des Sitzungsdokuments.

Durch die Verwendung der neuesten Schema-Version `Command`- und `Policy`-Dokumente können Sie die folgenden Features nutzen.

Features für Schema-Version 2.2-Dokumente

Funktion	Details
Dokumentbearbeitung	Dokumente können jetzt aktualisiert werden. Bei Version 1.2 mussten aktualisierte Dokument unter einem anderen Namen gespeichert werden.
Automatisches Versioning	Bei jeder Änderung an einem Dokument wird eine neue Version erstellt. Dies ist kein Schema-Version, sondern eine Version des Dokuments.
Standardversion	Wenn Sie mehrere Versionen eines Dokuments haben, können Sie festlegen, welche Version das Standarddokument ist.
Sequenzierung	Plugins oder Schritte in einem Dokument werden in der Reihenfolge ausgeführt, die Sie angegeben haben.
Unterstützung für plattformübergreifende Anweisungen	Die Unterstützung für plattformübergreifende Anweisungen ermöglicht die Angabe unterschi edlicher Betriebssysteme für verschiedene Plugins innerhalb desselben SSM-Dokum ents. Plattformübergreifende Anweisungen verwenden in einem Schritt den Parameter `precondition` .
Parameterinterpolation	Interpolation bedeutet, einen Variablenwert in einer Zeichenfolge einzufügen oder zu ersetzen. Sie füllen sozusagen eine Lücke mit Werten, bevor die Zeichenfolge verwendet wird. Im Zusammenhang mit SSM-Dokum enten ermöglicht die Parameterinterpolation,

Funktion	Details
	Zeichenfolgeparameter vor der Befehlsausführung in Umgebungsvariablen zu interpolieren, was mehr Schutz vor Befehlseinschleusungen bietet. Wenn diese Option auf `ENV_VAR` gesetzt ist, erstellt der Agent eine Umgebungsvariable mit dem Namen `SSM_`*`parameter-name`*, die den Wert des Parameters enthält.

> **Note**
>
> Sie müssen Ihre Instanzen mit der neuesten Version AWS Systems Manager SSM Agent auf dem neuesten Stand halten, um die neuen Systems Manager Manager-Funktionen und SSM-Dokumentfunktionen nutzen zu können. Weitere Informationen finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command.

In der folgenden Tabelle finden Sie die Unterschiede zwischen de Schema-Hauptversionen.

Version 1.2	Version 2.2 (neueste Version)	Details
runtimeConfig	mainSteps	In Version 2.2 ersetzt der Abschnitt `mainSteps` `runtimeConfig` . Im Abschnitt `mainSteps` erlaubt Systems Manager das Ausführen von nacheinander folgenden Schritten.
Eigenschaften	inputs	In Version 2.2 ersetzt der Abschnitt `inputs` den Abschnitt `properties` . Der Abschnitt `inputs` nimmt Parameter für Schritte entgegen.

Version 1.2	Version 2.2 (neueste Version)	Details
commands	runCommand	In Version 2.2 ersetzt im Abschnitt `inputs` der Parameter `runCommand` den Parameter `commands`.
id	action	In Version 2.2 ersetzt `Action ID`. Dies ist lediglich eine Umbenennung.
n.v.	name	In Version 2.2 ist `name` ein benutzerdefinierter Name für einen Schritt.

Verwenden des Parameters „precondition“

Bei Schema-Version 2.2 oder neuer können Sie mithilfe des Parameters `precondition` das Zielbetriebssystem für jedes Plugin angeben oder um Eingabeparameter zu validieren, die Sie in Ihrem SSM-Dokument definiert haben. Der `precondition`-Parameter unterstützt die Referenzierung der Eingabeparameter Ihres SSM-Dokuments und `platformType` unter Verwendung von Werten von `Linux`, `MacOS` und `Windows`. Nur der `StringEquals`-Operator wird unterstützt.

Wenn bei Dokumenten in Schema-Version 2.2 oder höher `precondition` nicht angegeben ist, werden Plugins entweder ausgeführt oder übersprungen, je nachdem, ob das Plugin mit dem jeweiligen Betriebssystem kompatibel ist. Plugin-Kompatibilität mit dem Betriebssystem wird vor der `precondition` ausgewertet. Bei Dokumenten, die Schema-Version 2.0 oder eine frühere Version verwenden, wird bei nicht kompatiblen Plugins ein Fehler ausgelöst.

Wenn beispielsweise in einem Schema-Version 2.2-Dokument `precondition` nicht angegeben ist und das `aws:runShellScript`-Plugin zur Ausführung aufgelistet ist, wird der Schritt auf Linux-Instances ausgeführt, aber auf Windows Server-Instances übersprungen, da `aws:runShellScript` nicht kompatibel mit Windows Server-Instances ist. Bei Schema-Version 2.0 Dokumenten schlägt jedoch die Ausführung fehl, wenn Sie das `aws:runShellScript`-Plugin angeben und dann das Dokument auf einer Windows Server-Instance ausführen. Weiter hinten in diesem Abschnitt finden Sie ein Beispiel der Vorbedingungsparameter in SSM-Dokumenten.

Schema der Version 2.2

Top-Level-Elemente

Das folgende Beispiel zeigt die Elemente der obersten Ebene eines SSM-Dokuments bei Verwendung von Schema-Version 2.2.

YAML

```
---
schemaVersion: "2.2"
description: A description of the document.
parameters:
  parameter 1:
    property 1: "value"
    property 2: "value"
  parameter 2:
    property 1: "value"
    property 2: "value"
mainSteps:
  - action: Plugin name
    name: A name for the step.
    inputs:
      input 1: "value"
      input 2: "value"
      input 3: "{{ parameter 1 }}"
```

JSON

```
{
   "schemaVersion": "2.2",
   "description": "A description of the document.",
   "parameters": {
       "parameter 1": {
          "property 1": "value",
          "property 2": "value"
       },
       "parameter 2":{
          "property 1": "value",
          "property 2": "value"
       }
    },
   "mainSteps": [
```

```
      {
         "action": "Plugin name",
         "name": "A name for the step.",
         "inputs": {
            "input 1": "value",
            "input 2": "value",
            "input 3": "{{ parameter 1 }}"
         }
      }
   ]
}
```

Schema-Version 2.2 -Beispiel

Im folgenden Beispiel wird das `aws:runPowerShellScript` Plugin verwendet, um einen PowerShell Befehl auf den Zielinstanzen auszuführen.

YAML

```
---
schemaVersion: "2.2"
description: "Example document"
parameters:
  Message:
    type: "String"
    description: "Example parameter"
    default: "Hello World"
    allowedValues:
    - "Hello World"
mainSteps:
  - action: "aws:runPowerShellScript"
    name: "example"
    inputs:
      timeoutSeconds: '60'
      runCommand:
      - "Write-Output {{Message}}"
```

JSON

```
{
   "schemaVersion": "2.2",
   "description": "Example document",
```

```
   "parameters": {
      "Message": {
         "type": "String",
         "description": "Example parameter",
         "default": "Hello World",
         "allowedValues": ["Hello World"]
      }
   },
   "mainSteps": [
      {
         "action": "aws:runPowerShellScript",
         "name": "example",
         "inputs": {
            "timeoutSeconds": "60",
            "runCommand": [
               "Write-Output {{Message}}"
            ]
         }
      }
   ]
}
```

Schema der Version 2.2 – Vorbedingungsparameterbeispielen

Schema-Version 2.2 bietet Unterstützung für plattformübergreifende Aktionen. Dies bedeutet, dass Sie in einem SSM-Dokument unterschiedliche Betriebssysteme für verschiedene Plugins angeben können. Plattformübergreifende Aktionen werden durch den Parameter `precondition` in einem Schritt aufgerufen, wie in dem folgenden Beispiel dargestellt. Sie können auch den `precondition`-Parameter verwenden, um Eingabeparameter zu validieren, die Sie in Ihrem SSM-Dokument definiert haben. Dies sehen Sie im zweiten der folgenden Beispiele.

YAML

```
---
schemaVersion: '2.2'
description: cross-platform sample
mainSteps:
- action: aws:runPowerShellScript
  name: PatchWindows
  precondition:
    StringEquals:
    - platformType
```

```
    - Windows
  inputs:
    runCommand:
    - cmds
- action: aws:runShellScript
  name: PatchLinux
  precondition:
    StringEquals:
    - platformType
    - Linux
  inputs:
    runCommand:
    - cmds
```

JSON

```
{
   "schemaVersion": "2.2",
   "description": "cross-platform sample",
   "mainSteps": [
      {
         "action": "aws:runPowerShellScript",
         "name": "PatchWindows",
         "precondition": {
            "StringEquals": [
               "platformType",
               "Windows"
            ]
         },
         "inputs": {
            "runCommand": [
               "cmds"
            ]
         }
      },
      {
         "action": "aws:runShellScript",
         "name": "PatchLinux",
         "precondition": {
            "StringEquals": [
               "platformType",
               "Linux"
            ]
```

```
         },
         "inputs": {
            "runCommand": [
               "cmds"
            ]
         }
      }
   ]
}
```

YAML

```
---
schemaVersion: '2.2'
parameters:
  action:
    type: String
    allowedValues:
    - Install
    - Uninstall
  confirmed:
    type: String
    allowedValues:
    - True
    - False
mainSteps:
- action: aws:runShellScript
  name: InstallAwsCLI
  precondition:
    StringEquals:
    - "{{ action }}"
    - "Install"
  inputs:
    runCommand:
    - sudo apt install aws-cli
- action: aws:runShellScript
  name: UninstallAwsCLI
  precondition:
    StringEquals:
    - "{{ action }} {{ confirmed }}"
    - "Uninstall True"
  inputs:
```

```
    runCommand:
    - sudo apt remove aws-cli
```

JSON

```
{
   "schemaVersion": "2.2",
   "parameters": {
      "action": {
         "type": "String",
         "allowedValues": [
            "Install",
            "Uninstall"
         ]
      },
      "confirmed": {
         "type": "String",
         "allowedValues": [
            true,
            false
         ]
      }
   },
   "mainSteps": [
      {
         "action": "aws:runShellScript",
         "name": "InstallAwsCLI",
         "precondition": {
            "StringEquals": [
               "{{ action }}",
               "Install"
            ]
         },
         "inputs": {
            "runCommand": [
               "sudo apt install aws-cli"
            ]
         }
      },
      {
         "action": "aws:runShellScript",
         "name": "UninstallAwsCLI",
         "precondition": {
```

```
            "StringEquals": [
               "{{ action }} {{ confirmed }}",
               "Uninstall True"
            ]
         },
         "inputs": {
            "runCommand": [
               "sudo apt remove aws-cli"
            ]
         }
      }
   ]
}
```

Interpolationsbeispiel für Schemaversion 2.2 mit SSM Agent-Versionen vor 3.3.2746.0

In SSM Agent-Versionen vor 3.3.2746.0 ignoriert der Agent den `interpolationType`-Parameter und ersetzt stattdessen einfach die Zeichenfolge. Wenn Sie `SSM_`*`parameter-name`* referenzieren, müssen Sie dies explizit festlegen. Im folgenden Beispiel für Linux wird explizit auf die Umgebungsvariable `SSM_Message` verwiesen.

```
{
    "schemaVersion": "2.2",
    "description": "An example document",
    "parameters": {
        "Message": {
            "type": "String",
            "description": "Message to be printed",
            "default": "Hello",
            "interpolationType" : "ENV_VAR",
      "allowedPattern: "^[^"]*$"

        }
    },
    "mainSteps": [{
        "action": "aws:runShellScript",
        "name": "printMessage",
        "inputs": {
            "runCommand": [
              "if [ -z "${SSM_Message+x}" ]; then",
              "    export SSM_Message=\"{{Message}}\"",
```

```
            "fi",
            "",
            "echo $SSM_Message"
          ]
        }
    }
}
```

> **Note**
>
> `allowedPattern` ist streng genommen nicht erforderlich, wenn ein SSM-Dokument keine doppelten Klammern verwendet: `{{ }}`

Schema-Version 2.2 State Manager-Beispiel

Sie können das folgende SSM-Dokument mit State Manager nutzen, einem Tool in Systems Manager, um die ClamAV-Antivirensoftware herunterzuladen und zu installieren. State Manager erzwingt eine bestimmte Konfiguration, d. h. jedes Mal, wenn die State Manager-Zuordnung ausgeführt wird, prüft das System, ob die ClamAV-Software installiert ist. Ist dies nicht der Fall, führt State Manager dieses Dokument erneut aus.

YAML

```
---
schemaVersion: '2.2'
description: State Manager Bootstrap Example
parameters: {}
mainSteps:
- action: aws:runShellScript
  name: configureServer
  inputs:
    runCommand:
    - sudo yum install -y httpd24
    - sudo yum --enablerepo=epel install -y clamav
```

JSON

```
{
   "schemaVersion": "2.2",
   "description": "State Manager Bootstrap Example",
```

```
   "parameters": {},
   "mainSteps": [
      {
         "action": "aws:runShellScript",
         "name": "configureServer",
         "inputs": {
            "runCommand": [
               "sudo yum install -y httpd24",
               "sudo yum --enablerepo=epel install -y clamav"
            ]
         }
      }
   ]
}
```

Schema Version 2.2 - Bestandsbeispiel

Sie können das folgende SSM-Dokument mit State Manager verwenden, um Bestandsmetadaten zu Ihren Instances zu erfassen.

YAML

```
---
schemaVersion: '2.2'
description: Software Inventory Policy Document.
parameters:
  applications:
    type: String
    default: Enabled
    description: "(Optional) Collect data for installed applications."
    allowedValues:
    - Enabled
    - Disabled
  awsComponents:
    type: String
    default: Enabled
    description: "(Optional) Collect data for AWS Components like amazon-ssm-agent."
    allowedValues:
    - Enabled
    - Disabled
  networkConfig:
    type: String
```

```
    default: Enabled
    description: "(Optional) Collect data for Network configurations."
    allowedValues:
    - Enabled
    - Disabled
  windowsUpdates:
    type: String
    default: Enabled
    description: "(Optional) Collect data for all Windows Updates."
    allowedValues:
    - Enabled
    - Disabled
  instanceDetailedInformation:
    type: String
    default: Enabled
    description: "(Optional) Collect additional information about the instance,
 including
      the CPU model, speed, and the number of cores, to name a few."
    allowedValues:
    - Enabled
    - Disabled
  customInventory:
    type: String
    default: Enabled
    description: "(Optional) Collect data for custom inventory."
    allowedValues:
    - Enabled
    - Disabled
mainSteps:
- action: aws:softwareInventory
  name: collectSoftwareInventoryItems
  inputs:
    applications: "{{ applications }}"
    awsComponents: "{{ awsComponents }}"
    networkConfig: "{{ networkConfig }}"
    windowsUpdates: "{{ windowsUpdates }}"
    instanceDetailedInformation: "{{ instanceDetailedInformation }}"
    customInventory: "{{ customInventory }}"
```

JSON

```
{
   "schemaVersion": "2.2",
```

```
   "description": "Software Inventory Policy Document.",
   "parameters": {
      "applications": {
         "type": "String",
         "default": "Enabled",
         "description": "(Optional) Collect data for installed applications.",
         "allowedValues": [
            "Enabled",
            "Disabled"
         ]
      },
      "awsComponents": {
         "type": "String",
         "default": "Enabled",
         "description": "(Optional) Collect data for AWS Components like amazon-ssm-
agent.",
         "allowedValues": [
            "Enabled",
            "Disabled"
         ]
      },
      "networkConfig": {
         "type": "String",
         "default": "Enabled",
         "description": "(Optional) Collect data for Network configurations.",
         "allowedValues": [
            "Enabled",
            "Disabled"
         ]
      },
      "windowsUpdates": {
         "type": "String",
         "default": "Enabled",
         "description": "(Optional) Collect data for all Windows Updates.",
         "allowedValues": [
            "Enabled",
            "Disabled"
         ]
      },
      "instanceDetailedInformation": {
         "type": "String",
         "default": "Enabled",
```

```
         "description": "(Optional) Collect additional information about the
 instance, including\nthe CPU model, speed, and the number of cores, to name a
 few.",
         "allowedValues": [
            "Enabled",
            "Disabled"
         ]
      },
      "customInventory": {
         "type": "String",
         "default": "Enabled",
         "description": "(Optional) Collect data for custom inventory.",
         "allowedValues": [
            "Enabled",
            "Disabled"
         ]
      }
   },
   "mainSteps": [
      {
         "action": "aws:softwareInventory",
         "name": "collectSoftwareInventoryItems",
         "inputs": {
            "applications": "{{ applications }}",
            "awsComponents": "{{ awsComponents }}",
            "networkConfig": "{{ networkConfig }}",
            "windowsUpdates": "{{ windowsUpdates }}",
            "instanceDetailedInformation": "{{ instanceDetailedInformation }}",
            "customInventory": "{{ customInventory }}"
         }
      }
   ]
}
```

Schema-Version 2.2 **`AWS-ConfigureAWSPackage`**-Beispiel

Das folgende Beispiel zeigt das `AWS-ConfigureAWSPackage`-Dokument. Der Abschnitt `mainSteps` enthält das `aws:configurePackage`-Plugin im Schritt `action`.

> **Note**
>
> In Linux-Betriebssystemen werden nur die `AmazonCloudWatchAgent`- und `AWSSupport-EC2Rescue`-Pakete unterstützt.

YAML

```
---
schemaVersion: '2.2'
description: 'Install or uninstall the latest version or specified version of an AWS
package. Available packages include the following: AWSPVDriver, AwsEnaNetworkDriver,
  AwsVssComponents, and AmazonCloudWatchAgent, and AWSSupport-EC2Rescue.'
parameters:
  action:
    description: "(Required) Specify whether or not to install or uninstall the
 package."
    type: String
    allowedValues:
    - Install
    - Uninstall
  name:
    description: "(Required) The package to install/uninstall."
    type: String
    allowedPattern: "^arn:[a-z0-9][-.a-z0-9]{0,62}:[a-z0-9][-.a-z0-9]{0,62}:([a-
z0-9][-.a-z0-9]{0,62})?:([a-z0-9][-.a-z0-9]{0,62})?:package\\/[a-zA-Z][a-zA-Z0-9\\-
_]{0,39}$|^[a-zA-Z][a-zA-Z0-9\\-_]{0,39}$"
  version:
    type: String
    description: "(Optional) A specific version of the package to install or
 uninstall."
mainSteps:
- action: aws:configurePackage
  name: configurePackage
  inputs:
    name: "{{ name }}"
    action: "{{ action }}"
    version: "{{ version }}"
```

JSON

```
{
```

```
   "schemaVersion": "2.2",
   "description": "Install or uninstall the latest version or specified version
 of an AWS package. Available packages include the following: AWSPVDriver,
 AwsEnaNetworkDriver, AwsVssComponents, and AmazonCloudWatchAgent, and AWSSupport-
EC2Rescue.",
   "parameters": {
      "action": {
         "description":"(Required) Specify whether or not to install or uninstall
 the package.",
         "type":"String",
         "allowedValues":[
            "Install",
            "Uninstall"
         ]
      },
      "name": {
         "description": "(Required) The package to install/uninstall.",
         "type": "String",
         "allowedPattern": "^arn:[a-z0-9][-.a-z0-9]{0,62}:[a-z0-9][-.a-z0-9]{0,62}:
([a-z0-9][-.a-z0-9]{0,62})?:([a-z0-9][-.a-z0-9]{0,62})?:package\\/[a-zA-Z][a-zA-
Z0-9\\-_]{0,39}$|^[a-zA-Z][a-zA-Z0-9\\-_]{0,39}$"
      },
      "version": {
         "type": "String",
         "description": "(Optional) A specific version of the package to install or
 uninstall."
      }
   },
   "mainSteps":[
      {
         "action": "aws:configurePackage",
         "name": "configurePackage",
         "inputs": {
            "name": "{{ name }}",
            "action": "{{ action }}",
            "version": "{{ version }}"
         }
      }
   ]
}
```

Schema der Version 1.2

Das folgende Beispiel zeigt die Elemente der obersten Ebene eines Dokuments in Schema-Version 1.2.

```
{
   "schemaVersion":"1.2",
   "description":"A description of the SSM document.",
   "parameters":{
      "parameter 1":{
         "one or more parameter properties"
      },
      "parameter 2":{
         "one or more parameter properties"
      },
      "parameter 3":{
         "one or more parameter properties"
      }
   },
   "runtimeConfig":{
      "plugin 1":{
         "properties":[
            {
               "one or more plugin properties"
            }
         ]
      }
   }
}
```

Schema-Version 1.2 **aws:runShellScript**-Beispiel

Das folgende Beispiel zeigt das `AWS-RunShellScript` SSM-Dokument. Der Abschnitt runtimeConfig bindet das Plugin `aws:runShellScript` ein.

```
{
    "schemaVersion":"1.2",
    "description":"Run a shell script or specify the commands to run.",
    "parameters":{
        "commands":{
            "type":"StringList",
            "description":"(Required) Specify a shell script or a command to run.",
            "minItems":1,
```

```
            "displayType":"textarea"
        },
        "workingDirectory":{
            "type":"String",
            "default":"",
            "description":"(Optional) The path to the working directory on your
 instance.",
            "maxChars":4096
        },
        "executionTimeout":{
            "type":"String",
            "default":"3600",
            "description":"(Optional) The time in seconds for a command to complete
 before it is considered to have failed. Default is 3600 (1 hour). Maximum is 172800
 (48 hours).",
            "allowedPattern":"([1-9][0-9]{0,3})|(1[0-9]{1,4})|(2[0-7][0-9]{1,3})|
(28[0-7][0-9]{1,2})|(28800)"
        }
    },
    "runtimeConfig":{
        "aws:runShellScript":{
            "properties":[
                {
                    "id":"0.aws:runShellScript",
                    "runCommand":"{{ commands }}",
                    "workingDirectory":"{{ workingDirectory }}",
                    "timeoutSeconds":"{{ executionTimeout }}"
                }
            ]
        }
    }
}
```

Schema der Version 0.3

Top-Level-Elemente

Im folgenden Beispiel werden die Elemente der obersten Ebene eines Automation-Runbook der Schema-Version 0.3im JSON-Format gezeigt.

```
{
    "description": "document-description",
    "schemaVersion": "0.3",
    "assumeRole": "{{assumeRole}}",
```

```
    "parameters": {
        "parameter1": {
            "type": "String",
            "description": "parameter-1-description",
            "default": ""
        },
        "parameter2": {
            "type": "String",
            "description": "parameter-2-description",
            "default": ""
        }
    },
    "variables": {
        "variable1": {
            "type": "StringMap",
            "description": "variable-1-description",
            "default": {}
        },
        "variable2": {
            "type": "String",
            "description": "variable-2-description",
            "default": "default-value"
        }
    },
    "mainSteps": [
        {
            "name": "myStepName",
            "action": "action-name",
            "maxAttempts": 1,
            "inputs": {
                "Handler": "python-only-handler-name",
                "Runtime": "runtime-name",
                "Attachment": "script-or-zip-name"
            },
            "outputs": {
                "Name": "output-name",
                "Selector": "selector.value",
                "Type": "data-type"
            }
        }
    ],
    "files": {
        "script-or-zip-name": {
            "checksums": {
```

```
                "sha256": "checksum"
            },
            "size": 1234
        }
    }
}
```

Beispiel für YAML-Automation-Runbook

Das folgende Beispiel zeigt den Inhalt eines Automation-Runbooks im YAML-Format. In diesem funktionierenden Beispiel der Version 0.3 des Dokumentschemas wird auch die Verwendung von Markdown zur Formatierung von Dokumentbeschreibungen veranschaulicht.

```
description: >-
  ##Title: LaunchInstanceAndCheckState

  -----

  **Purpose**: This Automation runbook first launches an EC2 instance
  using the AMI ID provided in the parameter ```imageId```. The second step of
  this document continuously checks the instance status check value for the
  launched instance until the status ```ok``` is returned.


  ##Parameters:

  -----

  Name | Type | Description | Default Value

  ------------ | ------------ | ------------ | ------------

  assumeRole | String | (Optional) The ARN of the role that allows Automation to
  perform the actions on your behalf. | -

  imageId  | String | (Optional) The AMI ID to use for launching the instance.
  The default value uses the latest Amazon Linux AMI ID available. | {{
  ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64 }}
schemaVersion: '0.3'
assumeRole: 'arn:aws:iam::111122223333::role/AutomationServiceRole'
parameters:
  imageId:
    type: String
```

```
    default: '{{ ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-
x86_64 }}'
    description: >-
      (Optional) The AMI ID to use for launching the instance. The default value
      uses the latest released Amazon Linux AMI ID.
  tagValue:
    type: String
    default: ' LaunchedBySsmAutomation'
    description: >-
      (Optional) The tag value to add to the instance. The default value is
      LaunchedBySsmAutomation.
  instanceType:
    type: String
    default: t2.micro
    description: >-
      (Optional) The instance type to use for the instance. The default value is
      t2.micro.
mainSteps:
  - name: LaunchEc2Instance
    action: 'aws:executeScript'
    outputs:
      - Name: payload
        Selector: $.Payload
        Type: StringMap
    inputs:
      Runtime: python3.11
      Handler: launch_instance
      Script: ''
      InputPayload:
        image_id: '{{ imageId }}'
        tag_value: '{{ tagValue }}'
        instance_type: '{{ instanceType }}'
      Attachment: launch.py
    description: >-
      **About This Step**


      This step first launches an EC2 instance using the ```aws:executeScript```
      action and the provided python script.
  - name: WaitForInstanceStatusOk
    action: 'aws:executeScript'
    inputs:
      Runtime: python3.11
      Handler: poll_instance
```

```
      Script: |-
        def poll_instance(events, context):
          import boto3
          import time

          ec2 = boto3.client('ec2')

          instance_id = events['InstanceId']

          print('[INFO] Waiting for instance status check to report ok', instance_id)

          instance_status = "null"

          while True:
            res = ec2.describe_instance_status(InstanceIds=[instance_id])

            if len(res['InstanceStatuses']) == 0:
              print("Instance status information is not available yet")
              time.sleep(5)
              continue

            instance_status = res['InstanceStatuses'][0]['InstanceStatus']['Status']

            print('[INFO] Polling to get status of the instance', instance_status)

            if instance_status == 'ok':
              break

            time.sleep(10)

          return {'Status': instance_status, 'InstanceId': instance_id}
      InputPayload: '{{ LaunchEc2Instance.payload }}'
    description: >-
      **About This Step**


      The python script continuously polls the instance status check value for
      the instance launched in Step 1 until the ```ok``` status is returned.
files:
  launch.py:
    checksums:
      sha256: 18871b1311b295c43d0f...[truncated]...772da97b67e99d84d342ef4aEXAMPLE
```

Beispiele für die sichere Parameterverarbeitung

In den folgenden Beispielen wird die sichere Verarbeitung von Parametern mithilfe der Umgebungsvariable `interpolationType` gezeigt.

Grundlegende sichere Befehlsausführung

Dieses Beispiel zeigt, wie ein Befehlsparameter sicher verarbeitet werden kann:

> **Note**
>
> `allowedPattern` ist in SSD-Dokumenten streng genommen nicht erforderlich, wenn sie keine doppelten Klammern enthalten: `{{ }}`

YAML

```
---

schemaVersion: '2.2'
description: An example document.
parameters:
  Message:
    type: String
    description: "Message to be printed"
    default: Hello
    interpolationType: ENV_VAR
    allowedPattern: "^[^"]*$"
mainSteps:
  - action: aws:runShellScript
    name: printMessage
    precondition:
      StringEquals:
        - platformType
        - Linux
    inputs:
      runCommand:
        - echo {{Message}}
```

JSON

```
{
```

```
    "schemaVersion": "2.2",
    "description": "An example document.",
    "parameters": {
        "Message": {
            "type": "String",
            "description": "Message to be printed",
            "default": "Hello",
            "interpolationType": "ENV_VAR",
            "allowedPattern": "^[^"]*$"
        }
    },
    "mainSteps": [{
        "action": "aws:runShellScript",
        "name": "printMessage",
        "precondition": {
           "StringEquals": ["platformType", "Linux"]
        },
        "inputs": {
            "runCommand": [
              "echo {{Message}}"
            ]
        }
    }]
}
```

Verwenden von Parametern in interpretierten Sprachen

Dieses Beispiel zeigt die sichere Parameterverarbeitung in Python:

YAML

```
---
schemaVersion: '2.2'
description: 'Secure Python script execution'
parameters:
  inputData:
    type: String
    description: 'Input data for processing'
    interpolationType: 'ENV_VAR'
mainSteps:
  - action: aws:runPowerShellScript
    name: runPython
    inputs:
```

```
      runCommand:
        - |
          python3 -c '
          import os
          import json

          # Safely access parameter through environment variable
          input_data = os.environ.get("SSM_inputData", "")

          # Process the data
          try:
              processed_data = json.loads(input_data)
              print(f"Successfully processed: {processed_data}")
          except json.JSONDecodeError:
              print("Invalid JSON input")
          '
```

Beispiel für die Abwärtskompatibilität

Dieses Beispiel zeigt, wie Sie Parameter sicher verarbeiten und die Abwärtskompatibilität wahren können:

YAML

```
---
schemaVersion: '2.2'
description: 'Backwards compatible secure parameter handling'
parameters:
  userInput:
    type: String
    description: 'User input to process'
    interpolationType: 'ENV_VAR'
    allowedPattern: '^[^"]*$'

mainSteps:
  - action: aws:runShellScript
    name: processInput
    inputs:
      runCommand:
        - |
          # Handle both modern and legacy agent versions
          if [ -z "${SSM_userInput+x}" ]; then
```

```
        # Legacy agent - fall back to direct parameter reference
        export SSM_userInput="{{userInput}}"
    fi

    # Process the input securely
    echo "Processing input: $SSM_userInput"
```

> **Note**
>
> `allowedPattern` ist in SSD-Dokumenten streng genommen nicht erforderlich, wenn sie keine doppelten Klammern enthalten: `{{ }}`

Bewährte Methoden für die Parametersicherheit

Folgen Sie diesen bewährten Methoden beim Verarbeiten von Parametern in SSM-Dokumenten:

- Interpolation von Umgebungsvariablen nutzen: Verwenden Sie immer `interpolationType: "ENV_VAR"` für Zeichenfolgeparameter, die in der Befehlsausführung verwendet werden.
- Eingabevalidierung implementieren: Verwenden Sie `allowedPattern`, um Parameterwerte auf sichere Muster zu beschränken.
- Legacy-Systeme: Fügen Sie Fallback-Logik für ältere Versionen von SSM Agent hinzu, die die Interpolation von Umgebungsvariablen nicht unterstützen.
- Sonderzeichen maskieren: Wenn Sie Parameterwerte in Befehlen verwenden, sollten Sie Sonderzeichen korrekt maskieren, um eine Interpretation durch die Shell zu verhindern.
- Parameterbereich einschränken: Verwenden Sie die restriktivsten Parametermuster, die für Ihren Anwendungsfall möglich sind.

Datenelemente und Parameter

In diesem Thema werden die in SSM-Dokumenten verwendeten Datenelemente beschrieben. Die Schemaversion, die zum Erstellen eines Dokuments verwendet wird, definiert die Syntax und die Datenelemente, die das Dokument akzeptiert. Es wird empfohlen, Schema-Version 2.2 oder höher für Befehlsdokumente zu verwenden. Automation-Runbooks verwenden die Schema-Version 0.3. Automation-Runbooks unterstützen darüber hinaus die Verwendung von Markdown, einer Markup-Sprache, mit der Sie Wiki-Beschreibungen zu Dokumenten und einzelnen Schritten innerhalb des Dokuments hinzufügen können. Weitere Informationen zur Verwendung von Markdown finden Sie

unter Verwenden von Markdown in der Konsole im AWS-Managementkonsole -Handbuch „Erste Schritte“.

Im folgenden Abschnitt werden die Datenelemente beschrieben, die Sie in ein SSM-Dokument aufnehmen können.

Top-Level-Datenelemente

schemaVersion

Die zu verwendende Schema-Version.

Typ: Version

Erforderlich: Ja

description

Von Ihnen angegebene Informationen, um den Zweck des Dokuments zu beschreiben. Sie können dieses Feld auch verwenden, um anzugeben, ob ein Parameter einen Wert für die Ausführung eines Dokuments benötigt oder ob die Bereitstellung eines Werts für den Parameter optional ist. Erforderliche und optionale Parameter sind in den Beispielen dieses Themas zu sehen.

Typ: Zeichenfolge

Erforderlich: Nein

Parameter

Eine Struktur, die die Parameter definiert, die das Dokument akzeptiert.

Um die Sicherheit bei der Verarbeitung von Zeichenfolgeparametern zu erhöhen, können Sie die Interpolation von Umgebungsvariablen verwenden, indem Sie die Eigenschaft `interpolationType` angeben. Wenn dies auf `ENV_VAR` gesetzt ist, wird eine Umgebungsvariable mit dem Namen `SSM_`*`parameter-name`* erstellt, die den Parameterwert enthält.

Das folgende Beispiel enthält einen Parameter mit der Umgebungsvariable `interpolationType`:

```
{
```

```
    "schemaVersion": "2.2",
    "description": "An example document.",
    "parameters": {
        "Message": {
            "type": "String",
            "description": "Message to be printed",
            "default": "Hello",
            "interpolationType" : "ENV_VAR",
            "allowedPattern": "^[^"]*$"

        }
    },
    "mainSteps": [{
        "action": "aws:runShellScript",
        "name": "printMessage",
        "precondition" : {
           "StringEquals" : ["platformType", "Linux"]
        },
        "inputs": {
            "runCommand": [
              "echo {{Message}}"
            ]
        }
    }
}
```

> **Note**
>
> `allowedPattern` ist in SSD-Dokumenten streng genommen nicht erforderlich, wenn sie keine doppelten Klammern enthalten: `{{ }}`

Wir empfehlen, häufig verwendete Parameter in Parameter Store zu speichern, einem Tool in AWS Systems Manager. Anschließend können Sie Parameter in Ihrem Dokument definieren, die Parameter Store-Parameter als Standardwert referenzieren. Um auf einen Parameter Store-Parameter zu verweisen, verwenden Sie die folgende Syntax.

```
{{ssm:parameter-name}}
```

Sie können einen Parameter, der auf einen Parameter Store-Parameter verweist, auf die gleiche Weise wie alle anderen Dokumentparameter verwenden. Im folgenden Beispiel ist der

Standardwert für den `commands`-Parameter der Parameter Store-Parameter `myShellCommands`. Durch Angabe des `commands`-Parameters als `runCommand`-Zeichenfolge führt das Dokument die im `myShellCommands`-Parameter gespeicherten Befehle aus.

YAML

```
---
schemaVersion: '2.2'
description: runShellScript with command strings stored as Parameter Store
 parameter
parameters:
  commands:
    type: StringList
    description: "(Required) The commands to run on the instance."
    default: ["{{ ssm:myShellCommands }}"],
            interpolationType : 'ENV_VAR'
            allowedPattern: '^[^"]*$'

mainSteps:
- action: aws:runShellScript
  name: runShellScriptDefaultParams
  inputs:
    runCommand:"{{ commands }}"
```

JSON

```
{
    "schemaVersion": "2.2",
    "description": "runShellScript with command strings stored as Parameter Store
 parameter",
    "parameters": {
      "commands": {
        "type": "StringList",
        "description": "(Required) The commands to run on the instance.",
        "default": ["{{ ssm:myShellCommands }}"],
        "interpolationType" : "ENV_VAR"
      }
    },
    "mainSteps": [
      {
        "action": "aws:runShellScript",
        "name": "runShellScriptDefaultParams",
        "inputs": {
```

```
            "runCommand": [
              "{{ commands }}"
          ]
        }
      }
    ]
  }
```

> **Note**
>
> Im `parameters`-Abschnitt Ihres Dokuments können Sie die -Parameter `String` und `StringList` Parameter Store referenzieren. Sie können nicht die Parameter `SecureString` Parameter Store referenzieren.

Mehr über Parameter Store erfahren Sie unter AWS Systems Manager Parameter Store.

Typ: Struktur

Die `parameters`-Struktur akzeptiert die folgenden Felder und Werte:

- `type`: (Erforderlich) Zulässige Werte umfassen die Folgenden: `String`, `StringList`, `Integer`, `Boolean`, `MapList` und `StringMap`. Beispiele für jeden Typ finden Sie Beispiele für den Parameter `type` in SSM-Dokumenten im nächsten Abschnitt.

  > **Note**
  >
  > Befehlstyp-Dokumente unterstützen nur die Parametertypen `String` und `StringList`.

- `description`: (Optional) Eine Beschreibung der Parametergruppe.
- `default`: (Optional) Der Standardwert des Parameters oder eine Referenz bezüglich eines Parameters in Parameter Store.
- `allowedValues`: (Optional) Ein Array von Werten, die für den Parameter zulässig sind. Durch das Definieren zulässiger Werte für den Parameter wird die Benutzereingabe überprüft. Wenn ein Benutzer einen Wert eingibt, der nicht zulässig ist, kann die Ausführung nicht gestartet werden.

  YAML

  ```
  DirectoryType:
    type: String
  ```

```
  description: "(Required) The directory type to launch."
  default: AwsMad
  allowedValues:
  - AdConnector
  - AwsMad
  - SimpleAd
```

JSON

```
"DirectoryType": {
  "type": "String",
  "description": "(Required) The directory type to launch.",
  "default": "AwsMad",
  "allowedValues": [
    "AdConnector",
    "AwsMad",
    "SimpleAd"
  ]
}
```

- `allowedPattern`: (Optional) Ein regulärer Ausdruck, der überprüft, ob die Benutzereingabe mit dem definierten Muster für den Parameter übereinstimmt. Wenn die Benutzereingabe nicht mit dem zulässigen Muster übereinstimmt, kann die Ausführung nicht gestartet werden.

> **Note**
>
> Systems Manager führt zwei Validierungen für `allowedPattern` aus. Die erste Validierung erfolgt unter Verwendung der Java regex library (Java-Regex-Bibliothek) auf API-Ebene, wenn Sie ein Dokument verwenden. Die zweite Validierung wird am SSM Agent ausgeführt, indem Sie die GO regexp library (GO-regexp-Bibliothek) verwenden, bevor Sie das Dokument bearbeiten.

YAML

```
InstanceId:
  type: String
  description: "(Required) The instance ID to target."
  allowedPattern: "^i-(?:[a-f0-9]{8}|[a-f0-9]{17})$"
  default: ''
```

JSON

```
"InstanceId": {
  "type": "String",
  "description": "(Required) The instance ID to target.",
  "allowedPattern": "^i-(?:[a-f0-9]{8}|[a-f0-9]{17})$",
  "default": ""
}
```

- `displayType`: (Optional) Wird verwendet, um entweder a `textfield` oder a `textarea` in der anzuzeigen AWS-Managementkonsole. `textfield`dist ein einzeiliges Textfeld. `textarea`ist ein mehrzeiliger Textbereich.
- `minItems`: (Optional) Die minimal zulässige Anzahl von Elementen.
- `maxItems`: (Optional) Die maximal zulässige Anzahl von Elementen.
- `minChars`: (Optional) Die minimal zulässige Anzahl an Parameterzeichen.
- `maxChars`: (Optional) Die maximal zulässige Anzahl an Parameterzeichen.
- `interpolationType`: (Optional) Definiert, wie Parameterwerte vor der Befehlsausführung verarbeitet werden. Wenn dies auf `ENV_VAR` gesetzt ist, wird der Parameterwert als Umgebungsvariable mit dem Namen `SSM_`*`parameter-name`* verfügbar gemacht. Dieses Feature trägt dazu bei, die Einschleusung von Befehlen zu verhindern, da Parameterwerte als Literalzeichenfolgen behandelt werden.

  Typ: Zeichenfolge

  Zulässige Werte: `ENV_VAR`

Erforderlich: Nein

variables

(Nur Schemaversion 0.3) Werte, auf die Sie während der einzelnen Schritte in einem Automation-Runbook verweisen oder diese aktualisieren können. Variablen ähneln Parametern, unterscheiden sich jedoch in einem sehr wichtigen Punkt. Parameterwerte sind im Kontext eines Runbooks statisch, aber die Werte von Variablen können im Kontext des Runbooks geändert werden. Beim Aktualisieren des Werts einer Variable muss der Datentyp dem definierten Datentyp entsprechen. Hinweise zum Aktualisieren von Variablenwerten in einer Automatisierung finden Sie unter [`aws:updateVariable` – Aktualisiert einen Wert für eine Runbook-Variable].

Typ: Boolean | Integer | | Zeichenfolge | MapList | StringList StringMap

Erforderlich: Nein

YAML

```
variables:
    payload:
        type: StringMap
        default: "{}"
```

JSON

```
{
    "variables": [
        "payload": {
            "type": "StringMap",
            "default": "{}"
        }
    ]
}
```

runtimeConfig

(Nur für Schemaversion 1.2) Die Konfiguration für die Instance, wie sie von mindestens einem Systems Manager-Plugin verwendet wird. Es wird nicht garantiert, dass Plugins nacheinander ausgeführt werden.

Typ: Dictionary<String, > PluginConfiguration

Erforderlich: Nein

mainSteps

(Nur Schema-Version 0.3, 2.0 und 2.2) Ein Objekt, das mehrere Schritte (Plugins) enthalten kann. Plugins werden innerhalb von Schritten definiert. Die Schritte werden in der Reihenfolge ausgeführt, in der sie im Dokument aufgeführt sind.

Typ: Dictionary<String, > PluginConfiguration

Erforderlich: Ja

outputs

(Nur Schema-Version 0.3) Daten, die durch die Ausführung dieses Dokuments generiert werden, die in anderen Prozessen verwendet werden können. Wenn Ihr Dokument beispielsweise ein neues erstelltAMI, können Sie "angeben. CreateImage ImageId"als Ausgabewert und

verwenden Sie diese Ausgabe dann, um in einer nachfolgenden Automatisierungsausführung neue Instanzen zu erstellen. Weitere Informationen zu Ausgaben finden Sie unter Verwenden von Aktionsausgaben als Eingaben.

Geben Sie ein: Dictionary<String, > OutputConfiguration

Erforderlich: Nein

files

(Nur Schema-Version 0.3) Die Skriptdateien (und ihre Prüfsummen), die dem Dokument zugeordnet sind und während einer Automatisierungsausführung ausgeführt werden. Gilt nur für Dokumente, die die `aws:executeScript` Aktion enthalten und für die Anfügungen in einem oder mehreren Schritten angegeben wurden.

Weitere Informationen zu den von Automation-Runbooks unterstützten Laufzeiten finden Sie unter `aws:executeScript` - Führen Sie ein Skript aus. Weitere Informationen zum Einbinden von Skripten in Automation-Runbooks finden Sie unter Verwenden von Skripten in Runbooks und Visuelle Designerfahrung für Automation-Runbooks.

Wenn Sie ein Automatisierungs-Runbook mit Anlagen erstellen, müssen Sie auch Anhangsdateien mit der `--attachments` Option (für AWS CLI) oder `Attachments` (für API und SDK) angeben. Sie können den Dateispeicherort für SSM-Dokumente und Dateien angeben, die in Amazon Simple Storage Service (Amazon S3)-Buckets gespeichert sind. Weitere Informationen finden Sie in der AWS Systems Manager API-Referenz unter Anlagen.

YAML

```
---
files:
  launch.py:
    checksums:
      sha256: 18871b1311b295c43d0f...
[truncated]...772da97b67e99d84d342ef4aEXAMPLE
```

JSON

```
"files": {
    "launch.py": {
        "checksums": {
            "sha256": "18871b1311b295c43d0f...
[truncated]...772da97b67e99d84d342ef4aEXAMPLE"
        }
```

```
    }
}
```

Geben Sie ein: Dictionary<String, > FilesConfiguration

Erforderlich: Nein

Beispiele für den Parameter **`type`** in SSM-Dokumenten

Parametertypen in SSM-Dokumenten sind statisch. Dies bedeutet, dass der Parametertyp nicht geändert werden kann, nachdem er definiert wurde. Bei der Verwendung von Parametern mit SSM-Dokumenten-Plugins kann der Typ eines Parameters innerhalb der Eingabe eines Plugins nicht dynamisch geändert werden. Beispielsweise können Sie nicht auf einen `Integer`-Parameter innerhalb der Eingabe `runCommand` des Plugins `aws:runShellScript` verweisen, da diese Eingabe eine Zeichenfolge oder eine Liste von Zeichenfolgen akzeptiert. Um einen Parameter für eine Plugin-Eingabe verwenden zu können, muss der Parametertyp mit dem akzeptierten Typ übereinstimmen. Sie müssen beispielsweise einen Parameter des Typs `Boolean` für die Eingabe `allowDowngrade` des Plugins `aws:updateSsmAgent` angeben. Wenn der Parametertyp nicht mit dem Eingabetyp für ein Plugin übereinstimmt, kann das SSM-Dokument nicht validiert werden und das System erstellt das Dokument nicht. Dies gilt auch, wenn Parameter nachgeschaltet in Eingaben für andere Plugins oder Automatisierungsaktionen verwendet werden. AWS Systems Manager Sie können beispielsweise keinen `StringList`-Parameter in der `documentParameters`-Eingabe des Plugins `aws:runDocument` referenzieren. Die `documentParameters`-Eingabe akzeptiert eine Zuordnung von Zeichenfolgen, auch wenn der nachgelagerte Parametertyp des SSM-Dokuments ein `StringList`-Parameter ist und dem referenzierten Parameter entspricht.

Wenn Sie Parameter mit -Automation-Aktionen verwenden, werden Parametertypen bei der Erstellung des SSM-Dokuments in den meisten Fällen nicht validiert. Nur wenn Sie die Aktion `aws:runCommand` verwenden, werden Parametertypen bei der Erstellen des SSM-Dokuments validiert. In allen anderen Fällen erfolgt die Parametervalidierung während der Automatisierungsausführung, wenn die Eingabe einer Aktion überprüft wird, bevor die Aktion ausgeführt wird. Wenn der Eingabeparameter beispielsweise ein `String` ist und Sie auf ihn als Wert für die Eingabe `MaxInstanceCount` der Aktion `aws:runInstances` verweisen, wird das SSM-Dokument erstellt. Beim Ausführen des Dokuments schlägt die Automatisierung jedoch fehl, wenn die Aktion `aws:runInstances` validiert wird, da für die Eingabe `MaxInstanceCount` ein `Integer` erforderlich ist.

Im Folgenden finden Sie für jeden Parametertype ein Beispiel.

String

Eine Abfolge von null oder mehr Unicode-Zeichen in Anführungszeichen. Zum Beispiel „i-1234567890abcdef0“. Verwenden Sie umgekehrte Schrägstriche als Escapezeichen.

Zeichenfolgeparameter können ein optionales `interpolationType`-Feld mit dem Wert `ENV_VAR` enthalten, um die Interpolation von Umgebungsvariablen zu ermöglichen und so die Sicherheit zu erhöhen.

YAML

```
---
InstanceId:
  type: String
  description: "(Optional) The target EC2 instance ID."
  interpolationType: ENV_VAR
```

JSON

```
"InstanceId":{
  "type":"String",
  "description":"(Optional) The target EC2 instance ID.",
  "interpolationType": "ENV_VAR"
}
```

StringList

Eine Liste von String-Elementen, die durch Kommas getrennt sind. Zum Beispiel ["cd ~", "pwd"].

YAML

```
---
commands:
  type: StringList
  description: "(Required) Specify a shell script or a command to run."
  default: ""
  minItems: 1
  displayType: textarea
```

JSON

```
"commands":{
  "type":"StringList",
```

```
  "description":"(Required) Specify a shell script or a command to run.",
  "minItems":1,
  "displayType":"textarea"
}
```

Boolesch

Akzeptiert nur `true` oder `false`. Akzeptiert nicht „true“ oder 0.

YAML

```
---
canRun:
  type: Boolean
  description: ''
  default: true
```

JSON

```
"canRun": {
  "type": "Boolean",
  "description": "",
  "default": true
}
```

Ganzzahl

Ganze Zahlen. Akzeptiert keine Dezimalzahlen, z. B. 3,14159, oder Zahlen in Anführungszeichen, z. B. „3“.

YAML

```
---
timeout:
  type: Integer
  description: The type of action to perform.
  default: 100
```

JSON

```
"timeout": {
  "type": "Integer",
  "description": "The type of action to perform.",
  "default": 100
```

```
}
```

StringMap

Ein Mapping von Schlüsseln zu Werten. Schlüssel und Werte müssen Zeichenfolgen sein. Zum Beispiel {"Env": "Prod"}.

YAML

```
---
notificationConfig:
  type: StringMap
  description: The configuration for events to be notified about
  default:
    NotificationType: 'Command'
    NotificationEvents:
    - 'Failed'
    NotificationArn: "$dependency.topicArn"
  maxChars: 150
```

JSON

```
"notificationConfig" : {
  "type" : "StringMap",
  "description" : "The configuration for events to be notified about",
  "default" : {
    "NotificationType" : "Command",
    "NotificationEvents" : ["Failed"],
    "NotificationArn" : "$dependency.topicArn"
  },
  "maxChars" : 150
}
```

MapList

Eine Liste von StringMap Objekten.

YAML

```
blockDeviceMappings:
  type: MapList
  description: The mappings for the create image inputs
  default:
  - DeviceName: "/dev/sda1"
```

```
    Ebs:
      VolumeSize: "50"
  - DeviceName: "/dev/sdm"
    Ebs:
      VolumeSize: "100"
  maxItems: 2
```

JSON

```
"blockDeviceMappings":{
  "type":"MapList",
  "description":"The mappings for the create image inputs",
  "default":[
    {
      "DeviceName":"/dev/sda1",
      "Ebs":{
        "VolumeSize":"50"
      }
    },
    {
      "DeviceName":"/dev/sdm",
      "Ebs":{
        "VolumeSize":"100"
      }
    }
  ],
  "maxItems":2
}
```

Inhalte von SSM-Befehlsdokument anzeigen

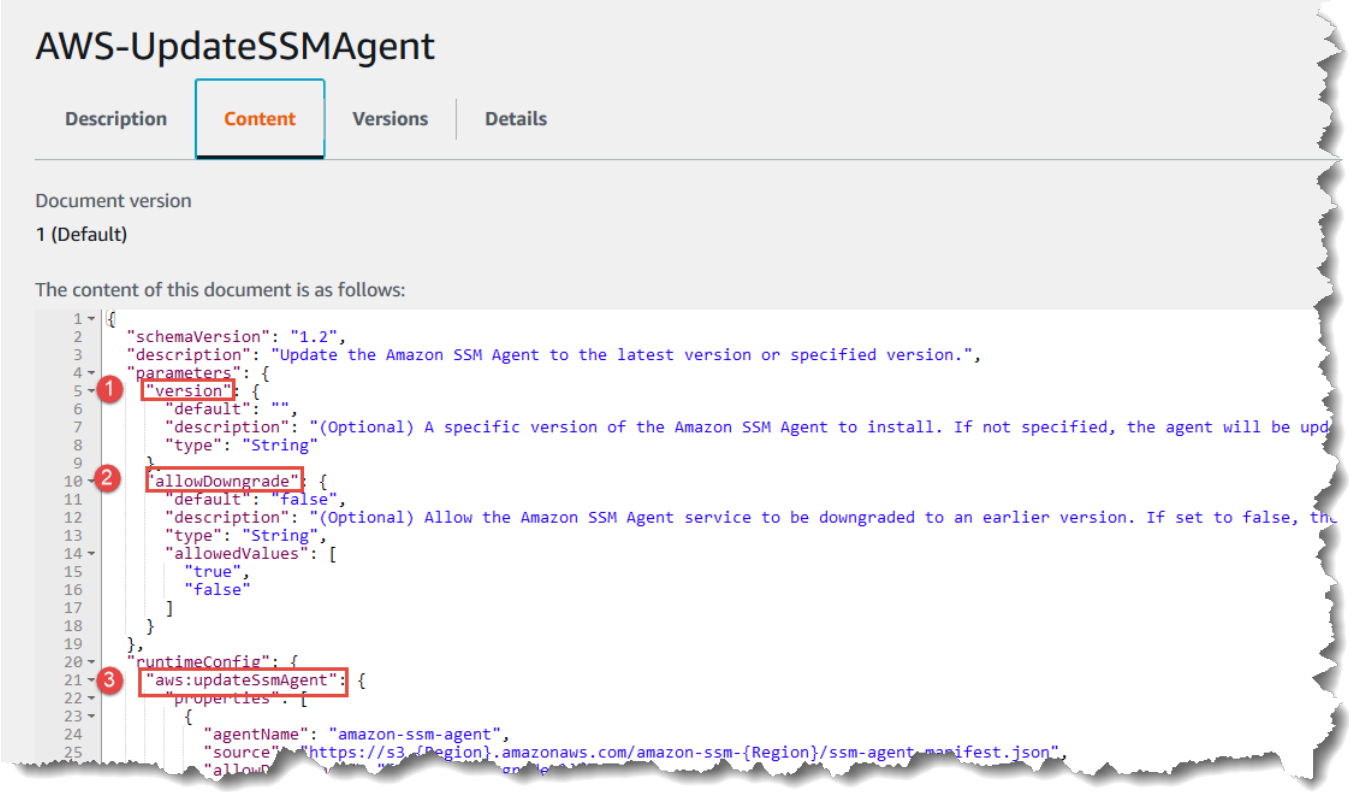
Um eine Vorschau der erforderlichen und optionalen Parameter für ein AWS Systems Manager (SSM-) Befehlsdokument anzuzeigen, können Sie zusätzlich zu den Aktionen, die das Dokument ausführt, den Inhalt des Dokuments in der Systems Manager Manager-Konsole anzeigen.

Inhalte von SSM-Befehlsdokument anzeigen

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie im Suchfeld Dokumenttyp und wählen Sie danach Befehl.

4. Wählen Sie den Namen eines Dokuments und dann die Registerkarte Content (Inhalt) aus.
5. Überprüfen Sie im Inhaltsfeld die verfügbaren Parameter und Aktionsschritte für das Dokument.

   Das folgende Image zeigt beispielsweise, dass (1) `version` und (2) `allowDowngrade` optionale Parameter für das `AWS-UpdateSSMAgent`-Dokument sind, und dass die erste Aktion, die vom Dokument ausgeführt wird, (3) `aws:updateSsmAgent` ist.



Referenz für Befehlsdokument-Plugins

Diese Referenz beschreibt die Plugins, die Sie in einem Dokument vom Typ AWS Systems Manager (SSM) Command angeben können. Diese Plugins können nicht in SSM-Automation-Runbooks verwendet werden, die Automation-Aktionen verwenden. Informationen zu AWS Systems Manager Automatisierungsaktionen finden Sie unterSystems Manager Automation Aktionen-Referenz.

Systems Manager bestimmt die Aktionen, die auf einer verwalteten Instance ausgeführt werden sollen, durch Lesen der Inhalte eines SSM-Dokuments. Jedes Dokument enthält einen Abschnitt zur Ausführung von Code. Abhängig von der Schemaversion des Dokuments umfasst dieser Abschnitt zu Codeausführung ein oder mehrere Plugins oder Schritte. Im Rahmen dieses Hilfethemas werden die Plugins und Schritte als Plugins bezeichnet. Dieser Abschnitt enthält Informationen zu allen Systems Manager-Plugins. Weitere Informationen zu Dokumenten, einschließlich Informationen zum

Erstellen von Dokumenten und zu den Unterschiede zwischen Schemaversionen finden Sie unter AWS Systems Manager-Documents.

Bei Plugins, die Zeichenfolgeparameter akzeptieren, z. B. `aws:runShellScript` und `aws:runPowerShellScript`, kann der Parameter `interpolationType` verwendet werden, um die Sicherheit zu erhöhen. Dazu werden Parametereingaben als Zeichenfolgeliterale und nicht als potenziell ausführbare Befehle behandelt. Zum Beispiel:

```
{
    "schemaVersion": "2.2",
    "description": "runShellScript with command strings stored as Parameter Store
 parameter",
    "parameters": {
      "commands": {
        "type": "StringList",
        "description": "(Required) The commands to run on the instance.",
        "default": ["{{ ssm:myShellCommands }}"],
        "interpolationType" : "ENV_VAR"
      }
    },
    //truncated
 }
```

> **Note**
>
> Manche der hier beschriebenen Plugins funktionieren nur auf Windows Server-Instances oder Linux-Instances. Für alle Plugins werden Plattformabhängigkeiten angegeben.
> Die folgenden Dokument-Plugins werden auf Amazon Elastic Compute Cloud (Amazon EC2) -Instances unterstützt fürmacOS:
>
> - `aws:refreshAssociation`
> - `aws:runShellScript`
> - `aws:runPowerShellScript`
> - `aws:softwareInventory`
> - `aws:updateSsmAgent`

Inhalt

- Gemeinsame Eingaben

- aws:applications
- aws:cloudWatch
- aws:configureDocker
- aws:configurePackage
- aws:domainJoin
- aws:downloadContent
- aws:psModule
- aws:refreshAssociation
- aws:runDockerAction
- aws:runDocument
- aws:runPowerShellScript
- aws:runShellScript
- aws:softwareInventory
- aws:updateAgent
- aws:updateSsmAgent

Gemeinsame Eingaben

Mit SSM Agent Version 3.0.502 und höher, können alle Plugins die folgenden Eingaben verwenden:

finallyStep

Der letzte Schritt, in dem das Dokument ausgeführt werden soll. Wenn diese Eingabe für einen Schritt definiert ist, hat sie Vorrang vor einem `exit`-Wert, der `onFailure`- oder `onSuccess`-Eingängen definiert ist. Damit ein Schritt mit dieser Eingabe erwartungsgemäß ausgeführt wird, muss der Schritt der letzte sein, der in den `mainSteps` Ihres Dokuments ausgeführt wird.

Typ: Boolesch

Zulässige Werte: `true` | `false`

Erforderlich: Nein

onFailure

Wenn Sie diese Eingabe für ein Plugin mit dem Wert `exit` angeben und der Schritt fehlschlägt, spiegelt der Schrittstatus den Fehler wider und das Dokument führt keine weiteren Schritte aus,

es sei denn, es wurde ein `finallyStep` definiert. Wenn Sie diese Eingabe für ein Plugin mit dem Wert `successAndExit` angeben und der Schritt fehlschlägt, zeigt der Schrittstatus Erfolg an und das Dokument führt keine weiteren Schritte aus, es sei denn, es wurde ein `finallyStep` definiert.

Typ: Zeichenfolge

Zulässige Werte: `exit` | `successAndExit`

Erforderlich: Nein

onSuccess

Wenn Sie diese Eingabe für ein Plugin angeben und der Schritt erfolgreich ausgeführt wird, führt das Dokument keine weiteren Schritte durch, es sei denn, es wurde ein `finallyStep`definiert.

Typ: Zeichenfolge

Zulässige Werte: `exit`

Erforderlich: Nein

YAML

```
---
schemaVersion: '2.2'
description: Shared inputs example
parameters:
  customDocumentParameter:
    type: String
    description: Example parameter for a custom Command-type document.
mainSteps:
- action: aws:runDocument
  name: runCustomConfiguration
  inputs:
    documentType: SSMDocument
    documentPath: "yourCustomDocument"
    documentParameters: '"documentParameter":{{customDocumentParameter}}'
    onSuccess: exit
- action: aws:runDocument
  name: ifConfigurationFailure
  inputs:
    documentType: SSMDocument
```

```
    documentPath: "yourCustomRepairDocument"
    onFailure: exit
- action: aws:runDocument
  name: finalConfiguration
  inputs:
    documentType: SSMDocument
    documentPath: "yourCustomFinalDocument"
    finallyStep: true
```

JSON

```
{
   "schemaVersion": "2.2",
   "description": "Shared inputs example",
   "parameters": {
      "customDocumentParameter": {
         "type": "String",
         "description": "Example parameter for a custom Command-type document."
      }
   },
   "mainSteps":[
      {
         "action": "aws:runDocument",
         "name": "runCustomConfiguration",
         "inputs": {
            "documentType": "SSMDocument",
            "documentPath": "yourCustomDocument",
            "documentParameters": "\"documentParameter\":
{{customDocumentParameter}}",
            "onSuccess": "exit"
         }
      },
      {
         "action": "aws:runDocument",
         "name": "ifConfigurationFailure",
         "inputs": {
            "documentType": "SSMDocument",
            "documentPath": "yourCustomRepairDocument",
            "onFailure": "exit"
         }
      },
      {
         "action": "aws:runDocument",
```

```
         "name":"finalConfiguration",
         "inputs": {
            "documentType": "SSMDocument",
            "documentPath": "yourCustomFinalDocument",
            "finallyStep": true
         }
      }
   ]
}
```

aws:applications

Installieren, reparieren oder deinstallieren Sie Anwendungen auf einer EC2 Instance. Dieses Plugin läuft nur unter Windows Server-Betriebssystemen.

Syntax

Schema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:applications plugin
parameters:
  source:
    description: "(Required) Source of msi."
    type: String
mainSteps:
- action: aws:applications
  name: example
  inputs:
    action: Install
    source: "{{ source }}"
```

JSON

```
{
  "schemaVersion":"2.2",
  "description":"aws:applications",
  "parameters":{
    "source":{
```

```
    "description":"(Required) Source of msi.",
    "type":"String"
    }
  },
  "mainSteps":[
    {
      "action":"aws:applications",
      "name":"example",
      "inputs":{
        "action":"Install",
        "source":"{{ source }}"
      }
    }
  ]
}
```

Schema 1.2

YAML

```
---
runtimeConfig:
  aws:applications:
    properties:
    - id: 0.aws:applications
      action: "{{ action }}"
      parameters: "{{ parameters }}"
      source: "{{ source }}"
      sourceHash: "{{ sourceHash }}"
```

JSON

```
{
   "runtimeConfig":{
      "aws:applications":{
         "properties":[
            {
               "id":"0.aws:applications",
               "action":"{{ action }}",
               "parameters":"{{ parameters }}",
               "source":"{{ source }}",
               "sourceHash":"{{ sourceHash }}"
```

```
            }
         ]
      }
   }
}
```

Eigenschaften

action

Die zu ergreifende Maßnahme.

Type: Zähler

Zulässige Werte: `Install` | `Repair` | `Uninstall`

Erforderlich: Ja

Parameter

Die Parameter für das Installationsprogramm.

Typ: Zeichenfolge

Erforderlich: Nein

Quelle

Die URL der `.msi`-Datei der Anwendung.

Typ: Zeichenfolge

Erforderlich: Ja

sourceHash

Der SHA256 Hash der `.msi` Datei.

Typ: Zeichenfolge

Erforderlich: Nein

`aws:cloudWatch`

Exportieren Sie Daten aus Windows Server Amazon CloudWatch oder Amazon CloudWatch Logs und überwachen Sie die Daten anhand von CloudWatch Metriken. Dieses Plugin läuft nur unter Windows Server-Betriebssystemen. Weitere Informationen zur Konfiguration der CloudWatch Integration mit Amazon Elastic Compute Cloud (Amazon EC2) finden Sie unter Erfassung von Metriken, Protokollen und Traces mit dem CloudWatch Agenten im CloudWatch Amazon-Benutzerhandbuch.

> **Important**
>
> Der Unified CloudWatch Agent wurde SSM Agent als Tool zum Senden von Protokolldaten an Amazon CloudWatch Logs ersetzt. Das SSM Agent-aws:cloudWatch-Plugin wird nicht unterstützt. Wir empfehlen, nur den Unified CloudWatch Agent für Ihre Protokollerfassungsprozesse zu verwenden. Weitere Informationen finden Sie unter den folgenden Themen:
>
> - Senden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent)
> - Migrieren Sie die Erfassung von Windows Server-Knotenprotokollen auf den CloudWatch Agenten
> - Erfassung von Metriken, Protokollen und Traces mit dem CloudWatch Agenten im CloudWatch Amazon-Benutzerhandbuch.

Sie können die folgenden Datentypen exportieren und überwachen:

ApplicationEventLog

Sendet Daten aus dem Anwendungsereignisprotokoll an CloudWatch Logs.

CustomLogs

Sendet jede textbasierte Protokolldatei an Amazon CloudWatch Logs. Das CloudWatch Plugin erstellt einen Fingerabdruck für Protokolldateien. Anschließend verknüpft das System einen Datenversatz mit jedem Fingerabdruck. Das Plugin lädt Dateien hoch, wenn Änderungen vorliegen, erfasst den Versatz und verknüpft ihn mit einem Fingerabdruck. Diese Methode wird verwendet, um zu verhindern, dass ein Benutzer das Plugin aktiviert, den Service mit einem Verzeichnis verknüpft, in dem sich eine große Anzahl von Dateien befindet, und das System alle Dateien hochlädt.

> **Warning**
>
> Hinweis: Falls Ihre Anwendung während der Abfrage Protokolle kürzt oder zu säubern versucht, besteht die Möglichkeit, dass alle Protokolle, die für `LogDirectoryPath` angegeben wurden, Einträge verlieren. Wenn Sie beispielsweise die Größe der Protokolldatei einschränken möchten, erstellen Sie eine neue Protokolldatei, wenn diese Beschränkung erreicht ist, und lassen Sie neue Daten dann in die neue Datei schreiben.

ETW

Sendet ETW-Daten (Event Tracing for Windows) an CloudWatch Logs.

IIS

Sendet IIS-Protokolldaten an CloudWatch Logs.

PerformanceCounter

Sendet Windows-Leistungsindikatoren an CloudWatch. Sie können verschiedene Kategorien auswählen, in die Daten hochgeladen CloudWatch werden sollen. Erstellen Sie für jeden Leistungsindikator, den Sie hochladen möchten, einen PerformanceCounterAbschnitt mit einer eindeutigen ID (z. B. "PerformanceCounter2"," PerformanceCounter 3" usw.) und konfigurieren Sie dessen Eigenschaften.

> **Note**
>
> Wenn das AWS Systems Manager SSM Agent oder das CloudWatch Plugin gestoppt ist, werden die Leistungsindikatordaten nicht protokolliert CloudWatch. Diese Verhaltensweise unterscheidet sich von der von benutzerdefinierten oder von Windows-Event-Protokollen.. In benutzerdefinierten Protokollen und Windows-Ereignisprotokollen werden die SSM Agent Leistungsindikatordaten gespeichert und CloudWatch nach der Verfügbarkeit des CloudWatch Plug-ins hochgeladen.

SecurityEventLog

Sendet Daten aus dem Sicherheitsereignisprotokoll an CloudWatch Logs.

SystemEventLog

Sendet Daten aus dem Systemereignisprotokoll an CloudWatch Logs.

Sie können die folgenden Ziele für die Daten definieren:

CloudWatch

Das Ziel, an das die Leistungsindikatormetrikdaten gesendet werden. Sie können weitere eindeutige Abschnitte hinzufügen IDs (z. B. "CloudWatch2", CloudWatch 3" usw.) und für jede neue ID eine andere Region angeben, um dieselben Daten an verschiedene Speicherorte zu senden.

CloudWatchLogs

Das Ziel, an das die Protokolldaten gesendet werden. Sie können weitere Abschnitte mit eindeutigen IDs Merkmalen hinzufügen (z. B. "CloudWatchLogs2", CloudWatchLogs 3" usw.) und für jede neue ID eine andere Region angeben, um dieselben Daten an verschiedene Standorte zu senden.

Syntax

```
"runtimeConfig":{
        "aws:cloudWatch":{
            "settings":{
                "startType":"{{ status }}"
            },
            "properties":"{{ properties }}"
        }
    }
```

Einstellungen und Eigenschaften

AccessKey

Ihre -Zugriffsschlüssel-ID Diese Eigenschaft ist erforderlich, wenn Sie die Instance mithilfe einer IAM-Rolle gestartet haben. Diese Eigenschaft kann nicht mit SSM verwendet werden.

Typ: Zeichenfolge

Erforderlich: Nein

CategoryName

Die Leistungsindikatorkategorie von Performance Monitor.

Typ: Zeichenfolge

Erforderlich: Ja

CounterName

Der Name des Leistungsindikators von Performance Monitor.

Typ: Zeichenfolge

Erforderlich: Ja

CultureName

Das Gebietsschema, unter dem der Zeitstempel protokolliert wird. Wenn dieses Feld leer CultureNameist, wird standardmäßig dasselbe Gebietsschema verwendet, das von Ihrer Windows Server Instanz verwendet wird.

Typ: Zeichenfolge

Gültige Werte: Eine Liste der unterstützten Werte finden Sie unter National Language Support (NLS) auf der Microsoft-Website. Die Werte div, div-MV, hu und hu-HU werden nicht unterstützt.

Erforderlich: Nein

DimensionName

Eine Dimension für Ihre CloudWatch Amazon-Metrik. Wenn Sie `DimensionName` angeben, müssen Sie auch `DimensionValue` angeben. Diese Parameter bieten eine andere Ansicht bei der Auflistung von Metriken. Sie können eine Dimension auch für mehrere Metriken verwenden, sodass Sie alle Metriken anzeigen können, die zu einer bestimmten Dimension gehören.

Typ: Zeichenfolge

Erforderlich: Nein

DimensionValue

Ein Dimensionswert für Ihre CloudWatch Amazon-Metrik.

Typ: Zeichenfolge

Erforderlich: Nein

Codierung

Die zu verwendende Dateikodierung (z. B: UTF-8). Verwenden Sie den Kodierungsnamen, nicht den Anzeigenamen.

Typ: Zeichenfolge

Gültige Werte: Eine Liste der unterstützten Werte finden Sie unter Encoding Class in der Microsoft Learn Bibliothek.

Erforderlich: Ja

Filter

Das Präfix des Protokollnamens. Lassen Sie diesen Parameter leer, um alle Dateien zu überwachen.

Typ: Zeichenfolge

Gültige Werte: Eine Liste der unterstützten Werte finden Sie unter „FileSystemWatcherFilter Eigenschaft“ in der MSDN-Bibliothek.

Erforderlich: Nein

Flüsse

Jeder Datentyp, der hochgeladen werden soll, zusammen mit dem Ziel für die Daten (CloudWatch oder CloudWatch Protokolle). Um beispielsweise einen unter definierten Leistungsindikator an das unter definierte CloudWatch Ziel `"Id": "PerformanceCounter"` zu senden`"Id": "CloudWatch"`, geben Sie "PerformanceCounter,CloudWatch" ein. Geben Sie auf ähnliche Weise „(ETW)," ein, um das benutzerdefinierte Protokoll**`"Id": "ETW"`**, das ETW-Protokoll und das Systemprotokoll an das unter definierte CloudWatch Protokollziel zu senden. CloudWatchLogs Außerdem können Sie dieselbe Leistungsindikator- oder Protokolldatei an mehrere Ziele senden. Um beispielsweise das Anwendungsprotokoll an zwei verschiedene Ziele zu senden, die Sie unter `"Id": "CloudWatchLogs"` und definiert haben`"Id": "CloudWatchLogs2"`, geben Sie ", (ApplicationEventLogCloudWatchLogs, CloudWatchLogs 2)“ ein.

Typ: Zeichenfolge

Gültige Werte (Quelle): `ApplicationEventLog` | `CustomLogs` | `ETW` | `PerformanceCounter` | `SystemEventLog` | `SecurityEventLog`

Gültige Werte (Ziel): `CloudWatch` | `CloudWatchLogs` | `CloudWatch` *n* | `CloudWatchLogs` *n*

Erforderlich: Ja

FullName

Der vollständige Name der Komponente.

Typ: Zeichenfolge

Erforderlich: Ja

Id

Identifiziert die Datenquelle bzw. das Ziel. Die ID muss innerhalb der Konfigurationsdatei eindeutig sein.

Typ: Zeichenfolge

Erforderlich: Ja

InstanceName

Der Name der Leistungsindikator-Instance. Verwenden Sie kein Sternchen (*) für alle Instances, da jede Leistungszählerkomponente nur eine Metrik unterstützt. Sie können jedoch _Total verwenden.

Typ: Zeichenfolge

Erforderlich: Ja

Levels

Die Arten von Nachrichten, die an Amazon gesendet werden sollen CloudWatch.

Typ: Zeichenfolge

Zulässige Werte:

- 1 – Nur Fehlermeldungen werden hochgeladen.
- 2 – Nur Warnmeldungen werden hochgeladen.
- 4 – Nur Informationsmeldungen werden hochgeladen.

Sie können Werte kombinieren, um mehr als einen Meldungstyp einzuschließen. Beispiel: 3 bedeutet, dass Fehlermeldungen (1) und Warnmeldungen (2) enthalten sind. Wenn Sie den Wert

7 eingeben, werden Fehlermeldungen (1), Warnmeldungen (2) und Informationsmeldungen (4) einbezogen.

Erforderlich: Ja

> **Note**
>
> Windows-Sicherheitsprotokolle müssen für „Levels“ den Wert „7“ festlegen.

LineCount

Die Anzahl der Zeilen im Header zur Identifikation der Protokolldatei. Beispielsweise haben IIS-Protokolldateien praktisch identische Header. Sie können 3 eingeben; dann würden die ersten drei Zeilen des Headers der Protokolldatei gelesen, um diese zu identifizieren. In IIS-Protokolldateien ist die dritte Zeile Datum und Zeitstempel, die sich zwischen Protokolldateien unterscheiden.

Typ: Ganzzahl

Erforderlich: Nein

LogDirectoryPath

Für den Pfad CustomLogs, in dem Protokolle auf Ihrer EC2 Instance gespeichert werden. Bei IIS-Protokollen der Ordner, in dem IIS-Protokolle für eine einzelne Site gespeichert werden (z. B. C:\\ inetpub\\ logs\\\LogFiles\ ***n*** W3SVC). Hinsichtlich IIS-Protokolle wird nur das Protokollformat W3C unterstützt. IIS, NCSA und benutzerdefinierte Formate werden nicht unterstützt.

Typ: Zeichenfolge

Erforderlich: Ja

LogGroup

Der Name für Ihre Protokollgruppe. Dieser Name wird auf dem Bildschirm Log Groups (Protokollgruppen) in der CloudWatch-Konsole angezeigt.

Typ: Zeichenfolge

Erforderlich: Ja

LogName

Der Name der Protokolldateien.

1. Zum Suchen des Protokollnamens klicken Sie in der Ereignisanzeige im Navigationsbereich auf Applications and Services Logs.
2. Klicken Sie in der Liste der Protokolle mit der rechten Maustaste auf das Protokoll, das Sie hochladen möchten (z. B. `Microsoft` > `Windows` > `Backup` > `Operational`), und klicken Sie dann auf Benutzerdefinierte Ansicht erstellen.
3. Klicken Sie im Dialogfeld Create Custom View auf die Registerkarte XML. Das LogNamebefindet sich im <Select Path=> -Tag (zum Beispiel). `Microsoft-Windows-Backup` Kopieren Sie diesen Text in den LogNameParameter.

Typ: Zeichenfolge

Zulässige Werte: `Application` | `Security` | `System` | `Microsoft-Windows-WinINet/Analytic`

Erforderlich: Ja

LogStream

Der Zielprotokollstream. Wenn Sie {instance_id}, verwenden, also den Standard, wird die Instance-ID dieser Instance als Name des Protokollstreams verwendet.

Typ: Zeichenfolge

Gültige Werte: `{instance_id}` | `{hostname}` | `{ip_address}` *`<log_stream_name>`*

Wenn Sie einen Log-Stream-Namen eingeben, der noch nicht existiert, erstellt CloudWatch Logs ihn automatisch für Sie. Sie können den Protokollstream mit einer Literalzeichenfolge, einer vordefinierten Variablen ({instance_id}, {hostname}, {ip_address}) oder einer Kombination aus allen drei Variablen definieren.

Der in diesem Parameter angegebene Protokollstreamname wird auf dem *`<YourLogStream>`* Bildschirm Protokollgruppen > Streams für in der CloudWatch Konsole angezeigt.

Erforderlich: Ja

MetricName

Die CloudWatch Metrik, unter der die Leistungsdaten enthalten sein sollen.

> **Note**
>
> Verwenden Sie keine Sonderzeichen in dem Namen. Andernfalls funktionieren die Metrik und die zugehörigen Alarme möglicherweise nicht.

Typ: Zeichenfolge

Erforderlich: Ja

NameSpace

Der Metrik-Namespace, in dem die Leistungsindikatordaten geschrieben werden sollen.

Typ: Zeichenfolge

Erforderlich: Ja

PollInterval

Die Anzahl der Sekunden, die vergehen muss, bevor neue Leistungsindikator- und Protokolldaten hochgeladen werden.

Typ: Ganzzahl

Gültige Werte: Legen Sie für diesen Wert 5 oder mehr Sekunden fest. Fünfzehn Sekunden (00:00:15) sind empfohlen.

Erforderlich: Ja

Region

Der AWS-Region Ort, an den Sie Protokolldaten senden möchten. Obwohl Sie Leistungszähler an eine andere Region als die, an die Sie Ihre Protokolldaten senden, senden können, empfehlen wir, diesen Parameter auf dieselbe Region zu setzen, in der Ihre Instance ausgeführt wird.

Typ: Zeichenfolge

Gültige Werte: Regionen IDs , die sowohl von Systems Manager als auch von CloudWatch Logs AWS-Regionen unterstützt werden`us-east-2`, wie`eu-west-1`, und`ap-southeast-1`. Eine Liste der von den einzelnen Services AWS-Regionen unterstützten Services finden Sie unter Amazon CloudWatch Logs Service Endpoints und Systems Manager Service Endpoints in der. Allgemeine Amazon Web Services-Referenz

Erforderlich: Ja

SecretKey

Ihr geheimer -Zugriffsschlüssel Diese Eigenschaft ist erforderlich, wenn Sie die Instance mithilfe einer IAM-Rolle gestartet haben.

Typ: Zeichenfolge

Erforderlich: Nein

startType

Schalten Sie die Instance ein oder aus CloudWatch .

Typ: Zeichenfolge

Zulässige Werte: `Enabled` | `Disabled`

Erforderlich: Ja

TimestampFormat

Das Zeitstempelformat, das Sie verwenden möchten. Eine Liste der unterstützten Werte finden Sie unter Custom Date and Time Format Strings in der MSDN-Bibliothek.

Typ: Zeichenfolge

Erforderlich: Ja

TimeZoneKind

Stellt Zeitzoneninformationen bereit, wenn der Zeitstempel Ihres Protokolls keine Zeitzoneninformationen enthält. Wenn dieser Parameter leer gelassen wird und Ihr Zeitstempel keine Zeitzoneninformationen enthält, verwendet CloudWatch Logs standardmäßig die lokale Zeitzone. Dieser Parameter wird ignoriert, wenn der Zeitstempel bereits Zeitzoneninformationen enthält.

Typ: Zeichenfolge

Zulässige Werte: `Local` | `UTC`

Erforderlich: Nein

Einheit

Die korrekte Maßeinheit für die Metrik.

Typ: Zeichenfolge

Gültige Werte: Sekunden | Mikrosekunden | Millisekunden | Byte | Kilobyte | Megabyte | Gigabyte | Terabyte | Bits | Kilobits | Kilobits | Megabit | Gigabit | Terabit | Prozent | Anzahl | | |

```
  "parameters": {
    "action": {
      "description": "(Required) The type of action to perform.",
      "type": "String",
      "default": "Install",
      "allowedValues": [
        "Install",
        "Uninstall"
      ]
    }
  },
  "mainSteps": [
    {
      "action": "aws:configureDocker",
      "name": "configureDocker",
      "inputs": {
        "action": "{{ action }}"
      }
    }
  ]
}
```

Eingaben

action

Der Typ der Aktion, die durchgeführt werden soll.

Type: Zähler

Zulässige Werte: `Install` | `Uninstall`

Erforderlich: Ja

`aws:configurePackage`

(Schemaversion AWS Systems Manager Distributor 2.0 oder höher) Installieren oder deinstallieren Sie ein Paket. Sie können die neueste Version, die Standardversion oder eine Version des angegebenen Pakets installieren. Pakete, die von bereitgestellt AWS werden, werden ebenfalls unterstützt. Dieses Plugin läuft unter Windows Server- und Linux-Betriebssystemen, wobei jedoch in Linux-Betriebssystemen nicht alle verfügbaren Pakete unterstützt werden.

Zu den verfügbaren AWS Paketen für Windows Server gehören: `AWSPVDriverAWSNVMe`,`AwsEnaNetworkDriver`,`AwsVssComponents`,`AmazonCloudWatchAgent`,`CodeD` und `AWSSupport-EC2Rescue`.

Zu den verfügbaren AWS Paketen für Linux-Betriebssysteme gehören die folgenden: `AmazonCloudWatchAgentCodeDeployAgent`, und`AWSSupport-EC2Rescue`.

Syntax

Schema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:configurePackage
parameters:
  name:
    description: "(Required) The name of the AWS package to install or uninstall."
    type: String
  action:
    description: "(Required) The type of action to perform."
    type: String
    default: Install
    allowedValues:
    - Install
    - Uninstall
  ssmParameter:
    description: "(Required) Argument stored in Parameter Store."
    type: String
    default: "{{ ssm:parameter_store_arg }}"
mainSteps:
- action: aws:configurePackage
  name: configurePackage
  inputs:
    name: "{{ name }}"
    action: "{{ action }}"
    additionalArguments:
      "{\"SSM_parameter_store_arg\": \"{{ ssmParameter }}\", \"SSM_custom_arg\":
 \"myValue\"}"
```

JSON

```
{
   "schemaVersion": "2.2",
   "description": "aws:configurePackage",
   "parameters": {
      "name": {
         "description": "(Required) The name of the AWS package to install or
 uninstall.",
         "type": "String"
      },
      "action": {
         "description": "(Required) The type of action to perform.",
         "type": "String",
         "default": "Install",
         "allowedValues": [
            "Install",
            "Uninstall"
         ]
      },
      "ssmParameter": {
         "description": "(Required) Argument stored in Parameter Store.",
         "type": "String",
         "default": "{{ ssm:parameter_store_arg }}"
      }
   },
   "mainSteps": [
      {
         "action": "aws:configurePackage",
         "name": "configurePackage",
         "inputs": {
            "name": "{{ name }}",
            "action": "{{ action }}",
            "additionalArguments": "{\"SSM_parameter_store_arg\":
 \"{{ ssmParameter }}\", \"SSM_custom_arg\": \"myValue\"}"
         }
      }
   ]
}
```

Eingaben

name

Der Name des zu installierenden oder zu deinstallierenden AWS Pakets. In verfügbaren Paketen ist Folgendes enthalten: `AWSPVDriver`, `AwsEnaNetworkDriver`, `AwsVssComponents` und `AmazonCloudWatchAgent`.

Typ: Zeichenfolge

Erforderlich: Ja

action

Installieren oder deinstallieren Sie ein Paket.

Type: Zähler

Zulässige Werte: `Install` | `Uninstall`

Erforderlich: Ja

installationType

Der Typ der auszuführenden Installation. Wenn Sie `Uninstall and reinstall` angeben, wird das Paket vollständig deinstalliert und anschließend neu installiert. Die Anwendung ist bis zum Abschluss der Neuinstallation nicht verfügbar. Wenn Sie `In-place update` angeben, werden der vorhandenen Installation nur neue oder geänderte Dateien hinzugefügt, entsprechend den Anweisungen, die Sie in einem Update-Skript bereitstellen. Die Anwendung ist während des Aktualisierungsprozesses weiterhin verfügbar. Die `In-place update` Option wird für Pakete mit dem Namen AWS-published nicht unterstützt. `Uninstall and reinstall`ist der Standardwert.

Type: Zähler

Zulässige Werte: `Uninstall and reinstall` | `In-place update`

Erforderlich: Nein

additionalArguments

Eine JSON-Zeichenkette mit den zusätzlichen Parametern, die Sie Ihren Installations-, Deinstallations- oder Update-Skripten bereitstellen müssen. Jedem Parameter muss das Präfix `SSM_` angefügt werden. Sie können in Ihren zusätzlichen Argumenten auf einen Parameter Store-

Parameter verweisen, indem Sie die Konvention `{{ssm:parameter-name}}` verwenden. Um den zusätzlichen Parameter in Ihrem Installations-, Deinstallations- oder Updateskript zu verwenden, müssen Sie den Parameter mithilfe der für das Betriebssystem geeigneten Syntax als Umgebungsvariable referenzieren. In verweisen Sie PowerShell beispielsweise auf das `SSM_arg` Argument als`$Env:SSM_arg`. Es gibt keine Begrenzung für die Anzahl der von Ihnen definierten Argumente, aber die Eingabe von zusätzlichen Argumenten hat eine Begrenzung von 4096 Zeichen. Dieser Grenzwert umfasst alle von Ihnen definierten Schlüssel und Werte.

Typ: StringMap

Erforderlich: Nein

version

Installieren oder deinstallieren Sie eine bestimmte Version des Pakets. Wenn Sie ein Installation vornehmen, installiert das System standardmäßig die neueste veröffentlichte Version. Wenn Sie eine Deinstallation vornehmen, deinstalliert das System standardmäßig die derzeit installierte Version. Wenn keine installierte Version gefunden wird, wird die neueste veröffentlichte Version heruntergeladen und die Deinstallationsaktion ausgeführt.

Typ: Zeichenfolge

Erforderlich: Nein

`aws:domainJoin`

Verknüpfen Sie eine EC2 Instanz mit einer Domain. Dieses Plugin läuft unter Linux- und Windows Server-Betriebssystemen. Dieses Plugin ändert den Hostnamen für Linux-Instances in das Format EC2 AMAZ-. *XXXXXXX* Weitere Informationen zum Beitreten von EC2 Instanzen finden Sie unter Join an EC2 Instance to Your AWS Managed Microsoft AD Directory im AWS Directory Service Administratorhandbuch.

Syntax

Schema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:domainJoin
```

```
parameters:
  directoryId:
    description: "(Required) The ID of the directory."
    type: String
  directoryName:
    description: "(Required) The name of the domain."
    type: String
  directoryOU:
    description: "(Optional) The organizational unit to assign the computer object
 to."
    type: String
  dnsIpAddresses:
    description: "(Required) The IP addresses of the DNS servers for your
 directory."
    type: StringList
  hostname:
    description: "(Optional) The hostname you want to assign to the node."
    type: String
mainSteps:
- action: aws:domainJoin
  name: domainJoin
  inputs:
    directoryId: "{{ directoryId }}"
    directoryName: "{{ directoryName }}"
    directoryOU: "{{ directoryOU }}"
    dnsIpAddresses: "{{ dnsIpAddresses }}"
    hostname: "{{ hostname }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:domainJoin",
  "parameters": {
    "directoryId": {
      "description": "(Required) The ID of the directory.",
      "type": "String"
    },
    "directoryName": {
      "description": "(Required) The name of the domain.",
      "type": "String"
    },
    "directoryOU": {
```

```
        "description": "(Optional) The organizational unit to assign the computer
 object to.",
        "type": "String"
      },
    "dnsIpAddresses": {
      "description": "(Required) The IP addresses of the DNS servers for your
 directory.",
      "type": "StringList"
    },
    "hostname": {
        "description": "(Optional) The hostname you want to assign to the node.",
        "type": "String"
      }
  },
  "mainSteps": [
    {
      "action": "aws:domainJoin",
      "name": "domainJoin",
      "inputs": {
        "directoryId": "{{ directoryId }}",
        "directoryName": "{{ directoryName }}",
        "directoryOU":"{{ directoryOU }}",
        "dnsIpAddresses":"{{ dnsIpAddresses }}",
        "hostname":"{{ hostname }}"
      }
    }
  ]
}
```

Schema 1.2

YAML

```
---
runtimeConfig:
  aws:domainJoin:
    properties:
      directoryId: "{{ directoryId }}"
      directoryName: "{{ directoryName }}"
      directoryOU: "{{ directoryOU }}"
      dnsIpAddresses: "{{ dnsIpAddresses }}"
```

JSON

```
{
   "runtimeConfig":{
      "aws:domainJoin":{
         "properties":{
            "directoryId":"{{ directoryId }}",
            "directoryName":"{{ directoryName }}",
            "directoryOU":"{{ directoryOU }}",
            "dnsIpAddresses":"{{ dnsIpAddresses }}"
         }
      }
   }
}
```

Eigenschaften

directoryId

Die ID des Verzeichnisses.

Typ: Zeichenfolge

Erforderlich: Ja

Beispiel: "directoryId": "d-1234567890"

directoryName

Der Name der Domain.

Typ: Zeichenfolge

Erforderlich: Ja

Beispiel: "directoryName": "example.com"

directoryOU

Die Organisationseinheit (OU).

Typ: Zeichenfolge

Erforderlich: Nein

Beispiel: "directoryOU": "OU=test,DC=example,DC=com"

dnsIpAddresses

Die IP-Adressen des DNS-Servers.

Typ: StringList

Erforderlich: Ja

Beispiel: "dnsIpAddresses„: [" 198.51.100.1", "198.51.100.2"]

hostname

Der Hostname, den Sie dem Knoten zuweisen möchten. Wenn nicht angegeben, gibt es keine Namensänderung für Windows Server-Instances. Linux-Instances hingegen verwenden das Standardbenennungsmuster. Wenn angegeben, verwenden Windows Server-Instances den entsprechenden Wert. Bei Linux-Instances dient er hingegen als Präfix (sofern `keepHostName` nicht auf „true“ gesetzt ist).

Typ: Zeichenfolge

Erforderlich: Nein

keepHostName

Ermittelt, ob der Hostname für Linux-Instances geändert wird, wenn sie der Domain beitreten. Dies ist ein reiner Linux-Parameter. Standardmäßig (ohne Eingaben für `hostnamehostnameNumAppendDigits`, und mit „false“) werden `keepHostName` Linux-Hosts nach dem Muster AMAZ-XXXXXX umbenannt. EC2 Wenn der Wert auf „true“ gesetzt ist, wird der ursprüngliche Hostname beibehalten und Eingaben von `hostname` und `hostnameNumAppendDigits` werden ignoriert.

Typ: Boolesch

Erforderlich: Nein

hostnameNumAppendZiffern

Definiert die Anzahl der zufälligen numerischen Ziffern, die nach dem Hostnamenwert angehängt werden sollen. Dies ist ein reiner Linux-Parameter, der zusammen mit dem Parameter `hostname` verwendet wird. Wenn `hostname` nicht angegeben wurde, wird er ignoriert.

Typ: Zeichenfolge

Zulässige Werte: 1 bis 5

Erforderlich: Nein

Beispiele

Beispiele finden Sie unter Join an Amazon EC2 Instance to your AWS Managed Microsoft AD im AWS Directory Service Administratorhandbuch.

`aws:downloadContent`

(Schemaversion 2.0 oder höher) Laden Sie SSM-Dokumente und -Skripte von Remote-Standorten herunter. GitHub Enterprise-Repositorys werden nicht unterstützt. Dieses Plugin wird unter Linux- und Windows Server-Betriebssystemen unterstützt.

Syntax

Schema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:downloadContent
parameters:
  sourceType:
    description: "(Required) The download source."
    type: String
  sourceInfo:
    description: "(Required) The information required to retrieve the content from
      the required source."
    type: StringMap
mainSteps:
- action: aws:downloadContent
  name: downloadContent
  inputs:
    sourceType: "{{ sourceType }}"
    sourceInfo: "{{ sourceInfo }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:downloadContent",
```

```
  "parameters": {
    "sourceType": {
    "description": "(Required) The download source.",
    "type": "String"
  },
  "sourceInfo": {
    "description": "(Required) The information required to retrieve the content from
 the required source.",
    "type": "StringMap"
    }
  },
  "mainSteps": [
    {
      "action": "aws:downloadContent",
      "name": "downloadContent",
      "inputs": {
        "sourceType":"{{ sourceType }}",
        "sourceInfo":"{{ sourceInfo }}"
      }
    }
  ]
}
```

Eingaben

sourceType

Die Downloadquelle. Systems Manager unterstützt derzeit die folgenden Quellarten für das Herunterladen von Skripts und SSM documents:-Dokumenten: `GitHub`, `Git`, `HTTP`, `S3` und `SSMDocument`.

Typ: Zeichenfolge

Erforderlich: Ja

sourceInfo

Die erforderlichen Informationen zum Abrufen der Inhalte aus der erforderlichen Quelle.

Typ: StringMap

Erforderlich: Ja

Für sourceType **`GitHub,`** geben Sie Folgendes an:

- owner: Die Eigentümer des Repositorys.
- repository: Der Name des Repositorys.
- path: Der Pfad zu der Datei oder dem Verzeichnis, die bzw. das Sie herunterladen möchten.
- getOptions: Zusätzliche Optionen zum Abrufen von Inhalten aus einem anderen Branch als dem Master-Branch oder aus einem bestimmten Commit im Repository. getOptions kann weggelassen werden, wenn Sie den letzten Commit in der Master-Branch verwenden. Wenn Ihr Repository nach dem 1. Oktober 2020 erstellt wurde, wird der Standardzweig möglicherweise „main“ statt „master“ genannt. In diesem Fall müssen Sie Werte für den getOptions-Parameter angeben.

  Dieser Parameter verwendet das folgende Format:

  - branch:refs/heads/ *branch_name*

    Der Standardwert ist `master`.

    Verwenden Sie das folgende Format, um einen nicht standardmäßigen Zweig anzugeben:

    Branch:refs/heads/ *branch_name*
  - commitID: *commitID*

    Der Standardwert ist `head`.

    Um die Version Ihres SSM-Dokuments in einem anderen als dem letzten Commit zu verwenden, geben Sie die vollständige Commit-ID an. Zum Beispiel:

    ```
    "getOptions": "commitID:bbc1ddb94...b76d3bEXAMPLE",
    ```
- tokenInfo: Der Systems Manager Manager-Parameter (ein SecureString Parameter), in dem Sie Ihre GitHub Zugriffstoken-Informationen speichern, im Format. `{{ssm-secure:`*secure-string-token-name*`}}`

  > **Note**
  >
  > Dieses `tokenInfo` Feld ist das einzige SSM-Dokument-Plugin-Feld, das einen Parameter unterstützt. SecureString SecureString Parameter werden weder für andere Felder noch für andere SSM-Dokument-Plugins unterstützt.

```
{
```

```
    "owner":"TestUser",
    "repository":"GitHubTest",
    "path":"scripts/python/test-script",
    "getOptions":"branch:master",
    "tokenInfo":"{{ssm-secure:secure-string-token}}"
}
```

Für sourceType **`Git`** müssen Sie Folgendes angeben:

- Repository

  Die URL des Git-Repositorys zu der Datei oder dem Verzeichnis, die bzw. das Sie herunterladen möchten.

  Typ: Zeichenfolge

Sie können zusätzlich einen der folgenden optionalen Parameter angeben:

- getOptions

  Zusätzliche Optionen zum Abrufen von Inhalten aus einem anderen Branch als dem Master-Branch oder aus einem bestimmten Commit im Repository. getOptions kann weggelassen werden, wenn Sie den letzten Commit in der Master-Branch verwenden.

  Typ: Zeichenfolge

  Dieser Parameter verwendet das folgende Format:

  - Branch:refs/heads/ *branch_name*

    Der Standardwert ist `master`.

    `"branch"` ist nur erforderlich, wenn Ihr SSM-Dokument in einer anderen Verzweigung als `master` gespeichert ist. Zum Beispiel:

    ```
    "getOptions": "branch:refs/heads/main"
    ```

  - commitID: *commitID*

    Der Standardwert ist `head`.

    Um die Version Ihres SSM-Dokuments in einem anderen als dem letzten Commit zu verwenden, geben Sie die vollständige Commit-ID an. Zum Beispiel:

```
"getOptions": "commitID:bbc1ddb94...b76d3bEXAMPLE",
```

- privat SSHKey

  Der SSH-Schlüssel, der beim Herstellen einer Verbindung zur `repository` verwendet werden soll. Sie können das folgende Format verwenden, um auf einen `SecureString`-Parameter für den Wert Ihres SSH-Schlüssels zu verweisen: `{{ssm-secure:`*`your-secure-string-parameter`*`}}`.

  Typ: Zeichenfolge

- skipHostKeyÜberprüfend

  Bestimmt den Wert der StrictHostKeyChecking Option, wenn eine Verbindung zu dem von `repository` Ihnen angegebenen hergestellt wird. Der Standardwert ist `false`.

  Typ: Boolesch

- username

  Der Benutzername, der bei der Verbindung mit der `repository` verwendet werden soll, die Sie mit HTTP angeben. Sie können das folgende Format verwenden, um auf einen `SecureString`-Parameter für den Wert Ihres Benutzernamens zu verweisen: `{{ssm-secure:`*`your-secure-string-parameter`*`}}`.

  Typ: Zeichenfolge

- password

  Das Passwort, das bei der Verbindung mit der `repository` verwendet werden soll, die Sie mit HTTP angeben. Sie können das folgende Format verwenden, um auf einen `SecureString`-Parameter für den Wert Ihres Passworts zu verweisen: `{{ssm-secure:`*`your-secure-string-parameter`*`}}`.

  Typ: Zeichenfolge

Für sourceType **`HTTP`** müssen Sie Folgendes angeben:

- URL

  Die URL zu der Datei oder dem Verzeichnis, die bzw. das Sie herunterladen möchten.

  Typ: Zeichenfolge

Sie können zusätzlich einen der folgenden optionalen Parameter angeben:

- allowInsecureDownload

  Bestimmt, ob ein Download über eine Verbindung durchgeführt werden kann, die nicht mit Secure Socket Layer (SSL) oder Transport Layer Security (TLS) verschlüsselt ist. Der Standardwert ist `false`. Wir raten davon ab, Downloads ohne Verschlüsselung durchzuführen. Wenn Sie sich dafür entscheiden, übernehmen Sie alle damit verbundenen Risiken. Sicherheit ist eine gemeinsame Verantwortung zwischen Ihnen AWS und Ihnen. Dies wird als Modell der geteilten Verantwortung beschrieben. Weitere Informationen hierzu finden Sie in Modell der geteilten Verantwortung.

  Typ: Boolesch

- authMethod

  Bestimmt, ob ein Benutzername und ein Passwort für die Authentifizierung verwendet werden, wenn eine Verbindung mit der `url` hergestellt wird, die Sie angeben. Wenn Sie `Basic` oder `Digest` angeben, müssen Sie Werte für die `username`- und `password`-Parameter bereitstellen. Um die `Digest`-Methode zu verwenden, muss SSM Agent-Version 3.0.1181.0 oder höher muss auf Ihrer Instance installiert sein. Die `Digest` Methode unterstützt MD5 SHA256 Verschlüsselung.

  Typ: Zeichenfolge

  Zulässige Werte: `None` | `Basic` | `Digest`

- username

  Der Benutzername, der bei der Verbindung mit der `url` verwendet werden soll, die Sie mit `Basic`-Authentifizierung angeben. Sie können das folgende Format verwenden, um auf einen `SecureString`-Parameter für den Wert Ihres Benutzernamens zu verweisen: `{{ssm-secure:`*`your-secure-string-parameter`*`}}`.

  Typ: Zeichenfolge

- password

  Das Passwort, das bei der Verbindung mit der `url` verwendet werden soll, die Sie mit `Basic`-Authentifizierung angeben. Sie können das folgende Format verwenden, um auf einen `SecureString`-Parameter für den Wert Ihres Passworts zu verweisen: `{{ssm-secure:`*`your-secure-string-parameter`*`}}`.

Typ: Zeichenfolge

Für sourceType **S3** geben Sie Folgendes an:

- Die URL zu der Datei oder dem Verzeichnis, die bzw. das Sie von Amazon S3 herunterladen möchten.

> **Note**
>
> Beim Herunterladen einer Datei aus einem S3-Bucket werden die .etag-Dateien im Download-Verzeichnis generiert.

```
{
    "path": "https://s3.amazonaws.com/amzn-s3-demo-bucket/powershell/
helloPowershell.ps1"
}
```

Geben Sie für sourceType **SSMDocument** eine der folgenden Optionen an:

- name: Der Name und die Version des Dokuments in folgendem Format: `name:version`. Version ist optional.

```
{
    "name": "Example-RunPowerShellScript:3"
}
```

- name: Der ARN für das Dokument im folgenden Format: `arn:aws:ssm:region:account_id:document/document_name`

```
{
   "name":"arn:aws:ssm:us-east-2:3344556677:document/MySharedDoc"
}
```

destinationPath

Ein optionaler lokaler Pfad auf der Instance, in den die Datei heruntergeladen werden soll. Wenn Sie keinen Pfad angeben, wird der Inhalt in einen Pfad relativ zu Ihrer Befehls-ID heruntergeladen.

Typ: Zeichenfolge

Erforderlich: Nein

aws:psModule

Installieren Sie PowerShell Module auf einer EC2 Amazon-Instance. Dieses Plugin läuft nur unter Windows Server-Betriebssystemen.

Syntax

Schema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:psModule
parameters:
  source:
    description: "(Required) The URL or local path on the instance to the
 application
     .zip file."
    type: String
mainSteps:
- action: aws:psModule
  name: psModule
  inputs:
    source: "{{ source }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:psModule",
  "parameters": {
    "source": {
      "description": "(Required) The URL or local path on the instance to the
 application .zip file.",
      "type": "String"
    }
  },
  "mainSteps": [
    {
      "action": "aws:psModule",
      "name": "psModule",
      "inputs": {
```

```
        "source": "{{ source }}"
      }
    }
  ]
}
```

Schema 1.2

YAML

```
---
runtimeConfig:
  aws:psModule:
    properties:
    - runCommand: "{{ commands }}"
      source: "{{ source }}"
      sourceHash: "{{ sourceHash }}"
      workingDirectory: "{{ workingDirectory }}"
      timeoutSeconds: "{{ executionTimeout }}"
```

JSON

```
{
   "runtimeConfig":{
      "aws:psModule":{
         "properties":[
            {
               "runCommand":"{{ commands }}",
               "source":"{{ source }}",
               "sourceHash":"{{ sourceHash }}",
               "workingDirectory":"{{ workingDirectory }}",
               "timeoutSeconds":"{{ executionTimeout }}"
            }
         ]
      }
   }
}
```

Eigenschaften

runCommand

Der PowerShell Befehl, der nach der Installation des Moduls ausgeführt werden soll.

Typ: StringList

Erforderlich: Nein

Quelle

Die URL bzw. der lokale Pfad auf der Instance zur `.zip`-Datei der Anwendung.

Typ: Zeichenfolge

Erforderlich: Ja

sourceHash

Der SHA256 Hash der `.zip` Datei.

Typ: Zeichenfolge

Erforderlich: Nein

timeoutSeconds

Die Zeit in Sekunden, die ein Befehl in Anspruch nehmen darf, bevor er als fehlgeschlagen betrachtet wird.

Typ: Zeichenfolge

Erforderlich: Nein

workingDirectory

Der Pfad zum Arbeitsverzeichnis auf der Instance.

Typ: Zeichenfolge

Erforderlich: Nein

`aws:refreshAssociation`

(Schemaversion 2.0 oder höher) Aktualisieren (Erzwingen) Sie bei Bedarf eine Zuweisung. Diese Aktion ändert den Systemstatus basierend auf was in der ausgewählten Verknüpfungen bzw. in allen

zielgebundenen Verknüpfungen definiert ist. Dieses Plugin läuft unter Linux- und Microsoft Windows Server-Betriebssystemen.

Syntax

Schema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:refreshAssociation
parameters:
  associationIds:
    description: "(Optional) List of association IDs. If empty, all associations
 bound
      to the specified target are applied."
    type: StringList
mainSteps:
- action: aws:refreshAssociation
  name: refreshAssociation
  inputs:
    associationIds:
    - "{{ associationIds }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:refreshAssociation",
  "parameters": {
    "associationIds": {
      "description": "(Optional) List of association IDs. If empty, all associations
 bound to the specified target are applied.",
      "type": "StringList"
    }
  },
  "mainSteps": [
    {
      "action": "aws:refreshAssociation",
      "name": "refreshAssociation",
      "inputs": {
        "associationIds": [
```

```
          "{{ associationIds }}"
        ]
      }
    }
  ]
}
```

Eingaben

associationIds

Liste der Assoziationen IDs. Wenn das Feld leer ist, werden alle Verknüpfungen mit dem angegebenen Ziel angewendet.

Typ: StringList

Erforderlich: Nein

`aws:runDockerAction`

(Schemaversion 2.0 oder höher) Führen Sie Docker-Aktionen auf Containern aus. Dieses Plugin läuft unter Linux- und Microsoft Windows Server-Betriebssystemen.

Syntax

Schema 2.2

YAML

```
---
mainSteps:
- action: aws:runDockerAction
  name: RunDockerAction
  inputs:
    action: "{{ action }}"
    container: "{{ container }}"
    image: "{{ image }}"
    memory: "{{ memory }}"
    cpuShares: "{{ cpuShares }}"
    volume: "{{ volume }}"
    cmd: "{{ cmd }}"
    env: "{{ env }}"
```

```
    user: "{{ user }}"
    publish: "{{ publish }}"
    workingDirectory: "{{ workingDirectory }}"
    timeoutSeconds: "{{ timeoutSeconds }}"
```

JSON

```
{
   "mainSteps":[
      {
         "action":"aws:runDockerAction",
         "name":"RunDockerAction",
         "inputs":{
            "action":"{{ action }}",
            "container":"{{ container }}",
            "image":"{{ image }}",
            "memory":"{{ memory }}",
            "cpuShares":"{{ cpuShares }}",
            "volume":"{{ volume }}",
            "cmd":"{{ cmd }}",
            "env":"{{ env }}",
            "user":"{{ user }}",
            "publish":"{{ publish }}",
            "workingDirectory": "{{ workingDirectory }}",
            "timeoutSeconds": "{{ timeoutSeconds }}"
         }
      }
   ]
}
```

Eingaben

action

Der Typ der Aktion, die durchgeführt werden soll.

Typ: Zeichenfolge

Erforderlich: Ja

Container

Die Container-ID des Dockers.

Typ: Zeichenfolge

Erforderlich: Nein

Abbild

Der Name des Docker-Image.

Typ: Zeichenfolge

Erforderlich: Nein

cmd

Der Container-Befehl.

Typ: Zeichenfolge

Erforderlich: Nein

memory

Die Grenze des Container-Speichers.

Typ: Zeichenfolge

Erforderlich: Nein

cpuShares

Die Container-CPU-Anteile (relative Gewichtung).

Typ: Zeichenfolge

Erforderlich: Nein

Volume

Die Container-Volume-Mounts.

Typ: StringList

Erforderlich: Nein

env

Die Container-Umgebungsvariablen.

Typ: Zeichenfolge

Erforderlich: Nein

user

Der Container-Benutzername.

Typ: Zeichenfolge

Erforderlich: Nein

publish

Die veröffentlichten Container-Ports.

Typ: Zeichenfolge

Erforderlich: Nein

workingDirectory

Der Pfad zum Arbeitsverzeichnis in Ihrem verwalteten Knoten.

Typ: Zeichenfolge

Erforderlich: Nein

timeoutSeconds

Die Zeit in Sekunden, die ein Befehl in Anspruch nehmen darf, bevor er als fehlgeschlagen betrachtet wird.

Typ: Zeichenfolge

Erforderlich: Nein

`aws:runDocument`

(Schema-Version 2.0 oder höher) Führt SSM-Dokumente aus, die in Systems Manager oder einem lokal freigegebenen Verzeichnis gespeichert sind. Sie können dieses Plugin mit dem Plugin `aws:downloadContent` verwenden, um ein SSM-Dokument von einem Remote-Standort in ein lokal freigegebenes Verzeichnis herunterzuladen, und es dann ausführen. Dieses Plugin wird

unter Linux- und Windows Server-Betriebssystemen unterstützt. Dieses Plugin unterstützt nicht das Ausführen des `AWS-UpdateSSMAgent`-Dokuments oder eines anderen Dokuments, das den `aws:updateSsmAgent`-Plugin verwendet.

Syntax

Schema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:runDocument
parameters:
  documentType:
    description: "(Required) The document type to run."
    type: String
    allowedValues:
    - LocalPath
    - SSMDocument
mainSteps:
- action: aws:runDocument
  name: runDocument
  inputs:
    documentType: "{{ documentType }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:runDocument",
  "parameters": {
    "documentType": {
      "description": "(Required) The document type to run.",
      "type": "String",
      "allowedValues": [
        "LocalPath",
        "SSMDocument"
      ]
    }
  },
  "mainSteps": [
    {
```

```
      "action": "aws:runDocument",
      "name": "runDocument",
      "inputs": {
        "documentType": "{{ documentType }}"
      }
    }
  ]
}
```

Eingaben

documentType

Der auszuführende Dokumenttyp. Sie können lokale Dokumente (`LocalPath`) oder in Systems Manager gespeicherte Dokumente (`SSMDocument`) ausführen.

Typ: Zeichenfolge

Erforderlich: Ja

documentPath

Der Pfad zu dem Dokument. Wenn `documentType` `LocalPath` ist, geben Sie den Pfad des Dokuments im lokal freigegebenen Verzeichnis an. Wenn `documentType` `SSMDocument` ist, geben Sie den Namen des Dokuments an.

Typ: Zeichenfolge

Erforderlich: Nein

documentParameters

Parameter für das Dokument.

Typ: StringMap

Erforderlich: Nein

`aws:runPowerShellScript`

Führen Sie PowerShell Skripts aus oder geben Sie den Pfad zu einem auszuführenden Skript an. Dieses Plugin läuft unter Microsoft Windows Server- und Linux-Betriebssystemen.

Syntax

Schema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:runPowerShellScript
parameters:
  commands:
    type: String
    description: "(Required) The commands to run or the path to an existing script
      on the instance."
    default: Write-Host "Hello World"
mainSteps:
- action: aws:runPowerShellScript
  name: runPowerShellScript
  inputs:
    timeoutSeconds: '60'
    runCommand:
    - "{{ commands }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:runPowerShellScript",
  "parameters": {
    "commands": {
      "type": "String",
      "description": "(Required) The commands to run or the path to an existing
 script on the instance.",
      "default": "Write-Host \"Hello World\""
    }
  },
  "mainSteps": [
    {
      "action": "aws:runPowerShellScript",
      "name": "runPowerShellScript",
      "inputs": {
        "timeoutSeconds": "60",
        "runCommand": [
```

```
          "{{ commands }}"
        ]
      }
    }
  ]
}
```

Schema 1.2

YAML

```
---
runtimeConfig:
  aws:runPowerShellScript:
    properties:
    - id: 0.aws:runPowerShellScript
      runCommand: "{{ commands }}"
      workingDirectory: "{{ workingDirectory }}"
      timeoutSeconds: "{{ executionTimeout }}"
```

JSON

```
{
   "runtimeConfig":{
      "aws:runPowerShellScript":{
         "properties":[
            {
               "id":"0.aws:runPowerShellScript",
               "runCommand":"{{ commands }}",
               "workingDirectory":"{{ workingDirectory }}",
               "timeoutSeconds":"{{ executionTimeout }}"
            }
         ]
      }
   }
}
```

Eigenschaften

runCommand

Geben Sie die auszuführenden Befehle oder den Pfad zu einem vorhandenen Skript auf der Instance an.

Typ: StringList

Erforderlich: Ja

timeoutSeconds

Die Zeit in Sekunden, die ein Befehl in Anspruch nehmen darf, bevor er als fehlgeschlagen betrachtet wird. Wenn der Wert für den Timeout erreicht ist, hält Systems Manager die Ausführung des Befehls an.

Typ: Zeichenfolge

Erforderlich: Nein

workingDirectory

Der Pfad zum Arbeitsverzeichnis auf der Instance.

Typ: Zeichenfolge

Erforderlich: Nein

`aws:runShellScript`

Führen Sie Linux-Shell-Skripts aus oder geben Sie den Pfad zu einem auszuführenden Skript an. Dieses Plugin läuft nur unter Linux-Betriebssystemen.

Syntax

Schema 2.2

YAML

```
---
schemaVersion: '2.2'
```

```
description: aws:runShellScript
parameters:
  commands:
    type: String
    description: "(Required) The commands to run or the path to an existing script
      on the instance."
    default: echo Hello World
mainSteps:
- action: aws:runShellScript
  name: runShellScript
  inputs:
    timeoutSeconds: '60'
    runCommand:
    - "{{ commands }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:runShellScript",
  "parameters": {
    "commands": {
      "type": "String",
      "description": "(Required) The commands to run or the path to an existing
 script on the instance.",
      "default": "echo Hello World"
    }
  },
  "mainSteps": [
    {
      "action": "aws:runShellScript",
      "name": "runShellScript",
      "inputs": {
        "timeoutSeconds": "60",
        "runCommand": [
          "{{ commands }}"
        ]
      }
    }
  ]
}
```

Schema 1.2

YAML

```
---
runtimeConfig:
  aws:runShellScript:
    properties:
    - runCommand: "{{ commands }}"
      workingDirectory: "{{ workingDirectory }}"
      timeoutSeconds: "{{ executionTimeout }}"
```

JSON

```
{
   "runtimeConfig":{
      "aws:runShellScript":{
         "properties":[
            {
               "runCommand":"{{ commands }}",
               "workingDirectory":"{{ workingDirectory }}",
               "timeoutSeconds":"{{ executionTimeout }}"
            }
         ]
      }
   }
}
```

Eigenschaften

runCommand

Geben Sie die auszuführenden Befehle oder den Pfad zu einem vorhandenen Skript auf der Instance an.

Typ: StringList

Erforderlich: Ja

timeoutSeconds

Die Zeit in Sekunden, die ein Befehl in Anspruch nehmen darf, bevor er als fehlgeschlagen betrachtet wird. Wenn der Wert für den Timeout erreicht ist, hält Systems Manager die Ausführung des Befehls an.

Typ: Zeichenfolge

Erforderlich: Nein

workingDirectory

Der Pfad zum Arbeitsverzeichnis auf der Instance.

Typ: Zeichenfolge

Erforderlich: Nein

`aws:softwareInventory`

(Schema-Version 2.0 oder höher) Erfassen von Metadaten zu Anwendungen, Dateien und Konfigurationen auf Ihren verwalteten Instances. Dieses Plugin läuft unter Linux- und Microsoft Windows Server-Betriebssystemen. Wenn Sie die Inventarerfassung konfigurieren, erstellen Sie zunächst eine AWS Systems Manager State Manager Zuordnung. Systems Manager erfasst die Bestandsdaten, wenn der Zuordnungsstatus ausgeführt wird. Wenn Sie den Zuordnungsstatus nicht zuerst erstellen und versuchen, das `aws:softwareInventory`-Plugin aufzurufen, gibt das System den folgenden Fehler aus:

```
The aws:softwareInventory plugin can only be invoked via ssm-associate.
```

Pro Instance kann nur jeweils ein Bestandszuordnungsstatus konfiguriert werden. Wenn Sie eine Instance mit zwei oder mehr Zuordnungen konfigurieren, wird der Bestandszuordnungsstatus nicht ausgeführt und es werden keine Bestandsdaten erfasst. Weitere Informationen über das Erfassen des Bestands finden Sie unter AWS Systems Manager-Bestand.

Syntax

Schema 2.2

YAML

```
---
```

```
mainSteps:
- action: aws:softwareInventory
  name: collectSoftwareInventoryItems
  inputs:
    applications: "{{ applications }}"
    awsComponents: "{{ awsComponents }}"
    networkConfig: "{{ networkConfig }}"
    files: "{{ files }}"
    services: "{{ services }}"
    windowsRoles: "{{ windowsRoles }}"
    windowsRegistry: "{{ windowsRegistry}}"
    windowsUpdates: "{{ windowsUpdates }}"
    instanceDetailedInformation: "{{ instanceDetailedInformation }}"
    customInventory: "{{ customInventory }}"
```

JSON

```
{
   "mainSteps":[
      {
         "action":"aws:softwareInventory",
         "name":"collectSoftwareInventoryItems",
         "inputs":{
            "applications":"{{ applications }}",
            "awsComponents":"{{ awsComponents }}",
            "networkConfig":"{{ networkConfig }}",
            "files":"{{ files }}",
            "services":"{{ services }}",
            "windowsRoles":"{{ windowsRoles }}",
            "windowsRegistry":"{{ windowsRegistry}}",
            "windowsUpdates":"{{ windowsUpdates }}",
            "instanceDetailedInformation":"{{ instanceDetailedInformation }}",
            "customInventory":"{{ customInventory }}"
         }
      }
   ]
}
```

Eingaben

applications

(Optional) Erfassen von Metadaten für installierte Anwendungen.

Typ: Zeichenfolge

Erforderlich: Nein

awsComponents

(Optional) Sammeln Sie Metadaten für AWS Komponenten wie amazon-ssm-agent.

Typ: Zeichenfolge

Erforderlich: Nein

files

(Optional, erfordert SSM Agent-Version 2.2.64.0 oder höher) Erfassen von Metadaten für Dateien, einschließlich Dateinamen, der Erstellungszeit der Dateien, der letzten Änderungs- und Zugriffszeit der Dateien oder Dateigrößen usw. Weitere Informationen zum Erfassen eines Dateibestands finden Sie unter Arbeiten mit Datei- und Windows-Registrierungsbestand.

Typ: Zeichenfolge

Erforderlich: Nein

networkConfig

(Optional) Erfassen von Metadaten für Netzwerkkonfigurationen.

Typ: Zeichenfolge

Erforderlich: Nein

billingInfo

(Optional) Sammeln von Metadaten für Plattformdetails, die mit dem Fakturierungscode des AMI verknüpft sind.

Typ: Zeichenfolge

Erforderlich: Nein

windowsUpdates

(Optional) Erfassen von Metadaten für alle Windows-Updates.

Typ: Zeichenfolge

Erforderlich: Nein

instanceDetailedInformation

(Optional) Erfassen weiterer Instance-Informationen neben den Informationen des Standardbestands-Plugins (`aws:instanceInformation`), einschließlich CPU-Modell, Geschwindigkeit und Anzahl der Kerne usw.

Typ: Zeichenfolge

Erforderlich: Nein

service

(Optional, nur Windows-BS, erfordert SSM Agent-Version 2.2.64.0 oder höher) Erfassen von Daten für Servicekonfigurationen.

Typ: Zeichenfolge

Erforderlich: Nein

windowsRegistry

(Optional, nur Windows-BS, erfordert SSM Agent-Version 2.2.64.0 oder höher) Erfassen von Windows Registry-Schlüsseln und -Werten. Sie können einen Schlüssel-Pfad auswählen und alle Schlüssel und Werte rekursiv erfassen. Sie können auch einen bestimmten Registrierungsschlüssel und seinen Wert für einen bestimmten Pfad erfassen. Inventory erfasst den Schlüsselpfad, den Namen, Typ und Wert. Weitere Informationen zur Erfassung von Windows Registry-Bestand finden Sie unter Arbeiten mit Datei- und Windows-Registrierungsbestand.

Typ: Zeichenfolge

Erforderlich: Nein

windowsRoles

(Optional, nur Windows-BS, erfordert SSM Agent-Version 2.2.64.0 oder höher) Erfassen von Metadaten für Microsoft Windows-Rollenkonfigurationen.

Typ: Zeichenfolge

Erforderlich: Nein

customInventory

(Optional) Erfassen von benutzerdefinierten Bestandsdaten. Weitere Informationen zum benutzerdefinierten Bestand finden Sie unter Arbeiten mit benutzerdefiniertem Bestand

Typ: Zeichenfolge

Erforderlich: Nein

customInventoryDirectory

(Optional) Erfassen benutzerdefinierter Inventardaten aus dem angegebenen Verzeichnis. Weitere Informationen zum benutzerdefinierten Bestand finden Sie unter Arbeiten mit benutzerdefiniertem Bestand

Typ: Zeichenfolge

Erforderlich: Nein

`aws:updateAgent`

Aktualisieren Sie den EC2 Config-Dienst auf die neueste Version oder geben Sie eine ältere Version an. Dieses Plugin läuft nur unter Microsoft Windows Server-Betriebssystemen. Weitere Informationen zum EC2 Config-Service finden Sie unter Konfiguration einer Windows-Instance mithilfe des EC2 Config-Service (Legacy) im EC2 Amazon-Benutzerhandbuch.

Syntax

Schema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:updateAgent
mainSteps:
- action: aws:updateAgent
  name: updateAgent
```

```
  inputs:
    agentName: Ec2Config
    source: https://s3.{Region}.amazonaws.com/aws-ssm-{Region}/manifest.json
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:updateAgent",
  "mainSteps": [
    {
      "action": "aws:updateAgent",
      "name": "updateAgent",
      "inputs": {
        "agentName": "Ec2Config",
        "source": "https://s3.{Region}.amazonaws.com/aws-ssm-{Region}/manifest.json"
      }
    }
  ]
}
```

Schema 1.2

YAML

```
---
runtimeConfig:
  aws:updateAgent:
    properties:
      agentName: Ec2Config
      source: https://s3.{Region}.amazonaws.com/aws-ssm-{Region}/manifest.json
      allowDowngrade: "{{ allowDowngrade }}"
      targetVersion: "{{ version }}"
```

JSON

```
{
   "runtimeConfig":{
      "aws:updateAgent":{
         "properties":{
            "agentName":"Ec2Config",
```

```
            "source":"https://s3.{Region}.amazonaws.com/aws-ssm-{Region}/
manifest.json",
            "allowDowngrade":"{{ allowDowngrade }}",
            "targetVersion":"{{ version }}"
         }
      }
   }
}
```

Eigenschaften

agentName

EC2Config. Dies ist der Name des Agenten, der den EC2 Config-Dienst ausführt.

Typ: Zeichenfolge

Erforderlich: Ja

allowDowngrade

Erlauben Sie, dass der EC2 Config-Dienst auf eine frühere Version herabgestuft wird. Wenn hierfür „false“ festgelegt ist, kann der Service nur auf neuere Versionen aktualisiert werden (Standard). Wenn hierfür „true“ festgelegt wurde, geben Sie die ältere Version an.

Typ: Boolesch

Erforderlich: Nein

Quelle

Der Speicherort, an den Systems Manager die Version von EC2 Config kopiert, um sie zu installieren. Sie können diesen Speicherort nicht ändern.

Typ: Zeichenfolge

Erforderlich: Ja

targetVersion

Eine bestimmte Version des EC2 Config-Dienstes, die installiert werden soll. Ist hierfür nichts angegeben, wird der Dienst auf die neueste Version aktualisiert.

Typ: Zeichenfolge

Erforderlich: Nein

`aws:updateSsmAgent`

Aktualisieren Sie SSM Agent auf die neueste Version oder geben Sie eine ältere Version an. Dieses Plugin läuft unter Linux- und Windows Server-Betriebssystemen. Weitere Informationen finden Sie unter Arbeiten mit SSM Agent.

Syntax

Schema 2.2

YAML

```
---
schemaVersion: '2.2'
description: aws:updateSsmAgent
parameters:
  allowDowngrade:
    default: 'false'
    description: "(Optional) Allow the Amazon SSM Agent service to be downgraded to
      an earlier version. If set to false, the service can be upgraded to newer
 versions
      only (default). If set to true, specify the earlier version."
    type: String
    allowedValues:
    - 'true'
    - 'false'
mainSteps:
- action: aws:updateSsmAgent
  name: updateSSMAgent
  inputs:
    agentName: amazon-ssm-agent
    source: https://s3.{Region}.amazonaws.com/amazon-ssm-{Region}/ssm-agent-
manifest.json
    allowDowngrade: "{{ allowDowngrade }}"
```

JSON

```
{
  "schemaVersion": "2.2",
  "description": "aws:updateSsmAgent",
```

```
  "parameters": {
    "allowDowngrade": {
      "default": "false",
      "description": "(Required) Allow the Amazon SSM Agent service to be downgraded
 to an earlier version. If set to false, the service can be upgraded to newer
 versions only (default). If set to true, specify the earlier version.",
      "type": "String",
      "allowedValues": [
        "true",
        "false"
      ]
    }
  },
  "mainSteps": [
    {
      "action": "aws:updateSsmAgent",
      "name": "awsupdateSsmAgent",
      "inputs": {
        "agentName": "amazon-ssm-agent",
        "source": "https://s3.{Region}.amazonaws.com/amazon-ssm-{Region}/ssm-agent-
manifest.json",
        "allowDowngrade": "{{ allowDowngrade }}"
      }
    }
  ]
}
```

Schema 1.2

YAML

```
---
runtimeConfig:
  aws:updateSsmAgent:
    properties:
    - agentName: amazon-ssm-agent
      source: https://s3.{Region}.amazonaws.com/aws-ssm-{Region}/manifest.json
      allowDowngrade: "{{ allowDowngrade }}"
```

JSON

```
{
```

```
   "runtimeConfig":{
      "aws:updateSsmAgent":{
         "properties":[
            {
               "agentName":"amazon-ssm-agent",
               "source":"https://s3.{Region}.amazonaws.com/aws-ssm-{Region}/
manifest.json",
               "allowDowngrade":"{{ allowDowngrade }}"
            }
         ]
      }
   }
}
```

Eigenschaften

agentName

amazon-ssm-agent. Dies ist der Name des Systems Manager Manager-Agenten, der Anfragen verarbeitet und Befehle auf der Instanz ausführt.

Typ: Zeichenfolge

Erforderlich: Ja

allowDowngrade

Erlauben Sie, dass SSM Agent auf eine ältere Version zurückgesetzt wird. Wenn hierfür „false“ festgelegt ist, kann der Agent nur auf neuere Versionen aktualisiert werden (Standard). Wenn hierfür „true“ festgelegt wurde, geben Sie die ältere Version an.

Typ: Boolesch

Erforderlich: Ja

Quelle

Der Ort, an den Systems Manager die zu installierende Version von SSM Agent kopiert. Sie können diesen Speicherort nicht ändern.

Typ: Zeichenfolge

Erforderlich: Ja

targetVersion

Eine bestimmte zu installierende Version von SSM Agent. Ist hierfür nichts angegeben, wird der Agent auf die neueste Version aktualisiert.

Typ: Zeichenfolge

Erforderlich: Nein

Erstellen von SSM-Dokumentinhalten

Wenn die AWS Systems Manager öffentlichen Dokumente nicht alle Aktionen ausführen, die Sie für Ihre AWS Ressourcen ausführen möchten, können Sie Ihre eigenen SSM-Dokumente erstellen. Sie können SSM-Dokumente auch über die Konsole klonen. Beim Klonen von Dokumenten werden Inhalte aus einem vorhandenen Dokument in ein neues Dokument kopiert, das Sie ändern können. Beim Erstellen oder Klonen eines Dokuments darf der Inhalt des Dokuments 64 KB nicht überschreiten. Dieses Kontingent beinhaltet auch den zur Laufzeit für Eingabeparameter angegebenen Inhalt. Wenn Sie ein neues `Command`- oder `Policy`-Dokument erstellen, wird empfohlen, Schemaversion 2.2 oder höher zu verwenden, damit Sie die neuesten Features wie Dokumentbearbeitung, automatisches Versioning, Sequenzierung usw. nutzen können.

Schreiben von SSM-Dokumentinhalt

Um eigene SSM-Dokumentinhalte zu erstellen, müssen Sie die verschiedenen Schemas, Features, Plugins und die Syntax für SSM-Dokumente verstehen. Wir empfehlen Ihnen, sich mit den folgenden Ressourcen vertraut zu machen.

- Schreiben Sie Ihre eigenen Dokumente AWS Systems Manager
- Datenelemente und Parameter
- Schemata, Features und Beispiele
- Referenz für Befehlsdokument-Plugins
- Systems Manager Automation Aktionen-Referenz
- Systemvariablen für Automation
- Weitere Runbook-Beispiele
- Arbeiten mit Systems Manager Automation-Runbooks mithilfe des AWS Toolkit for Visual Studio Code

- Visuelle Designerfahrung für Automation-Runbooks
- Verwenden von Skripten in Runbooks

AWS Vordefinierte SSM-Dokumente können einige der von Ihnen benötigten Aktionen ausführen. Sie können diese Dokumente je nach Dokumenttyp mithilfe der Plugins `aws:runDocument`, `aws:runCommand` oder `aws:executeAutomation` in Ihrem benutzerdefinierten SSM-Dokument aufrufen. Sie können Teile dieser Dokumente auch in ein benutzerdefiniertes SSM-Dokument kopieren und den Inhalt entsprechend Ihren Anforderungen bearbeiten.

Tip

Beim Erstellen von SSM-Dokumentinhalten können Sie den Inhalt ändern und das SSM-Dokument während des Tests mehrmals aktualisieren. Mit den folgenden Befehlen wird das SSM-Dokument mit dem neuesten Inhalt aktualisiert und die Standardversion des Dokuments wird auf die neueste Dokumentversion aktualisiert.

Note

Die Linux- und Windows-Befehle nutzen das `jq`-Befehlszeilen-Tool, um die JSON-Antwortdaten zu filtern.

Linux & macOS

```
latestDocVersion=$(aws ssm update-document \
    --content file://path/to/file/documentContent.json \
    --name "ExampleDocument" \
    --document-format JSON \
    --document-version '$LATEST' \
    | jq -r '.DocumentDescription.LatestVersion')

aws ssm update-document-default-version \
    --name "ExampleDocument" \
    --document-version $latestDocVersion
```

Windows

```
latestDocVersion=$(aws ssm update-document ^
    --content file://C:\path\to\file\documentContent.json ^
```

```
    --name "ExampleDocument" ^
    --document-format JSON ^
    --document-version "$LATEST" ^
    | jq -r '.DocumentDescription.LatestVersion')

aws ssm update-document-default-version ^
    --name "ExampleDocument" ^
    --document-version $latestDocVersion
```

PowerShell

```
$content = Get-Content -Path "C:\path\to\file\documentContent.json" | Out-
String
$latestDocVersion = Update-SSMDocument `
    -Content $content `
    -Name "ExampleDocument" `
    -DocumentFormat "JSON" `
    -DocumentVersion '$LATEST' `
    | Select-Object -ExpandProperty LatestVersion

Update-SSMDocumentDefaultVersion `
    -Name "ExampleDocument" `
    -DocumentVersion $latestDocVersion
```

Bewährte Methoden für die Sicherheit für SSM-Dokumente

Beachten Sie beim Erstellen von SSM-Dokumenten die folgenden bewährten Methoden, um das Einschleusen von Befehlen zu verhindern und eine sichere Parameterbearbeitung zu gewährleisten:

- Verwenden Sie die Interpolation von Umgebungsvariablen für Zeichenfolgeparameter, die in Befehlen oder Skripten verwendet werden. Fügen Sie Ihren Zeichenfolgeparametern die Eigenschaft `interpolationType` mit dem Wert `ENV_VAR` hinzu:

```
{
    "command": {
        "type": "String",
        "description": "Command to execute",
        "interpolationType": "ENV_VAR"
    }
}
```

Sie können die Sicherheit Ihrer SSM-Dokumente weiter verbessern, indem Sie angeben, dass doppelte Anführungszeichen in durch Interpolation gelieferten Werten nicht zulässig sind:

```
{
    "command": {
        "type": "String",
        "description": "Command to execute",
        "interpolationType": "ENV_VAR",
            "allowedPattern": "^[^"]*$"
    }
}
```

- Wenn Sie interpretierte Sprachen wie Python, Ruby oder Node.js verwenden, verweisen Sie auf Parameter mit der entsprechenden Umgebungsvariablensyntax:

```
# Python example
import os
command = os.environ['SSM_Message']
```

- Um die Abwärtskompatibilität mit älteren Versionen von SSM Agent zu gewährleisten (vor Version 3.3.2746.0), fügen Sie eine Fallback-Logik für Umgebungsvariablen hinzu:

```
if [ -z "${SSM_command+x}" ]; then
    export SSM_command="{{command}}"
fi
```

- Kombinieren Sie die Interpolation von Umgebungsvariablen mit `allowedPattern`, um eine zusätzliche Eingabevalidierung bereitzustellen. Im folgenden Beispiel verhindert der `allowedPattern`-Wert `^[^"]*$` doppelte Anführungszeichen im Zeichenfolgewert:

```
{
    "command": {
        "type": "String",
        "interpolationType": "ENV_VAR",
        "allowedPattern": "^[a-zA-Z0-9_-]+$"
    }
}
```

- Bevor Sie Ihr SSM-Dokument implementieren, sollten Sie die folgenden Sicherheitsaspekte prüfen:

- Alle Zeichenfolgeparameter, die Benutzereingaben akzeptieren, verwenden gegebenenfalls die Interpolation mit Umgebungsvariablen.
- Die Eingabevalidierung wird nach Möglichkeit mit `allowedPattern` implementiert.
- Das Dokument beinhaltet eine angemessene Fehlerbehandlung für die Parameterverarbeitung.
- Für Umgebungen, die ältere Versionen von SSM Agent verwenden, wird die Abwärtskompatibilität beibehalten.

Informationen zu AWS diensteigenen Ressourcen, auf die Systems Manager zugreift, und zur Konfiguration von Datenperimeterrichtlinien finden Sie unter. Datenperimeter in AWS Systems Manager

Klonen eines SSM-Dokuments

Sie können AWS Systems Manager Dokumente mit der Systems Manager Documents Console klonen, um SSM-Dokumente zu erstellen. Durch das Klonen von SSM-Dokumenten werden Inhalte aus einem vorhandenen Dokument in ein neues Dokument kopiert, das Sie ändern können. Sie können kein Dokument klonen, das größer als 64 KB ist.

Klonen eines SSM-Dokuments

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Geben Sie in das Suchfeld den Namen des Dokuments ein, das Sie klonen möchten.
4. Wählen Sie den Namen des Dokuments aus, das Sie klonen möchten. Wählen Sie anschließend die Option Clone document (Dokument klonen) im Dropdownmenü Aktionen.
5. Ändern Sie das Dokument nach Belieben und wählen Sie dann Create document (Dokument erstellen), um das Dokument zu speichern.

Nachdem Sie den SSM-Dokumentinhalt geschrieben haben, können Sie mithilfe eines der folgenden Methoden ein SSM-Dokument erstellen.

Erstellen eines SSM-Dokuments

- Erstellen von zusammengesetzten Dokumenten

Erstellen von zusammengesetzten Dokumenten

Ein zusammengesetztes Dokument AWS Systems Manager (SSM) ist ein benutzerdefiniertes Dokument, das eine Reihe von Aktionen ausführt, indem es ein oder mehrere sekundäre SSM-Dokumente ausführt. Zusammengesetzte Dokumente fördern Infrastruktur als Code, indem sie Ihnen ermöglichen, einen Standardsatz an SSM-Dokumenten für allgemeine Aufgaben wie das Bootstrapping von Software oder den Domain-Betritt von Instances zu erstellen. Sie können diese Dokumente dann gemeinsam nutzen, um die Wartung von SSM-Dokumenten AWS-Region zu reduzieren und die Konsistenz AWS-Konten zu gewährleisten.

Sie können beispielsweise ein zusammengesetztes Dokument erstellen, das die folgenden Aktionen ausführt:

1. Installiert alle Patches in der Zulassungsliste.
2. Installieren von Antivirensoftware
3. Herunterladen von Skripts von GitHub und Ausführen der Skripts.

In diesem Beispiel umfasst das benutzerdefinierte SSM-Dokument die folgenden Plugins für die Ausführung dieser Aktionen:

1. Das `aws:runDocument`-Plugin zum Ausführen des `AWS-RunPatchBaseline`-Dokuments, das alle aufgeführten Patches installiert.
2. Das Plugin `aws:runDocument` zum Ausführen des Dokuments `AWS-InstallApplication`, das die Antivirensoftware installiert
3. Das Plugin `aws:downloadContent` zum Herunterladen von Skripts von GitHub und Ausführen der Skripts.

Zusammengesetzte und sekundäre Dokumente können in Systems Manager, GitHub (öffentliche und private Repositorys) oder Amazon S3 gespeichert werden. Zusammengesetzte und sekundäre Dokumente lassen sich im JSON- oder YAML-Format erstellen.

> **Note**
>
> Zusammengesetzte Dokumente können maximal drei Dokumente tief ausgeführt werden. Dies bedeutet, dass ein zusammengesetztes Dokument ein untergeordnetes Dokument aufrufen kann, das wiederum ein letztes Dokument aufrufen kann.

Zum Erstellen eines zusammengesetzten Dokuments fügen Sie das `aws:runDocument`-Plugin einem benutzerdefinierten SSM-Dokument hinzu und geben die erforderlichen Eingaben an. Folgendes ist ein Beispiel eines zusammengesetzten Dokuments, das die folgenden Aktionen ausführt:

1. Ausführen des `aws:downloadContent`-Plugins zum Herunterladen eines SSM-Dokuments von einem öffentlichen Repository auf GitHub in ein lokales Verzeichnis mit der Bezeichnung „bootstrap“. Das SSM-Dokument heißt StateManagerBootstrap .yml (ein YAML-Dokument).
2. Führt das `aws:runDocument` Plugin aus, um das .yml-Dokument auszuführen. StateManagerBootstrap Es wurden keine Parameter angegeben.
3. Führt das Plugin `aws:runDocument` aus, um das `AWS-ConfigureDocker pre-defined` SSM-Dokument auszuführen. Die angegebenen Parameter installieren Docker in der Instance.

```
{
  "schemaVersion": "2.2",
  "description": "My composite document for bootstrapping software and installing
 Docker.",
  "parameters": {
  },
  "mainSteps": [
    {
      "action": "aws:downloadContent",
      "name": "downloadContent",
      "inputs": {
        "sourceType": "GitHub",
        "sourceInfo": "{\"owner\":\"TestUser1\",\"repository\":\"TestPublic\", \"path
\":\"documents/bootstrap/StateManagerBootstrap.yml\"}",
        "destinationPath": "bootstrap"
      }
    },
    {
      "action": "aws:runDocument",
      "name": "runDocument",
      "inputs": {
        "documentType": "LocalPath",
        "documentPath": "bootstrap",
        "documentParameters": "{}"
      }
    },
    {
```

```
      "action": "aws:runDocument",
      "name": "configureDocker",
      "inputs": {
        "documentType": "SSMDocument",
        "documentPath": "AWS-ConfigureDocker",
        "documentParameters": "{\"action\":\"Install\"}"
      }
    }
  ]
}
```

Weitere Informationen

- Informationen zum Neustarten von Servern und Instances bei Verwendung von Run Command für den Aufruf von Skripts finden Sie unter Umgang mit Neustarts beim Ausführen von Befehlen.
- Weitere Informationen zu den Plugins, die Sie einem benutzerdefinierten SSM-Dokument hinzufügen können, finden Sie unter Referenz für Befehlsdokument-Plugins.
- Informationen zum Ausführen von Dokumenten von einem Remote-Standort (ohne Erstellen eines zusammengesetzten Dokuments) finden Sie unter Ausführen von -Dokumenten von Remote-Standorten.

Arbeiten mit Dokumenten

Dieser Abschnitt enthält Informationen darüber, wie Sie SSM-Dokumente verwenden und mit ihnen arbeiten können.

Themen

- Vergleichen von SSM-Dokumentversionen
- Erstellen eines SSM-Dokuments
- Löschen benutzerdefinierter SSM-Dokumente
- Ausführen von -Dokumenten von Remote-Standorten
- Freigeben von SSM-Dokumenten
- Suchen nach SSM-Dokumenten

Vergleichen von SSM-Dokumentversionen

Sie können die Unterschiede im Inhalt zwischen Versionen von AWS Systems Manager (SSM)-Dokumenten in der Systems Manager-Dokumentenkonsole vergleichen. Beim Vergleich von

Versionen eines SSM-Dokuments werden Unterschiede zwischen dem Inhalt der Versionen hervorgehoben.

Vergleichen von SSM-Dokumentinhalten (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie in der Dokumentliste das freizugebende Dokument, dessen Inhalt Sie teilen möchten.
4. Wählen Sie auf der Registerkarte Content (Inhalt) die Option Compare versions (Versionen vergleichen) und wählen Sie die Version des Dokuments aus, mit der Sie den Inhalt vergleichen möchten.

Erstellen eines SSM-Dokuments

Nachdem Sie den Inhalt wie unter Schreiben von SSM-Dokumentinhalt beschrieben für das benutzerdefinierte SSM-Dokument erstellt haben, können Sie mithilfe der Systems Manager-Konsole ein SSM-Dokument mit Ihrem Inhalt erstellen.

So erstellen Sie ein SSM-Dokument

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie Create command or session (Befehl oder Sitzung erstellen) aus.
4. Geben Sie einen aussagekräftigen Namen für das Dokument ein.
5. (Optional) Geben Sie in Target type (Zieltyp) den Typ der Ressourcen an, auf denen das Dokument ausgeführt werden kann.
6. Wählen Sie in der Liste Document type den Typ des zu erstellenden Dokuments aus.
7. Löschen Sie die Klammern im Feld Content (Inhalt) und fügen Sie dann den zuvor erstellten Dokumentinhalt ein.
8. (Optional) Wenden Sie im Abschnitt Document tags (Dokument-Tags) ein oder mehrere Tag-Schlüssel-Name/Wert-Paare auf das Dokument an.

Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Möglicherweise möchten Sie ein Dokument markieren, um den Typ der von ihm ausgeführten Aufgaben, den Typ der Betriebssysteme, auf die es ausgerichtet ist, und die Umgebung, in der es ausgeführt wird, zu identifizieren. In diesem Fall könnten Sie z.B. die folgenden Schlüsselname-Wert-Paare angeben:

- `Key=TaskType,Value=MyConfigurationUpdate`
- `Key=OS,Value=AMAZON_LINUX_2`
- `Key=Environment,Value=Production`

9. Wählen Sie Create document aus, um das Dokument zu speichern.

Löschen benutzerdefinierter SSM-Dokumente

Wenn Sie ein benutzerdefiniertes SSM-Dokument nicht mehr verwenden möchten, können Sie es über die AWS Systems Manager-Konsole löschen.

Löschen eines SSM-Dokuments

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie das Dokument aus, die Sie löschen möchten.
4. Wählen Sie Löschen. Wenn Sie zum Löschen des Dokuments aufgefordert werden, wählen Sie Löschen.

Beispiele für die Verwendung von Befehlszeilentools oder SDKs zum Löschen von SSM-Dokumenten finden Sie unter Verwendung `DeleteDocument` mit einem AWS SDK oder CLI.

Ausführen von -Dokumenten von Remote-Standorten

Sie können AWS Systems Manager (SSM-) Dokumente von entfernten Standorten aus ausführen, indem Sie das `AWS-RunDocument` vordefinierte SSM-Dokument verwenden. Dieses Dokument unterstützt die Ausführung von SSM-Dokumenten, die an den folgenden Speicherorten gespeichert sind:

- Öffentliche und private GitHub-Repositorys (GitHub Enterprise wird nicht unterstützt)

- Amazon-S3-Buckets
- Systems Manager

Sie können Remote-Dokumente zwar auch mithilfe von State Manager oder Automation, Tools unter, ausführen. Im folgenden Verfahren wird jedoch nur beschrieben AWS Systems Manager, wie Sie Remote-SSM-Dokumente mithilfe AWS Systems Manager Run Command der Systems Manager Manager-Konsole ausführen.

> **Note**
>
> `AWS-RunDocument` kann verwendet werden, um nur SSM-Dokumente vom Befehlstyp auszuführen, nicht andere Typen wie Automation-Runbooks. Das `AWS-RunDocument` verwendet das `aws:downloadContent`-Plugin. Weitere Informationen zum `aws:downloadContent`-Plugin finden Sie unter `aws:downloadContent`.

Bevor Sie beginnen

Bevor Sie ein Remote-Dokument ausführen, müssen Sie die folgenden Aufgaben erledigen.

- Erstellen Sie ein SSM-Befehlsdokument und speichern Sie es an einem Remote-Standort. Weitere Informationen finden Sie unter Erstellen von SSM-Dokumentinhalten.
- Wenn Sie ein Remote-Dokument ausführen möchten, das in einem privaten GitHub-Repository gespeichert ist, müssen Sie einen Systems-Manager-`SecureString`-Parameter für Ihr GitHub-Sicherheitszugriffstoken erstellen. Sie können nicht auf ein Remote-Dokument in einem privaten GitHub-Repository zugreifen, indem Sie Ihr Token manuell über SSH übergeben. Das Zugriffstoken muss als `SecureString`-Systems Manager-Parameter übertragen werden. Weitere Informationen zum Erstellen eines `SecureString`-Parameters finden Sie unter Parameter Store-Parameter im Systems Manager erstellen.

Ausführen eines Remote-Dokuments (Konsole)

So führen Sie ein Remote-Dokument aus

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Run Command aus.

3. Wählen Sie Befehl ausführen aus.
4. Wählen Sie in der Liste Dokument die Option **`AWS-RunDocument`**.
5. Wählen Sie unter Command parameters (Befehlsparameter) für Source Type (Quelltyp) eine Option aus.

   - Wenn Sie GitHub auswählen, geben Sie für Quallinformationen Informationen in folgendem Format an:

```
{
    "owner": "owner_name",
    "repository": "repository_name",
    "path": "path_to_document",
    "getOptions":"branch:branch_name",
    "tokenInfo": "{{ssm-secure:secure-string-token}}"
}
```

   Zum Beispiel:

```
{
    "owner":"TestUser",
    "repository":"GitHubTestExamples",
    "path":"scripts/python/test-script",
    "getOptions":"branch:exampleBranch",
    "tokenInfo":"{{ssm-secure:my-secure-string-token}}"
}
```

> **Note**
>
> `getOptions` sind zusätzliche Optionen zum Abrufen von Inhalten aus einem anderen Branch als dem Master-Branch oder aus einem bestimmten Commit im Repository. `getOptions` kann weggelassen werden, wenn Sie den letzten Commit in der Master-Branch verwenden. Der `branch`-Parameter ist nur erforderlich, wenn Ihr SSM-Dokument in einer anderen Verzweigung als `master` gespeichert ist.
> Um die Version Ihres SSM-Dokuments in einem bestimmten Commit in Ihrem Repository zu verwenden, verwenden Sie `commitID` mit `getOptions`statt `branch`. Zum Beispiel:

```
"getOptions": "commitID:bbc1ddb94...b76d3bEXAMPLE",
```

- Wenn Sie S3 auswählen, geben Sie für Source Info Informationen in folgendem Format an:

```
{"path":"URL_to_document_in_S3"}
```

Zum Beispiel:

```
{"path":"https://s3.amazonaws.com/amzn-s3-demo-bucket/scripts/ruby/
mySSMdoc.json"}
```

- Wenn Sie SSMDocument auswählen, geben Sie für Quallinformationen Informationen in folgendem Format an:

```
{"name": "document_name"}
```

Zum Beispiel:

```
{"name": "mySSMdoc"}
```

6. Geben Sie im Feld Document Parameters Parameter für das Remote-SSM-Dokument ein. Wenn Sie beispielsweise das Dokument `AWS-RunPowerShell` ausführen, könnten Sie Folgendes angeben:

```
{"commands": ["date", "echo \"Hello World\""]}
```

Wenn Sie das Dokument `AWS-ConfigureAWSPack` ausführen, könnten Sie Folgendes angeben:

```
{
   "action":"Install",
   "name":"AWSPVDriver"
}
```

7. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

> **Tip**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

8. Für Weitere Parameter:

   - Geben Sie im Feld Kommentar Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (Sekunden) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

9. Für Ratenregelung:

   - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

10. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

    > **Note**
    >
    > Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, sind diejenigen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle

(hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, nicht die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

11. Aktivieren Sie das Kontrollkästchen SNS-Benachrichtigungen aktivieren im Abschnitt SNS-Benachrichtigungen, wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon-SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

12. Klicken Sie auf Ausführen.

> **Note**
>
> Informationen zum Neustarten von Servern und Instances bei Verwendung von Run Command für den Aufruf von Skripts finden Sie unter Umgang mit Neustarts beim Ausführen von Befehlen.

Freigeben von SSM-Dokumenten

Sie können AWS Systems Manager (SSM) Dokumente privat oder öffentlich mit Konten in demselben AWS-Region teilen. Um ein Dokument privat freizugeben, ändern Sie die Dokumentberechtigungen und erlauben bestimmten Personen entsprechend ihrer AWS-Konto -ID den Zugriff darauf. Wenn Sie ein SSM-Dokument öffentlich freigeben möchten, ändern Sie die Zugriffsberechtigungen des Dokuments und geben `All` an. Dokumente können nicht gleichzeitig öffentlich und privat freigegeben werden.

> **Warning**
>
> Verwenden Sie freigegebene SSM-Dokumente nur, wenn sie aus vertrauenswürdigen Quellen stammen. Wenn Sie ein freigegebenes Dokument verwenden, überprüfen Sie

den Inhalt des Dokuments sorgfältig, bevor Sie es verwenden, damit Sie verstehen, wie es die Konfiguration der Instance ändert. Weitere Informationen zu bewährten Methoden für freigegebene Dokumente finden Sie unter Bewährte Methoden für freigegebene SSM-Dokumente.

Einschränkungen

Beachten Sie die folgenden Einschränkungen, wenn Sie zum ersten Mal mit SSM-Dokumenten arbeiten.

- Nur der Eigentümer eines Dokuments kann ein Dokument freigeben.
- Sie müssen die Freigabe eines Dokuments aufheben, bevor Sie ein Dokument löschen können. Weitere Informationen finden Sie unter Modifizieren von Berechtigungen für ein freigegebenes SSM-Dokument.
- Sie können ein Dokument mit maximal 1000 AWS-Konten Personen teilen. Sie können über das Support Center eine Erhöhung dieses Limits anfordern. Wählen Sie als Limittyp EC2 Systems Manager und beschreiben Sie den Grund für die Anfrage.
- Sie können maximal fünf SSM-Dokumente öffentlich freigeben. Sie können über das Support Center eine Erhöhung dieses Limits anfordern. Wählen Sie als Limittyp EC2 Systems Manager und beschreiben Sie den Grund für die Anfrage.
- Dokumente können AWS-Region nur in demselben Konto mit anderen Konten geteilt werden. Die Freigabe über Regionsgrenzen hinweg wird nicht unterstützt.

Important

In Systems Manager ist ein zu Amazon gehörendes SSM-Dokument ein Dokument, das von Amazon Web Services selbst erstellt und verwaltet wird. Dokumente, die Amazon gehören, enthalten ein Präfix wie `AWS-*` im Dokumentnamen. Als Eigentümer des Dokuments gilt Amazon und nicht als ein bestimmtes Benutzerkonto innerhalb AWS. Diese Dokumente sind öffentlich zugänglich und können von allen verwendet werden.

Weitere Informationen zu Service Quotas für Systems Manager finden Sie unter AWS Systems Manager Service Quotas.

Inhalt

- Bewährte Methoden für freigegebene SSM-Dokumente
- Öffentliche Freigabe für SSM-Dokumente blockieren
- Freigeben eines SSM-Dokuments
- Modifizieren von Berechtigungen für ein freigegebenes SSM-Dokument
- Verwenden von freigegebenen SSM-Dokumenten

Bewährte Methoden für freigegebene SSM-Dokumente

Überprüfen Sie die folgenden Richtlinien, bevor Sie ein Dokument freigeben oder ein gemeinsam genutztes Dokument verwenden.

Entfernen sensibler Daten

Überprüfen Sie Ihr AWS Systems Manager (SSM-) Dokument sorgfältig und entfernen Sie alle vertraulichen Informationen. Stellen Sie beispielsweise sicher, dass das Dokument Ihre AWS Anmeldeinformationen nicht enthält. Wenn Sie ein Dokument für bestimmten Personen freigeben, können die Informationen in dem Dokument anzeigen. Wenn Sie ein Dokument öffentlich freigeben, können beliebige Personen die Informationen in dem Dokument anzeigen.

Öffentliche Freigabe für Dokumente blockieren

Überprüfen Sie alle öffentlich geteilten SSM-Dokumente in Ihrem Konto und bestätigen Sie, ob Sie sie weiterhin teilen möchten. Um die gemeinsame Nutzung eines Dokuments für die Öffentlichkeit zu beenden, müssen Sie die Einstellung für die Dokumentberechtigungen wie im Modifizieren von Berechtigungen für ein freigegebenes SSM-Dokument-Abschnitt dieses Themas beschrieben ändern. Die Aktivierung der Einstellung „Öffentliches Teilen blockieren“ hat keine Auswirkungen auf Dokumente, die Sie derzeit für die Öffentlichkeit freigeben. Sofern Ihr Anwendungsfall nicht erfordert, dass Sie Dokumente öffentlich freigeben, empfehlen wir Ihnen, die Einstellung „Öffentliche Freigabe blockieren“ für Ihre SSM-Dokumente im Abschnitt Einstellungen der Systems-Manager-Dokumentenkonsole zu aktivieren. Wenn Sie diese Einstellung aktivieren, wird unerwünschter Zugriff auf Ihre SSM-Dokumente verhindert. Die Einstellung „Öffentliche Freigabe sperren“ ist eine Einstellung auf Kontoebene, die sich für jede AWS-Region unterscheiden kann.

Beschränken von Run Command-Aktionen mithilfe einer IAM-Vertrauensrichtlinie

Erstellen Sie eine restriktive Richtlinie AWS Identity and Access Management (IAM) für Benutzer, die Zugriff auf das Dokument haben werden. Die IAM-Richtlinie legt fest, welche SSM-Dokumente ein Benutzer entweder in der Amazon Elastic Compute Cloud (Amazon EC2) -Konsole oder durch Aufrufen `ListDocuments` mit AWS Command Line Interface (AWS CLI) oder sehen kann. AWS

Tools for Windows PowerShell Die Richtlinie schränkt auch die Aktionen ein, die der Benutzer mit SSM-Dokumenten durchführen kann. Sie können eine restriktive Richtlinie erstellen, damit Benutzer nur bestimmte Dokumente verwenden können. Weitere Informationen finden Sie unter Beispiele für vom Kunden verwaltete Richtlinien.

Vorsicht bei der Verwendung freigegebener SSM-Dokumente

Überprüfen Sie den Inhalt jedes Dokuments, das für Sie freigegeben ist, insbesondere öffentliche Dokumente, um die Befehle zu verstehen, die über Ihre Instances ausgeführt werden. Ein Dokument kann absichtlich oder unbeabsichtigterweise negative Auswirkungen haben, wenn es ausgeführt wird. Wenn das Dokument auf ein externes Netzwerk verweist, überprüfen Sie die externe Quelle, bevor Sie das Dokument verwenden.

Versenden von Befehlen mit dem Dokument-Hash

Wenn Sie ein Dokument freigeben, erstellt das System einen SHA-256-Hash und weist diesen dem Dokument zu. Das System speichert außerdem einen Snapshot des Dokumentinhalts. Wenn Sie mit einem freigegebenen Dokument einen Befehl senden, können Sie für den Befehl diesen Hash angeben, um sicherzustellen, dass die folgenden Bedingungen erfüllt sind:

- Sie führen den Befehl über das richtige Systems Manager-Dokument aus.
- Der Inhalt des Dokuments wurde nicht geändert, seit es für Sie freigegeben wurde.

Wenn der Hash nicht mit dem angegebenen Dokument übereinstimmt oder der Inhalt des freigegebenen Dokuments geändert wurde, löst der Befehl eine `InvalidDocument`-Ausnahmebedingung aus. Mit dem Hash können keine Dokumentinhalte von externen Standorten überprüft werden.

Verwenden des Interpolationsparameters, um die Sicherheit zu verbessern

Verwenden Sie für Parameter vom Typ `String` in Ihren SSM-Dokumenten den Parameter und den Wert `interpolationType": "ENV_VAR`, um die Sicherheit vor Befehlseinschleusungen zu erhöhen. Parametereingaben werden dadurch als Zeichenfolgeliterale und nicht als potenziell ausführbare Befehle behandelt. In diesem Fall erstellt der Agent eine Umgebungsvariable mit dem Namen `SSM_`*`parameter-name`*, die den Wert des Parameters enthält. Wir empfehlen, allen SSM-Dokumenten mit Parametern vom Typ `String` den Parameter `"interpolationType": "ENV_VAR"` hinzuzufügen. Weitere Informationen finden Sie unter the section called “Schreiben von SSM-Dokumentinhalt”.

Öffentliche Freigabe für SSM-Dokumente blockieren

Bevor Sie beginnen, überprüfen Sie alle öffentlich freigegebenen SSM-Dokumente in Ihrem AWS-Konto und bestätigen Sie, ob Sie diese weiterhin freigeben möchten. Um die gemeinsame Nutzung eines SSM-Dokuments für die Öffentlichkeit zu beenden, müssen Sie die Einstellung für Dokumentberechtigungen wie im Modifizieren von Berechtigungen für ein freigegebenes SSM-Dokument-Abschnitt dieses Themas beschrieben ändern. Die Aktivierung der Einstellung „Öffentliches Teilen blockieren“ hat keine Auswirkungen auf SSM-Dokumente, die Sie derzeit für die Öffentlichkeit freigeben. Wenn die Einstellung „Öffentliches Teilen blockieren“ aktiviert ist, können Sie keine weiteren SSM-Dokumente mit der Öffentlichkeit teilen.

Sofern Ihr Anwendungsfall nicht erfordert, dass Sie Dokumente öffentlich freigeben, empfehlen wir Ihnen, die Einstellung „Öffentliche Freigabe blockieren“ für Ihre SSM-Dokumente zu aktivieren. Wenn Sie diese Einstellung aktivieren, wird unerwünschter Zugriff auf Ihre SSM-Dokumente verhindert. Bei der Einstellung „Öffentliches Teilen blockieren“ handelt es sich um eine Einstellung auf Kontoebene, die für jedes Konto unterschiedlich sein kann. AWS-Region Führen Sie die folgenden Aufgaben aus, um die öffentliche Freigabe für alle SSM-Dokumente zu blockieren, die Sie derzeit nicht freigeben.

Öffentliche Freigabe blockieren (Konsole)

Blockieren der öffentlichen Freigabe Ihrer SSM-Dokumente

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie Preferences (Präferenzen) und dann Edit (Bearbeiten) im Abschnitt Block public sharing (Öffentliche Freigabe blockieren) .
4. Wählen Sie das Kontrollkästchen Block public sharing (Öffentliche Freigabe blockieren) und wählen Sie aus Save (speichern).

Öffentliche Freigabe blockieren (Befehlszeile)

Öffnen Sie AWS Command Line Interface (AWS CLI) oder AWS Tools for Windows PowerShell auf Ihrem lokalen Computer und führen Sie den folgenden Befehl aus, um das öffentliche Teilen Ihrer SSM-Dokumente zu blockieren.

Linux & macOS

```
aws ssm update-service-setting  \
```

```
    --setting-id /ssm/documents/console/public-sharing-permission \
    --setting-value Disable \
    --region 'The AWS-Region you want to block public sharing in'
```

Windows

```
aws ssm update-service-setting ^
    --setting-id /ssm/documents/console/public-sharing-permission ^
    --setting-value Disable ^
    --region "The AWS-Region you want to block public sharing in"
```

PowerShell

```
Update-SSMServiceSetting `
    -SettingId /ssm/documents/console/public-sharing-permission `
    -SettingValue Disable `
    -Region The AWS-Region you want to block public sharing in
```

Überprüfen Sie, ob der Einstellungswert aktualisiert wurde, indem Sie den folgenden Befehl verwenden.

Linux & macOS

```
aws ssm get-service-setting   \
    --setting-id /ssm/documents/console/public-sharing-permission \
    --region The AWS-Region you blocked public sharing in
```

Windows

```
aws ssm get-service-setting  ^
    --setting-id /ssm/documents/console/public-sharing-permission ^
    --region "The AWS-Region you blocked public sharing in"
```

PowerShell

```
Get-SSMServiceSetting `
    -SettingId /ssm/documents/console/public-sharing-permission `
    -Region The AWS-Region you blocked public sharing in
```

Beschränken des Zugriffs zum Blockieren der öffentlichen Freigabe mit IAM

Sie können AWS Identity and Access Management (IAM) -Richtlinien erstellen, die Benutzer daran hindern, die Einstellung „Öffentliches Teilen blockieren“ zu ändern. Dadurch wird verhindert, dass Benutzer unerwünschten Zugriff auf Ihre SSM-Dokumente zulassen.

Nachfolgend finden Sie ein Beispiel für eine IAM-Richtlinie, die verhindert, dass Benutzer die Einstellung zum Blockieren der öffentlichen Freigabe zu aktualisieren. Um dieses Beispiel zu verwenden, müssen Sie die Beispiel-Konto-ID für Amazon Web Services durch Ihre eigene Konto-ID ersetzen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ssm:UpdateServiceSetting",
            "Resource": "arn:aws:ssm:*:444455556666:servicesetting/ssm/documents/
console/public-sharing-permission"
        }
    ]
}
```

Freigeben eines SSM-Dokuments

Sie können AWS Systems Manager (SSM) Dokumente mithilfe der Systems Manager Manager-Konsole teilen. Beim Teilen von Dokumenten über die Konsole kann nur die Standardversion des Dokuments geteilt werden. Sie können SSM-Dokumente auch programmgesteuert teilen, indem Sie die `ModifyDocumentPermission` API-Operation mit dem AWS Command Line Interface (AWS CLI) AWS Tools for Windows PowerShell, oder dem SDK aufrufen. AWS Bevor Sie ein Dokument teilen, sollten Sie die Personen AWS-Konto IDs ermitteln, mit denen Sie es teilen möchten. Sie geben diese Konten an IDs , wenn Sie das Dokument teilen.

Freigeben eines Dokuments (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie in der Dokumentenliste das Dokument aus, das Sie freigeben möchten, und klicken Sie dann auf View details (Details anzeigen). Überprüfen Sie dann auf der Registerkarte Permissions, ob Sie der Besitzer des Dokuments sind. Nur der Eigentümer eines Dokuments kann ein Dokument freigeben.
4. Wählen Sie Bearbeiten aus.
5. Um den Befehl öffentlich freizugeben, wählen Sie Public und dann die Option Save. Wählen Sie zur privaten Freigabe des Befehls die Option Private aus, geben Sie die AWS-Konto -ID ein und wählen Sie Add permission sowie anschließend die Option Save aus.

Freigeben eines Dokuments (Befehlszeile)

Das folgende Verfahren erfordert, dass Sie eine AWS-Region für Ihre Befehlszeilensitzung angeben.

1. Öffnen Sie AWS CLI oder AWS Tools for Windows PowerShell auf Ihrem lokalen Computer und führen Sie den folgenden Befehl aus, um Ihre Anmeldeinformationen anzugeben.

   Ersetzen Sie den Befehl im folgenden Befehl *`region`* durch Ihre eigenen Informationen. Eine Liste der unterstützten *`region`* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

   Linux & macOS

   ```
   aws config

   AWS Access Key ID: [your key]
   AWS Secret Access Key: [your key]
   Default region name: region
   Default output format [None]:
   ```

   Windows

   ```
   aws config

   AWS Access Key ID: [your key]
   AWS Secret Access Key: [your key]
   Default region name: region
   Default output format [None]:
   ```

PowerShell

```
Set-AWSCredentials -AccessKey your key -SecretKey your key
Set-DefaultAWSRegion -Region region
```

2. Verwenden Sie den folgenden Befehl, um alle SSM-Dokumente aufzulisten, die für Sie verfügbar sind. Die Liste enthält Dokumente, die Sie erstellt haben, und Dokumente, die für Sie freigegeben wurden.

   Linux & macOS

   ```
   aws ssm list-documents
   ```

   Windows

   ```
   aws ssm list-documents
   ```

   PowerShell

   ```
   Get-SSMDocumentList
   ```

3. Verwenden Sie den folgenden Befehl, um ein bestimmtes Dokument abzurufen.

   Linux & macOS

   ```
   aws ssm get-document \
       --name document name
   ```

   Windows

   ```
   aws ssm get-document ^
       --name document name
   ```

   PowerShell

   ```
   Get-SSMDocument `
       -Name document name
   ```

4. Verwenden Sie den folgenden Befehl, um eine Beschreibung des Dokuments abzurufen.

Linux & macOS

```
aws ssm describe-document \
    --name document name
```

Windows

```
aws ssm describe-document ^
    --name document name
```

PowerShell

```
Get-SSMDocumentDescription `
    -Name document name
```

5. Verwenden Sie den folgenden Befehl, um die Zugriffsberechtigungen für das Dokument anzuzeigen.

Linux & macOS

```
aws ssm describe-document-permission \
    --name document name \
    --permission-type Share
```

Windows

```
aws ssm describe-document-permission ^
    --name document name ^
    --permission-type Share
```

PowerShell

```
Get-SSMDocumentPermission `
    -Name document name `
    -PermissionType Share
```

6. Verwenden Sie den folgenden Befehl, um die Zugriffsberechtigungen für das Dokument zu ändern und das Dokument freizugeben. Sie müssen der Eigentümer des Dokuments sein, um die Berechtigungen bearbeiten zu können. Optional können Sie für Dokumente, die mit

bestimmten AWS-Konto IDs Personen geteilt wurden, mithilfe des `--shared-document-version` Parameters eine Version des Dokuments angeben, die Sie teilen möchten. Wenn Sie keine Version angeben, gibt das System die `Default`-Version des Dokuments frei. Wenn Sie ein Dokument öffentlich (mit `all`) teilen, werden standardmäßig alle Versionen des angegebenen Dokuments geteilt. Mit dem folgenden Beispielbefehl wird das Dokument privat für eine bestimmte Person freigegeben, basierend auf der AWS-Konto ID dieser Person.

Linux & macOS

```
aws ssm modify-document-permission \
    --name document name \
    --permission-type Share \
    --account-ids-to-add AWS-Konto ID
```

Windows

```
aws ssm modify-document-permission ^
    --name document name ^
    --permission-type Share ^
    --account-ids-to-add AWS-Konto ID
```

PowerShell

```
Edit-SSMDocumentPermission `
    -Name document name `
    -PermissionType Share `
    -AccountIdsToAdd AWS-Konto ID
```

7. Verwenden Sie den folgenden Befehl, um ein Dokument öffentlich freizugeben.

> **Note**
>
> Wenn Sie ein Dokument öffentlich (mit `all`) teilen, werden standardmäßig alle Versionen des angegebenen Dokuments geteilt.

Linux & macOS

```
aws ssm modify-document-permission \
    --name document name \
```

```
    --permission-type Share \
    --account-ids-to-add 'all'
```

Windows

```
aws ssm modify-document-permission ^
    --name document name ^
    --permission-type Share ^
    --account-ids-to-add "all"
```

PowerShell

```
Edit-SSMDocumentPermission `
    -Name document name `
    -PermissionType Share `
    -AccountIdsToAdd ('all')
```

Modifizieren von Berechtigungen für ein freigegebenes SSM-Dokument

Wenn Sie einen Befehl teilen, können Benutzer diesen Befehl anzeigen und verwenden, bis Sie entweder den Zugriff auf das AWS Systems Manager (SSM-) Dokument aufheben oder das SSM-Dokument löschen. Sie können ein Dokument jedoch erst löschen, wenn es nicht mehr freigegeben ist. Sie müssen also zuerst die Freigabe beenden und können erst anschließend die Datei löschen.

Beenden der Freigabe eines Dokuments (Konsole)

Beenden der Freigabe eines Dokuments

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie in der Dokumentenliste das Dokument aus, dessen Freigabe Sie beenden möchten, und klicken Sie dann auf Details anzeigen. Überprüfen Sie dann auf der Registerkarte Berechtigungen, ob Sie der Besitzer des Dokuments sind. Nur der Eigentümer eines Dokuments kann die Freigabe eines Dokuments beenden.
4. Wählen Sie Bearbeiten aus.
5. Wählen Sie X aus, um die AWS-Konto ID zu löschen, die keinen Zugriff mehr auf den Befehl haben sollte, und wählen Sie dann Speichern.

Beenden der Freigabe eines Dokuments (Befehlszeile)

Öffnen Sie AWS CLI oder AWS Tools for Windows PowerShell auf Ihrem lokalen Computer und führen Sie den folgenden Befehl aus, um die gemeinsame Nutzung eines Befehls zu beenden.

Linux & macOS

```
aws ssm modify-document-permission \
    --name document name \
    --permission-type Share \
    --account-ids-to-remove 'AWS-Konto ID'
```

Windows

```
aws ssm modify-document-permission ^
    --name document name ^
    --permission-type Share ^
    --account-ids-to-remove "AWS-Konto ID"
```

PowerShell

```
Edit-SSMDocumentPermission `
    -Name document name `
    -PermissionType Share `
    -AccountIdsToRemove AWS-Konto ID
```

Verwenden von freigegebenen SSM-Dokumenten

Wenn Sie ein AWS Systems Manager (SSM) -Dokument teilen, generiert das System einen Amazon-Ressourcennamen (ARN) und weist ihn dem Befehl zu. Wenn Sie ein freigegebenes Dokument über die Systems-Manager-Konsole auswählen und ausführen, wird der ARN nicht angezeigt. Wenn Sie jedoch ein freigegebenes SSM-Dokument mit einer anderen Methode als der Systems-Manager-Konsole ausführen möchten, müssen Sie den vollständigen ARN des Dokuments für den `DocumentName`-Anforderungsparameter angeben. Wenn Sie den Befehl zum Auflisten der Dokumente ausführen, wird jeweils der vollständige ARN für SSM-Dokumente angezeigt.

> **Note**
>
> Sie müssen nicht angeben, ob es sich ARNs um AWS öffentliche Dokumente (Dokumente, die mit 1 beginnen`AWS-*`) oder um Dokumente handelt, deren Eigentümer Sie sind.

Verwenden eines freigegebenen SSM-Dokuments (Befehlszeile)

So listen Sie öffentliche SSM-Dokumente auf

Linux & macOS

```
aws ssm list-documents \
    --filters Key=Owner,Values=Public
```

Windows

```
aws ssm list-documents ^
    --filters Key=Owner,Values=Public
```

PowerShell

```
$filter = New-Object Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$filter.Key = "Owner"
$filter.Values = "Public"

Get-SSMDocumentList `
    -Filters @($filter)
```

So listen Sie private SSM-Dokumente auf, die für Sie freigegeben wurden

Linux & macOS

```
aws ssm list-documents \
    --filters Key=Owner,Values=Private
```

Windows

```
aws ssm list-documents ^
    --filters Key=Owner,Values=Private
```

PowerShell

```
$filter = New-Object Amazon.SimpleSystemsManagement.Model.DocumentKeyValuesFilter
$filter.Key = "Owner"
$filter.Values = "Private"

Get-SSMDocumentList `
    -Filters @($filter)
```

So listen Sie alle SSM-Dokumente auf, die für Sie verfügbar sind

Linux & macOS

```
aws ssm list-documents
```

Windows

```
aws ssm list-documents
```

PowerShell

```
Get-SSMDocumentList
```

So rufen Sie Informationen zu einem SSM-Dokument ab, das für Sie freigegeben wurde

Linux & macOS

```
aws ssm describe-document \
    --name arn:aws:ssm:us-east-2:12345678912:document/documentName
```

Windows

```
aws ssm describe-document ^
    --name arn:aws:ssm:us-east-2:12345678912:document/documentName
```

PowerShell

```
Get-SSMDocumentDescription `
    -Name arn:aws:ssm:us-east-2:12345678912:document/documentName
```

So führen Sie ein freigegebenes SSM-Dokument aus

Linux & macOS

```
aws ssm send-command \
    --document-name arn:aws:ssm:us-east-2:12345678912:document/documentName \
    --instance-ids ID
```

Windows

```
aws ssm send-command ^
    --document-name arn:aws:ssm:us-east-2:12345678912:document/documentName ^
    --instance-ids ID
```

PowerShell

```
Send-SSMCommand `
    -DocumentName arn:aws:ssm:us-east-2:12345678912:document/documentName `
    -InstanceIds ID
```

Suchen nach SSM-Dokumenten

Sie können den AWS Systems Manager (SSM-) Dokumentenspeicher nach SSM-Dokumenten durchsuchen, indem Sie entweder die Freitextsuche oder eine filterbasierte Suche verwenden. Sie können auch Dokumente als Favoriten markieren, um häufig verwendete SSM-Dokumente zu finden. In den folgenden Abschnitten wird beschrieben, wie Sie diese Funktionen nutzen können.

Verwenden der Freitextsuche

Das Suchfeld auf der Seite Systems Manager-Dokumente unterstützt die Freitextsuche. Die Freitextsuche vergleicht den bzw. die eingegebenen Suchbegriffe mit dem Dokumentnamen in jedem SSM-Dokument. Wenn Sie einen einzelnen Suchbegriff eingeben, z. B. **`ansible`**, gibt Systems Manager alle SSM-Dokumente zurück, in denen dieser Begriff erkannt wurde. Wenn Sie mehrere Suchbegriffe eingeben, sucht Systems Manager mithilfe einer `OR`-Anweisung. Wenn Sie z. B. **`ansible`** und **`linux`** angeben, gibt die Suche alle Dokumente zurück, die eines der beiden Schlüsselwörter im Namen tragen.

Wenn Sie einen Freitext-Suchbegriff eingeben und eine Suchoption wählen, z. B. Plattformtyp, dann verwendet die Suche eine `AND`-Anweisung und gibt alle Dokumente zurück, die das Schlüsselwort im Namen und den angegebenen Plattformtyp enthalten.

> **Note**
>
> Beachten Sie die folgenden Details zur Freitextsuche.
>
> - Bei der Freitextsuche wird nicht zwischen Groß- und Kleinschreibung unterschieden.
> - Die Suchbegriffe müssen mindestens drei und dürfen höchstens 20 Zeichen lang sein.
> - Die Freitextsuche akzeptiert bis zu fünf Suchbegriffe.
> - Wenn Sie ein Leerzeichen zwischen den Suchbegriffen eingeben, schließt das System das Leerzeichen bei der Suche ein.
> - Sie können die Freitextsuche mit anderen Suchoptionen wie Dokumenttyp oder Plattformtyp kombinieren.
> - Der Filter Dokumentname-Präfix und die Freitextsuche können nicht zusammen verwendet werden, da sie sich gegenseitig ausschließen.

Suchen nach SSM-Dokumenten

1. Öffnen Sie die Konsole unter AWS Systems Manager . https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Geben Sie Ihre Suchbegriffe in das Suchfeld ein und drücken Sie die Eingabetaste.

Durchführen einer Freitextdokumentsuche mit dem AWS CLI

Durchführen der Freitextdokumentsuche mithilfe der CLI

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus, um eine Freitextdokumentsuche mit einem einzelnen Begriff durchzuführen. Ersetzen Sie diesen Befehl *search_term* durch Ihre eigenen Informationen.

   ```
   aws ssm list-documents --filters Key="SearchKeyword",Values="search_term"
   ```

Ein Beispiel:

```
aws ssm list-documents --filters Key="SearchKeyword",Values="aws-asg" --region us-
east-2
```

Um mit mehreren Begriffen zu suchen, die eine AND-Anweisung erstellen, führen Sie den folgenden Befehl aus. Ersetzen Sie in diesem Befehl *search_term_1* und *search_term_2* durch Ihre eigenen Informationen.

```
aws ssm list-documents --filters
 Key="SearchKeyword",Values="search_term_1","search_term_2","search_term_3" --
region us-east-2
```

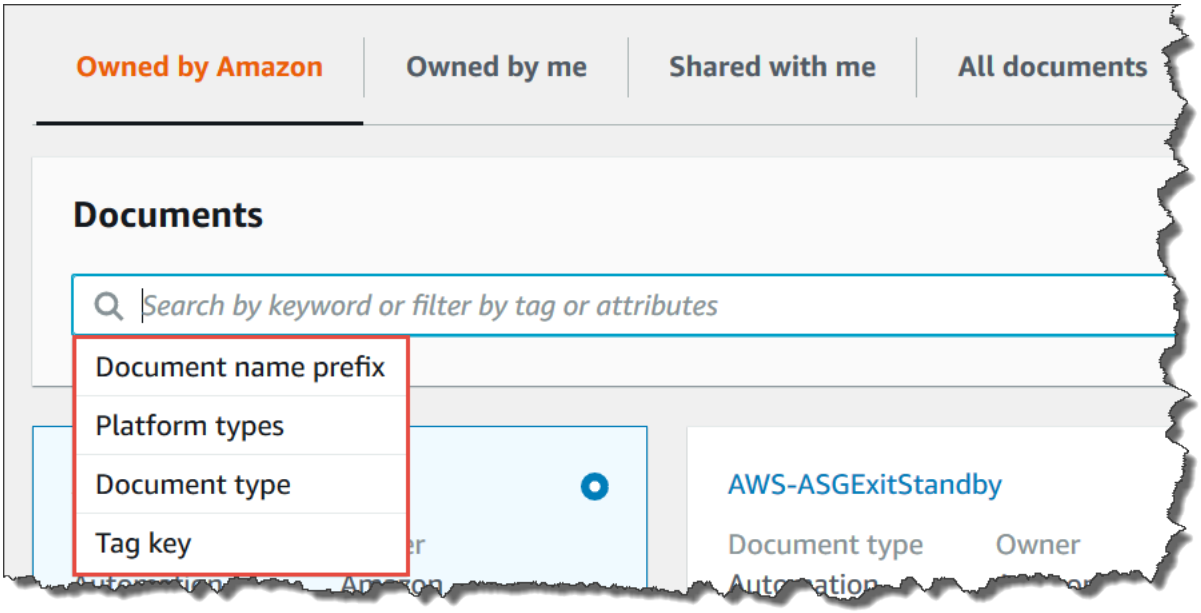
Ein Beispiel:

```
aws ssm list-documents --filters Key="SearchKeyword",Values="aws-asg","aws-
ec2","restart" --region us-east-2
```

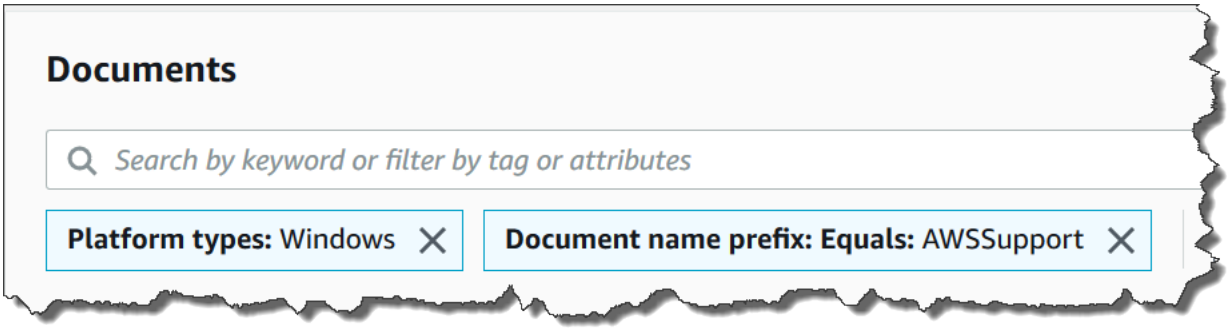
Verwenden von Filtern

Der Systems Manager–Seite Dokumente zeigt automatisch die folgenden Filter an, wenn Sie das Suchfeld auswählen.

- Dokumentnamenpräfix
- Plattformentypen
- Dokumenttyp
- Tag-Schlüssel



Sie können mit einem einzigen Filter nach SSM-Dokumenten suchen. Wenn Sie einen spezifischeren Satz von SSM-Dokumenten zurückgeben möchten, können Sie mehrere Filter anwenden. Hier ein Beispiel für eine Suche, bei der die Filter Plattformtypen und Dokumentnamenpräfix verwendet werden.



Wenn Sie mehrere Filter anwenden, erstellt Systems Manager verschiedene Suchanweisungen basierend auf den ausgewählten Filtern:

- Wenn Sie denselben Filter mehrfach anwenden, z. B. das Präfix für den Dokumentennamenpräfix, sucht Systems Manager mit Hilfe einer OR-Anweisung. Wenn Sie z. B. einen Filter Dokumentname Präfix=**`AWS`** und einen zweiten Filter Dokumentnamenpräfix=**`Lambda`** angeben, liefert die Suche alle Dokumente mit dem Präfix „`AWS`“ und alle Dokumente mit dem Präfix „`Lambda`“.
- Wenn Sie verschiedene Filter anwenden, z. B. Document name prefix (Präfix Dokumentname) und Platform types (Plattformtypen), sucht Systems Manager mithilfe einer AND-Anweisung. Wenn Sie z. B. den Filter Document name prefix (Präfix Dokumentname) = **`AWS`** und den Filter Platform types

(Plattformtypen) = **`Linux`** angeben, gibt die Suche alle Dokumente mit dem Präfix „`AWS`“ zurück, die spezifisch für die Linux-Plattform sind.

> **Note**
>
> Dabei wird Groß- und Kleinschreibung beachtet.

Hinzufügen von Dokumenten zu Ihren Favoriten

Fügen Sie Dokumente zu Ihren Favoriten hinzu, um häufig verwendete SSM-Dokumente leichter zu finden. Sie können bis zu 20 Dokumente pro Dokumenttyp, pro AWS-Konto und als Favorit markieren AWS-Region. Sie können Ihre Favoriten in der Dokumenten- AWS-Managementkonsole auswählen, ändern und anzeigen. Die folgenden Verfahren beschreiben, wie Sie Ihre Favoriten auswählen, ändern und anzeigen.

So markieren Sie ein SSM-Dokument als Favorit

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie das Sternsymbol neben dem Namen des Dokuments aus, das Sie als Favorit markieren möchten.

So entfernen Sie ein SSM-Dokument aus Ihren Favoriten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie das Sternsymbol neben dem Namen des Dokuments ab, das Sie nicht mehr als Favorit markieren möchten.

Um Ihre Favoriten aus den Dokumenten anzuzeigen AWS-Managementkonsole

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich die Option Dokumente aus.
3. Wählen Sie die Registerkarte Favoriten aus.

Beheben von Fehlern mit der Parameterverarbeitung

Häufige Probleme bei der Parameterverarbeitung

Umgebungsvariablen sind während der Ausführung nicht verfügbar

Problem: Befehle schlagen fehl, weil Umgebungsvariablen (`SSM_parameter-name`) nicht gefunden werden.

Mögliche Ursachen:

- Die Version von SSM Agent unterstützt keine Interpolation von Umgebungsvariablen.
- `interpolationType` ist nicht auf `ENV_VAR` gesetzt.
- Der Parametername stimmt nicht mit dem erwarteten Namen der Umgebungsvariablen überein.

Solution (Lösung):

- Stellen Sie sicher, dass die SSM Agent-Version 3.3.2746.0 oder höher ist.
- Fügen Sie eine Fallback-Logik für ältere Agentenversionen hinzu:

```
if [ -z "${SSM_parameterName+x}" ]; then
    export SSM_parameterName="{{parameterName}}"
fi
```

Parameterwerte mit Sonderzeichen

Problem: Befehle schlagen fehl, wenn Parameterwerte Leerzeichen, Anführungszeichen oder andere Sonderzeichen enthalten.

Solution (Lösung):

- Verwenden Sie korrekte Anführungszeichen, wenn Sie auf Umgebungsvariablen verweisen:

```
# Correct
echo "$SSM_parameter-name"

# Incorrect
echo $SSM_parameter-name
```

- Fügen Sie mit `allowedPattern` eine Eingabevalidierung hinzu, um Sonderzeichen einzuschränken.

Inkonsistentes Verhalten auf verschiedenen Plattformen

Problem: Die Parameterverarbeitung funktioniert in Linux- und Windows Server-Systemen unterschiedlich.

Solution (Lösung):

- Verwenden Sie die plattformspezifische Syntax für Umgebungsvariablen:

```
# PowerShell
$env:SSM_parameter-name

# Bash
$SSM_parameter-name
```

- Verwenden Sie plattformspezifische Vorbedingungsprüfungen in Ihrem Dokument.

Parameterwerte wurden nicht korrekt maskiert

Problem: Mangelnder Schutz vor Befehlseinschleusung trotz Interpolation von Umgebungsvariablen.

Solution (Lösung):

- Achten Sie immer auf die ordnungsgemäße Maskierung, wenn Sie Parameterwerte in Befehle aufnehmen:

```
# Correct
mysql_command="mysql -u \"$SSM_username\" -p\"$SSM_password\""

# Incorrect
mysql_command="mysql -u $SSM_username -p$SSM_password"
```

Tipps für die Parametervalidierung

Nutzen Sie diese Methoden, um Ihre Parameterverarbeitung zu validieren:

1. Verfügbarkeit von Umgebungsvariablen testen:

```
#!/bin/bash
```

```
# Print all SSM_ environment variables
env | grep ^SSM_

# Test specific parameter
if [ -n "$SSM_parameter" ]; then
    echo "Parameter is available"
else
    echo "Parameter is not available"
fi
```

2. Parametermuster überprüfen:

```
parameters:
  myParameter:
    type: String
    allowedPattern: "^[a-zA-Z0-9_-]+$"
    description: "Test this pattern with sample inputs"
```

3. Fehlerbehandlung einbeziehen:

```
if [[ ! "$SSM_parameter" =~ ^[a-zA-Z0-9_-]+$ ]]; then
    echo "Parameter validation failed"
    exit 1
fi
```

AWS Systems Manager Maintenance Windows

Maintenance Windows, ein Tool in AWS Systems Manager, hilft Ihnen dabei, einen Zeitplan für die Ausführung potenziell störender Aktionen auf Ihren Knoten zu definieren, z. B. das Patchen eines Betriebssystems, das Aktualisieren von Treibern oder das Installieren von Software oder Patches.

> **Note**
>
> State Manager und Maintenance Windows können ähnliche Arten von Updates für Ihre verwalteten Knoten ausführen. Welche Option Sie wählen, hängt davon ab, ob Sie die System-Compliance automatisieren oder zeitkritische Aufgaben mit hoher Priorität während der von Ihnen angegebenen Zeiträume ausführen müssen.
> Weitere Informationen finden Sie unter Auswahl zwischen State Manager und Maintenance Windows.

Mit Maintenance Windows können Sie Aktionen für zahlreiche andere AWS Ressourcentypen planen, z. B. Amazon Simple Storage Service (Amazon S3) -Buckets, Amazon Simple Queue Service (Amazon SQS) -Warteschlangen, AWS Key Management Service (AWS KMS) -Schlüssel und vieles mehr.

Eine vollständige Liste der unterstützten Ressourcentypen, die Sie in ein Wartungsfensterziel aufnehmen können, finden Sie unter Ressourcen, die Sie mit verwenden können AWS - Ressourcengruppen und Tag-Editor im AWS -Ressourcengruppen Benutzerhandbuch. Um mit Maintenance Windows zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Maintenance Windows aus.

Jedes Wartungsfenster hat einen Zeitplan, eine maximale Dauer, eine Reihe registrierter Ziele (die verwalteten Knoten oder andere AWS Ressourcen, auf die reagiert wird) und eine Reihe registrierter Aufgaben. Sie können Tags zu Ihren Wartungsfenstern hinzufügen, wenn Sie sie erstellen oder aktualisieren. Tags sind Schlüssel, die das Identifizieren und Sortieren der Ressourcen in Ihrer Organisation ermöglichen. Sie können auch Daten angeben, vor oder nach denen ein Wartungsfenster nicht ausgeführt werden soll und Sie können die internationale Zeitzone angeben, auf der der Zeitplan des Wartungsfensters basieren soll.

Eine Beschreibung des Verhältnisses zwischen den verschiedenen zeitplanbezogenen Optionen für Wartungsfenster finden Sie unter Wartungsfenster-Optionen für Planung und aktive Zeiträume.

Weitere Informationen zum Arbeiten mit der `--schedule`-Option finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

Unterstützte Aufgabentypen

Mit Wartungsfenstern können Sie vier Aufgabentypen ausführen:

- Befehle in Run Command, einem Tool in Systems Manager

  Mehr über Run Command erfahren Sie unter AWS Systems Manager Run Command.
- Workflows in Automation, ein Tool in Systems Manager

  Weitere Informationen über Automation-Workflows finden Sie unter AWS Systems Manager-Automatisierung.
- Funktionen in AWS Lambda

  Weitere Informationen über Lambda-Funktionen finden Sie unter Erste Schritte mit Lambda im AWS Lambda -Entwicklerhandbuch.

- Aufgaben in AWS Step Functions

> **Note**
>
> Aufgaben im Wartungsfenster unterstützen nur Zustandsmaschinen-Workflows von Step Functions Standard. Sie unterstützen keine Express-Zustandsmaschinen-Workflows. Informationen zu Workflowtypen für Zustandsmaschinen finden Sie unter Standard- gegenüber Express-Workflows im AWS Step Functions -Entwicklerhandbuch.

Weitere Informationen zu Step Functions finden Sie im AWS Step Functions Developer Guide.

Das bedeutet, dass Sie mit Wartungsfenstern z. B. folgende Aufgaben für Ihre ausgewählten Ziele durchführen können.

- Installieren oder Aktualisieren von Anwendungen.
- Anwenden von Patches.
- Installieren oder Aktualisieren von SSM Agent
- Führen Sie PowerShell Befehle und Linux-Shell-Skripts mithilfe einer Systems Manager Run Command Manager-Task aus.
- Erstellen von Amazon Machine Images (AMIs), Bootstrappen von Software und Konfigurieren von Knoten mit einer Systems-Manager-Automatisierungs-Aufgabe.
- Führen Sie AWS Lambda Funktionen aus, die zusätzliche Aktionen aufrufen, z. B. das Scannen Ihrer Knoten nach Patch-Updates.
- Führen Sie AWS Step Functions Zustandsmaschinen aus, um Aufgaben wie das Entfernen eines Knotens aus einer ELB-Umgebung, das Patchen des Knotens und das anschließende Hinzufügen des Knotens zur ELB-Umgebung auszuführen.
- Zielknoten, die offline sind, indem Sie eine AWS Ressourcengruppe als Ziel angeben.

> **Note**
>
> Ein oder mehrere Ziele für Wartungsfenster Run Command-Typ-Aufgaben müssen angegeben werden. Je nach Aufgabe sind Ziele für andere Aufgabentypen im Wartungsfenster (Automatisierung AWS Lambda, und AWS Step Functions) optional. Weitere

Informationen zur Ausführung von Aufgaben, die keine Ziele angeben, finden Sie unter Wartungsfenster-Tasks ohne Ziele registrieren.

EventBridge Unterstützung

Dieses Systems Manager Manager-Tool wird in den EventBridge Amazon-Regeln als Ereignistyp unterstützt. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge und Referenz: EventBridge Amazon-Ereignistypen und -muster für Systems Manager.

Inhalt

- Einrichten von Maintenance Windows
- Wartungsfenster mit der Konsole erstellen und verwalten
- Tutorials
- Verwendung von Pseudo-Parametern bei der Registrierung von Aufgaben im Wartungsfenster
- Wartungsfenster-Optionen für Planung und aktive Zeiträume
- Wartungsfenster-Tasks ohne Ziele registrieren
- Fehlerbehebung bei Wartungsfenstern

Einrichten von Maintenance Windows

Bevor Benutzer in Ihrem AWS-Konto Wartungsfensteraufgaben mit Maintenance Windows, einem Tool in AWS Systems Manager, planen und erstellen können, müssen ihnen die erforderlichen Berechtigungen erteilt werden. Darüber hinaus müssen Sie eine IAM-Servicerolle für Wartungsfenster und die IAM-Richtlinie erstellen, die an diese angehängt werden soll.

Bevor Sie beginnen

Zusätzlich zu den Berechtigungen, die Sie in diesem Abschnitt konfigurieren, sollten die IAM-Entitäten (Benutzer, Rollen oder Gruppen), die mit Wartungsfenstern arbeiten, bereits über allgemeine Wartungsfensterberechtigungen verfügen. Sie können diese Berechtigungen erteilen, indem Sie den Entitäten die IAM-Richtlinie `AmazonSSMFullAccess` zuweisen oder eine benutzerdefinierte IAM-Richtlinie zuweisen, die einen kleineren Satz von Zugriffsberechtigungen für Systems Manager bereitstellt, der Aufgaben des Wartungsfensters abdeckt.

Themen

- Steuern des Zugriffs auf Wartungsfenster mithilfe der Konsole
- Steuern Sie den Zugriff auf Wartungsfenster mithilfe der AWS CLI

Steuern des Zugriffs auf Wartungsfenster mithilfe der Konsole

Die folgenden Verfahren beschreiben, wie Sie die AWS Systems Manager Konsole verwenden, um die erforderlichen Berechtigungen und Rollen für Wartungsfenster zu erstellen.

Themen

- Aufgabe 1: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster-Aufgaben
- Aufgabe 2: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster mithilfe der Konsole
- Aufgabe 3: Bestimmten Benutzern die Berechtigung erteilen, Wartungsfensteraufgaben über die Konsole zu registrieren
- Aufgabe 4: Verhindern, dass bestimmte Benutzer Aufgaben im Wartungsfenster über die Konsole registrieren

Aufgabe 1: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster-Aufgaben

Wartungsfenster-Aufgaben erfordern eine IAM-Rolle, um die Berechtigungen bereitzustellen, die für die Ausführung für die Zielressourcen erforderlich sind. Die Berechtigungen werden durch eine IAM-Richtlinie bereitgestellt, die der Rolle angefügt wird. Die Arten von Aufgaben, die Sie ausführen, und Ihre anderen betrieblichen Anforderungen bestimmen den Inhalt dieser Richtlinie. Wir bieten eine Basisrichtlinie an, die Sie Ihren Bedürfnissen anpassen können. Abhängig von den Aufgaben und Arten von Aufgaben, die Ihre Wartungsfenster ausführen, benötigen Sie möglicherweise nicht alle Berechtigungen in dieser Richtlinie und müssen möglicherweise zusätzliche Berechtigungen einschließen. Sie hängen diese Richtlinie an die Rolle an, die Sie später in Aufgabe 2: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster mithilfe der Konsole erstellen.

Erstellen einer benutzerdefinierten Richtlinie mithilfe der Konsole

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Richtlinien und dann Richtlinie erstellen.
3. Wählen Sie im Abschnitt Richtlinien-Editor JSON aus.
4. Ersetzen Sie die Standardinhalte durch folgenden Inhalt:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:CancelCommand",
                "ssm:ListCommands",
                "ssm:ListCommandInvocations",
                "ssm:GetCommandInvocation",
                "ssm:GetAutomationExecution",
                "ssm:StartAutomationExecution",
                "ssm:ListTagsForResource",
                "ssm:DescribeInstanceInformation",
                "ssm:GetParameters"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "states:DescribeExecution",
                "states:StartExecution"
            ],
            "Resource": [
                "arn:aws:states:*:*:execution:*:*",
                "arn:aws:states:*:*:stateMachine:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda:*:*:function:*"
            ]
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "resource-groups:ListGroups",
                "resource-groups:ListGroupResources"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/maintenance-window-
role-name",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ssm.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

5. Ändern Sie den JSON-Inhalt nach Bedarf für die Wartungsaufgaben, die Sie in Ihrem Konto ausführen. Die Änderungen, die Sie vornehmen, beziehen sich auf Ihre geplanten Abläufe.

   Beispiel:

   - Sie können Amazon-Ressourcennamen (ARNs) für bestimmte Funktionen und Zustandsmaschinen angeben, anstatt Platzhalterkennungen (*) zu verwenden.
   - Wenn Sie nicht vorhaben, AWS Step Functions Aufgaben auszuführen, können Sie die `states` Berechtigungen und () ARNs entfernen.

- Wenn Sie nicht vorhaben, AWS Lambda Aufgaben auszuführen, können Sie die `lambda` Berechtigungen und entfernen ARNs.
- Wenn Sie keine Automatisierungs-Aufgaben ausführen möchten, können Sie die `ssm:GetAutomationExecution`- und `ssm:StartAutomationExecution`-Berechtigungen entfernen.
- Fügen Sie zusätzliche Berechtigungen hinzu, die möglicherweise für die Ausführung der Aufgaben erforderlich sind. Manche Automatisierungsaktionen basieren z. B. auf AWS CloudFormation -Stacks. Aus diesem Grund sind die Berechtigungen `cloudformation:CreateStack`, `cloudformation:DescribeStacks` und `cloudformation:DeleteStack` erforderlich.

  Als weiteres Beispiel benötigt das Automation-Runbook `AWS-CopySnapshot` Berechtigungen zum Erstellen eines Amazon Elastic Block Store (Amazon EBS)-Snapshots. Daher benötigt die Servicerolle die Berechtigung `ec2:CreateSnapshot`.

  Informationen zu den Rollenberechtigungen, die von Automation-Runbooks benötigt werden, finden Sie in den Runbook-Beschreibungen in der Referenz zum AWS Systems Manager - Automation-Runbook.

6. Nachdem Sie die Richtlinienüberarbeitungen abgeschlossen haben, wählen Sie Weiter: Tags.
7. Geben Sie für Richtlinienname einen Namen ein, der dies als Richtlinie identifiziert, die von der von Ihnen erstellten Servicerolle verwendet wird. Beispiel: **`my-maintenance-window-role-policy`**.
8. (Optional) Fügen Sie im Bereich Tags hinzufügen ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Richtlinie zu organisieren, zu verfolgen oder zu steuern.
9. Wählen Sie Richtlinie erstellen aus.

   Notieren Sie sich den Namen, den Sie für die Richtlinie angegeben haben. Sie beziehen sich im nächsten Verfahren, Aufgabe 2: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster mithilfe der Konsole, darauf.

Aufgabe 2: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster mithilfe der Konsole

Die Richtlinie, die Sie in der vorherigen Aufgabe erstellt haben, ist an die Wartungsfenster-Servicerolle angehängt, die Sie in dieser Aufgabe erstellen. Wenn Benutzer eine Wartungsfenster-Aufgabe registrieren, geben sie diese IAM-Rolle als Teil der Aufgabenkonfiguration an. Die

Berechtigungen in dieser Rolle ermöglichen es Systems Manager, Wartungsfenster-Aufgaben in Ihrem Namen auszuführen.

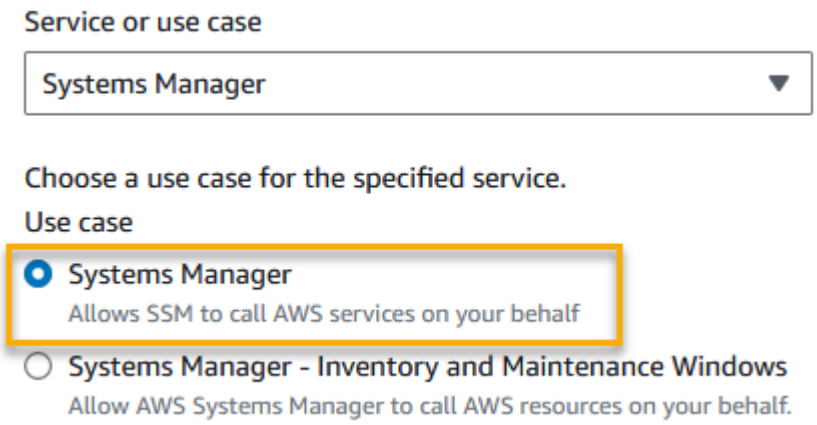
> **Important**
>
> Bisher bot Ihnen die Systems Manager Manager-Konsole die Möglichkeit, die AWS verwaltete, mit dem IAM-Dienst verknüpfte Rolle auszuwählen`AWSServiceRoleForAmazonSSM`, die Sie als Wartungsrolle für Ihre Aufgaben verwenden möchten. Die Verwendung dieser Rolle und der zugehörigen Richtlinie, `AmazonSSMServiceRolePolicy`, für Wartungsfenster-Aufgaben wird nicht mehr empfohlen. Wenn Sie diese Rolle jetzt für Wartungsfenster-Aufgaben verwenden, empfehlen wir Ihnen, sie nicht mehr zu verwenden. Erstellen Sie stattdessen Ihre eigene IAM-Rolle, die die Kommunikation zwischen Systems Manager und anderen ermöglicht, AWS-Services wenn Ihre Wartungsfensteraufgaben ausgeführt werden.

Verwenden Sie das folgende Verfahren, um eine benutzerdefinierte Servicerolle für Maintenance Windows zu erstellen, damit Systems Manager Maintenance Windows-Aufgaben in Ihrem Namen ausführen kann. Sie fügen die Richtlinie, die Sie in der vorherigen Aufgabe erstellt haben, an die von Ihnen erstellte benutzerdefinierte Servicerolle an.

Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster mithilfe der Konsole

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen und dann Rolle erstellen.
3. Wählen Sie für Select trusted entity (Vertrauenswürdige Entität auswählen) die folgenden Optionen:
    1. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option AWS -Service aus.
    2. Wählen Sie für Anwendungsfall Systems Manager aus
    3. Wählen Sie Systems Manager aus.

        In der folgenden Abbildung wird die Position der Systems-Manager-Option hervorgehoben.



4. Wählen Sie Weiter aus.
5. Geben Sie im Bereich Berechtigungsrichtlinien in das Suchfeld den Namen der Richtlinie ein, die Sie in Aufgabe 1: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster-Aufgaben erstellt haben, aktivieren Sie das Kontrollkästchen neben dem Namen und wählen Sie dann Weiter aus.
6. Geben Sie unter Role name (Rollenname) einen Namen ein, der diese Rolle als Maintenance Windows-Rolle identifiziert. Beispiel: **`my-maintenance-window-role`**.
7. (Optional) Ändern der Standardrollenbeschreibung, um den Zweck dieser Rolle anzuzeigen. Beispiel: **`Performs maintenance window tasks on your behalf`**.
8. Stellen Sie für Schritt 1: Vertrauenswürdige Entitäten auswählen sicher, dass die folgende Richtlinie im Feld Vertrauenswürdige Richtlinie angezeigt wird.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ssm.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

9. Stellen Sie für Schritt 2: Berechtigungen hinzufügen sicher, dass die Richtlinie, die Sie in Aufgabe 1: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster-Aufgaben erstellt haben, vorhanden ist.
10. (Optional) Fügen Sie in Schritt 3: Tags hinzufügen ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Rolle zu organisieren, zu verfolgen oder zu steuern.
11. Wählen Sie Create role (Rolle erstellen) aus. Das System leitet Sie zur Seite Rollen zurück.
12. Wählen Sie den Namen der IAM-Rolle aus, die Sie gerade erstellt haben.
13. Kopieren oder Notieren Sie sich Rollenname und ARN-Wert im Übersicht-Bereich. Benutzer in Ihrem Konto geben diese Informationen an, wenn sie Wartungsfenster erstellen.

Aufgabe 3: Bestimmten Benutzern die Berechtigung erteilen, Wartungsfensteraufgaben über die Konsole zu registrieren

Wenn Sie Benutzern Berechtigungen für den Zugriff auf die Servicerolle des benutzerdefinierten Wartungsfensters erteilen, können sie sie mit ihren Wartungsfenstern verwenden. Dies gilt zusätzlich zu den Berechtigungen, die Sie ihnen bereits zur Arbeit mit den Systems-Manager-API-Befehlen für das Tool Maintenance Windows erteilt haben. Diese IAM-Rolle vermittelt, dass Berechtigungen zum Ausführen einer Wartungsfenster-Aufgabe erforderlich sind. Infolgedessen kann ein Benutzer einem Wartungsfenster mithilfe Ihrer benutzerdefinierten Servicerolle keine Aufgaben registrieren, ohne diese IAM-Berechtigungen übergeben zu können.

Wenn Sie eine Aufgabe bei einem Wartungsfenster registrieren, geben Sie eine Servicerolle an, um die eigentlichen Aufgabenvorgänge auszuführen. Hierbei handelt es sich um die Rolle, die vom Service angenommen wird, wenn Aufgaben in Ihrem Namen ausgeführt werden. Um die Aufgabe selbst zu registrieren, weisen Sie zuvor die IAM-`PassRole`-Richtlinie einer IAM-Entität (z. B. einem Benutzer oder einer Gruppe) zu. Dadurch kann die IAM Entität als Teil der Registrierung dieser Aufgaben im Wartungsfenster die Rolle angeben, die beim Ausführen der Aufgaben verwendet werden soll. Weitere Informationen finden Sie unter Erteilen von Berechtigungen, mit denen ein Benutzer eine Rolle an einen AWS-Serviceübergeben kann im IAM-Benutzerhandbuch.

So konfigurieren Sie Berechtigungen, die es Benutzern ermöglichen, Wartungsfensteraufgaben zu registrieren

Wenn eine IAM-Entität (Benutzer, Rolle oder Gruppe) mit Administratorberechtigungen eingerichtet ist, hat der IAM-Benutzer oder die Rolle Zugriff auf Wartungsfenster. Für Entitäten ohne Administratorberechtigungen muss ein Administrator der IAM-Entität die folgenden Berechtigungen

gewähren. Dies sind die Mindestberechtigungen, die erforderlich sind, um Aufgaben in einem Wartungsfenster zu registrieren:

- Die von `AmazonSSMFullAccess` verwaltete Richtlinie oder eine Richtlinie, die vergleichbare Berechtigungen bereitstellt.
- Die folgenden `iam:PassRole`- und `iam:ListRoles`-Berechtigungen.

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/my-maintenance-window-
role"
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "arn:aws:iam::111122223333:role/"
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/
ssm.amazonaws.com/"
        }
    ]
}
```

*my-maintenance-window-role*steht für den Namen der benutzerdefinierten Wartungsfenster-Servicerolle, die Sie zuvor erstellt haben.

*account-id*steht für die ID Ihres AWS-Konto. Durch das Hinzufügen dieser Berechtigung für die Ressource `arn:aws:iam::`*account-id*`:role/` können Benutzer Kundenrollen in der Konsole anzeigen und auswählen, wenn sie eine Wartungsfensteraufgabe erstellen. Durch das Hinzufügen dieser Berechtigung für `arn:aws:iam::`*account-id*`:role/aws-service-role/ssm.amazonaws.com/` können Benutzer die mit dem Systems Manager-Service verknüpfte Rolle in der Konsole auswählen, wenn sie eine Wartungsfensteraufgabe erstellen.

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

  Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center -Benutzerhandbuch.

- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

  Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anleitung unter Eine Rolle für einen externen Identitätsanbieter (Verbund) erstellen im IAM-Benutzerhandbuch.

- IAM-Benutzer:
  - Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Befolgen Sie die Anleitung unter Eine Rolle für einen IAM-Benutzer erstellen im IAM-Benutzerhandbuch.
  - (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter Hinzufügen von Berechtigungen zu einem Benutzer (Konsole) im IAM-Benutzerhandbuch.

Konfigurieren von Berechtigungen für Gruppen, die Wartungsfensteraufgaben registrieren dürfen mithilfe der Konsole

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Klicken Sie im Navigationsbereich auf Groups oder Users.
3. Wählen Sie in der Gruppenliste den Namen der Gruppe aus, der Sie die `iam:PassRole`-Berechtigung zuweisen möchten, oder erstellen Sie ggf. zunächst eine neue Gruppe
4. Wählen Sie auf der Registerkarte Permissions (Berechtigungen) die Optionen Add permissions, Create Inline Policy (Berechtigungen hinzufügen, Inline-Richtlinie erstellen) aus.
5. Wählen Sie im Bereich Richtlinien-Editor JSON und ersetzen Sie den Standardinhalt des Felds durch Folgendes.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/my-maintenance-
window-role"
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "arn:aws:iam::111122223333:role/"
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/
ssm.amazonaws.com/"
        }
    ]
}
```

*my-maintenance-window-role*steht für den Namen der benutzerdefinierten Rolle im Wartungsfenster, die Sie zuvor erstellt haben.

*account-id*steht für die ID Ihres AWS-Konto. Durch das Hinzufügen dieser Berechtigung für die Ressource `arn:aws:iam::`*account-id*`:role/` können Benutzer Kundenrollen in der Konsole anzeigen und auswählen, wenn sie eine Wartungsfensteraufgabe erstellen. Durch das Hinzufügen dieser Berechtigung für `arn:aws:iam::`*account-id*`:role/aws-service-role/ssm.amazonaws.com/` können Benutzer die mit dem Systems Manager-Service verknüpfte Rolle in der Konsole auswählen, wenn sie eine Wartungsfensteraufgabe erstellen.

6. Wählen Sie Weiter aus.
7. Geben Sie auf der Seite Überprüfen und erstellen einen Namen in das Feld Richtlinienname ein, um diese `PassRole`-Richtlinie zu identifizieren (beispielsweise **`my-group-iam-passrole-policy`**) und wählen Sie dann Richtlinie erstellen aus.

Aufgabe 4: Verhindern, dass bestimmte Benutzer Aufgaben im Wartungsfenster über die Konsole registrieren

Sie können den Benutzern in Ihrem Bereich, die Sie nicht möchten AWS-Konto , die `ssm:RegisterTaskWithMaintenanceWindow` Erlaubnis verweigern, Aufgaben in

Wartungsfenstern zu registrieren. Dies bietet eine zusätzliche Verhinderungsebene für Benutzer, die keine Wartungsfenster-Aufgaben registrieren sollten.

Konfigurieren von Berechtigungen für Gruppen, die keine Wartungsfensteraufgaben registrieren dürfen mithilfe der Konsole

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Klicken Sie im Navigationsbereich auf Groups oder Users.
3. Wählen Sie in der Gruppenliste den Namen der Gruppe aus, der Sie die `ssm:RegisterTaskWithMaintenanceWindow`-Berechtigung verweigern möchten, oder erstellen Sie ggf. zunächst eine neue Gruppe.
4. Wählen Sie auf der Registerkarte Permissions (Berechtigungen) die Optionen Add permissions, Create Inline Policy (Berechtigungen hinzufügen, Inline-Richtlinie erstellen) aus.
5. Wählen Sie im Bereich Richtlinien-Editor JSON und ersetzen Sie dann den Standardinhalt des Felds durch Folgendes.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ssm:RegisterTaskWithMaintenanceWindow",
            "Resource": "*"
        }
    ]
}
```

6. Wählen Sie Weiter aus.
7. Geben Sie auf der Seite Überprüfen und Erstellen bei Richtlinienname einen Namen ein, um diese Richtlinie zu identifizieren (beispielsweise **`my-groups-deny-mw-tasks-policy`**) und wählen Sie dann Richtlinie erstellen aus.

Steuern Sie den Zugriff auf Wartungsfenster mithilfe der AWS CLI

In den folgenden Verfahren wird beschrieben, wie Sie das AWS Command Line Interface (AWS CLI) verwenden, um die erforderlichen Berechtigungen und Rollen für Maintenance Windows ein Tool in zu erstellen AWS Systems Manager.

Themen

- Aufgabe 1: Vertrauensrichtlinien und vom Kunden verwaltete Richtliniendateien im JSON-Format erstellen
- Aufgabe 2: Erstellen und verifizieren Sie eine benutzerdefinierte Servicerolle für Wartungsfenster mithilfe der AWS CLI
- Aufgabe 3: Bestimmten Benutzern die Berechtigung erteilen, Wartungsfensteraufgaben über die AWS CLI zu registrieren
- Aufgabe 4: Verhindern Sie, dass bestimmte Benutzer Aufgaben im Wartungsfenster registrieren, indem Sie AWS CLI

Aufgabe 1: Vertrauensrichtlinien und vom Kunden verwaltete Richtliniendateien im JSON-Format erstellen

Wartungsfenster-Aufgaben erfordern eine IAM-Rolle, um die Berechtigungen bereitzustellen, die für die Ausführung für die Zielressourcen erforderlich sind. Die Berechtigungen werden durch eine IAM-Richtlinie bereitgestellt, die der Rolle angefügt wird. Die Arten von Aufgaben, die Sie ausführen, und Ihre anderen betrieblichen Anforderungen bestimmen den Inhalt dieser Richtlinie. Wir bieten eine Basisrichtlinie an, die Sie Ihren Bedürfnissen anpassen können. Abhängig von den Aufgaben und Arten von Aufgaben, die Ihre Wartungsfenster ausführen, benötigen Sie möglicherweise nicht alle Berechtigungen in dieser Richtlinie und müssen möglicherweise zusätzliche Berechtigungen einschließen.

In dieser Aufgabe geben Sie die Berechtigungen, die für Ihre benutzerdefinierte Rolle im Wartungsfenster erforderlich sind, in einem Paar von JSON-Dateien an. Sie hängen diese Richtlinie an die Rolle an, die Sie später in Aufgabe 2: Erstellen und verifizieren Sie eine benutzerdefinierte Servicerolle für Wartungsfenster mithilfe der AWS CLI erstellen.

Um Vertrauensrichtlinien und vom Kunden verwaltete Richtliniendateien zu erstellen

1. Kopieren Sie die folgende Vertrauensrichtlinie in eine Textdatei. Speichen Sie diese Datei mit folgendem Namen und folgender Dateierweiterung: **`mw-role-trust-policy.json`**.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

2. Kopieren Sie die folgende JSON-Richtlinie und fügen Sie sie in eine andere Textdatei ein. Speichern Sie diese Datei in demselben Verzeichnis, in dem Sie die erste Datei erstellt haben, mit dem folgenden Namen und der folgenden Dateierweiterung: **`mw-role-custom-policy.json`**.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:CancelCommand",
                "ssm:ListCommands",
                "ssm:ListCommandInvocations",
                "ssm:GetCommandInvocation",
                "ssm:GetAutomationExecution",
                "ssm:StartAutomationExecution",
                "ssm:ListTagsForResource",
                "ssm:GetParameters"
            ],
            "Resource": "*"
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "states:DescribeExecution",
                "states:StartExecution"
            ],
            "Resource": [
                "arn:aws:states:*:*:execution:*:*",
                "arn:aws:states:*:*:stateMachine:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda:*:*:function:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "resource-groups:ListGroups",
                "resource-groups:ListGroupResources"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/maintenance-window-
role-name",
```

```
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ssm.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

3. Ändern Sie den Inhalt von `mw-role-custom-policy.json` nach Bedarf für die Wartungsaufgaben, die Sie in Ihrem Konto ausführen. Die Änderungen, die Sie vornehmen, beziehen sich auf Ihre geplanten Abläufe.

   Beispiel:

   - Sie können Amazon-Ressourcennamen (ARNs) für bestimmte Funktionen und Zustandsmaschinen angeben, anstatt Platzhalterkennungen (*) zu verwenden.
   - Wenn Sie nicht vorhaben, AWS Step Functions Aufgaben auszuführen, können Sie die `states` Berechtigungen und () ARNs entfernen.
   - Wenn Sie nicht vorhaben, AWS Lambda Aufgaben auszuführen, können Sie die `lambda` Berechtigungen und entfernen ARNs.
   - Wenn Sie keine Automatisierungs-Aufgaben ausführen möchten, können Sie die `ssm:GetAutomationExecution`- und `ssm:StartAutomationExecution`-Berechtigungen entfernen.
   - Fügen Sie zusätzliche Berechtigungen hinzu, die möglicherweise für die Ausführung der Aufgaben erforderlich sind. Manche Automatisierungsaktionen basieren z. B. auf AWS CloudFormation -Stacks. Aus diesem Grund sind die Berechtigungen `cloudformation:CreateStack`, `cloudformation:DescribeStacks` und `cloudformation:DeleteStack` erforderlich.

     Als weiteres Beispiel benötigt das Automation-Runbook `AWS-CopySnapshot` Berechtigungen zum Erstellen eines Amazon Elastic Block Store (Amazon EBS)-Snapshots. Daher benötigt die Servicerolle die Berechtigung `ec2:CreateSnapshot`.

     Informationen zu den Rollenberechtigungen, die von Automation-Runbooks benötigt werden, finden Sie in den Runbook-Beschreibungen in der Referenz zum AWS Systems Manager - Automation-Runbook.

Speichern Sie die Datei erneut, nachdem Sie alle erforderlichen Änderungen vorgenommen haben.

Aufgabe 2: Erstellen und verifizieren Sie eine benutzerdefinierte Servicerolle für Wartungsfenster mithilfe der AWS CLI

Die Richtlinie, die Sie in der vorherigen Aufgabe erstellt haben, ist an die Wartungsfenster-Servicerolle angehängt, die Sie in dieser Aufgabe erstellen. Wenn Benutzer eine Wartungsfenster-Aufgabe registrieren, geben sie diese IAM-Rolle als Teil der Aufgabenkonfiguration an. Die Berechtigungen in dieser Rolle ermöglichen es Systems Manager, Wartungsfenster-Aufgaben in Ihrem Namen auszuführen.

> **Important**
>
> Bisher bot Ihnen die Systems Manager Manager-Konsole die Möglichkeit, die AWS verwaltete, mit dem IAM-Dienst verknüpfte Rolle auszuwählen`AWSServiceRoleForAmazonSSM`, die Sie als Wartungsrolle für Ihre Aufgaben verwenden möchten. Die Verwendung dieser Rolle und der zugehörigen Richtlinie, `AmazonSSMServiceRolePolicy`, für Wartungsfenster-Aufgaben wird nicht mehr empfohlen. Wenn Sie diese Rolle jetzt für Wartungsfenster-Aufgaben verwenden, empfehlen wir Ihnen, sie nicht mehr zu verwenden. Erstellen Sie stattdessen Ihre eigene IAM-Rolle, die die Kommunikation zwischen Systems Manager und anderen ermöglicht, AWS-Services wenn Ihre Wartungsfensteraufgaben ausgeführt werden.

In dieser Aufgabe führen Sie CLI-Befehle aus, um Ihre Windows-Wartungsservicerolle zu erstellen, und fügen dabei den Richtlinieninhalt aus den von Ihnen erstellten JSON-Dateien hinzu.

Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster mithilfe der AWS CLI

1. Öffnen Sie das AWS CLI und führen Sie den folgenden Befehl in dem Verzeichnis aus, in dem Sie `mw-role-custom-policy.json` und `mw-role-trust-policy.json` platziert haben. Der Befehl erstellt eine Wartungsfenster-Servicerolle namens `my-maintenance-window-role` und fügt ihr die Vertrauensrichtlinie hinzu.

Linux & macOS

```
aws iam create-role \
    --role-name "my-maintenance-window-role" \
    --assume-role-policy-document file://mw-role-trust-policy.json
```

Windows

```
aws iam create-role ^
    --role-name "my-maintenance-window-role" ^
    --assume-role-policy-document file://mw-role-trust-policy.json
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": "sts:AssumeRole",
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "ssm.amazonaws.com"
                    }
                }
            ]
        },
        "RoleId": "AROAIIZKPBKS2LEXAMPLE",
        "CreateDate": "2024-08-19T03:40:17.373Z",
        "RoleName": "my-maintenance-window-role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/my-maintenance-window-role"
    }
}
```

> **Note**
>
> Notieren Sie sich die Werte für `RoleName` und `Arn`. Sie brauchen diese Informationen im nächsten Befehl.

2. Führen Sie den folgenden Befehl aus, um der Rolle die vom Kunden verwaltete Richtlinie anzufügen. Ersetzen Sie den *`account-id`* Platzhalter durch Ihre eigene ID AWS-Konto

   Linux & macOS

   ```
   aws iam attach-role-policy \
       --role-name "my-maintenance-window-role" \
       --policy-arn "arn:aws:iam::account-id:policy/mw-role-custom-policy.json"
   ```

   Windows

   ```
   aws iam attach-role-policy ^
       --role-name "my-maintenance-window-role" ^
       --policy-arn "arn:aws:iam::account-id:policy/mw-role-custom-policy.json"
   ```

3. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob Ihre Rolle erstellt wurde und ob die Vertrauensrichtlinie angefügt wurde.

   ```
   aws iam get-role --role-name my-maintenance-window-role
   ```

   Die vom System zurückgegebenen Informationen ähneln den Folgenden:

   ```
   {
       "Role": {
           "Path": "/",
           "RoleName": "my-maintenance-window-role",
           "RoleId": "AROA123456789EXAMPLE",
           "Arn": "arn:aws:iam::123456789012:role/my-maintenance-window-role",
           "CreateDate": "2024-08-19T14:13:32+00:00",
           "AssumeRolePolicyDocument": {
               "Version": "2012-10-17",
               "Statement": [
                   {
                       "Effect": "Allow",
                       "Principal": {
   ```

```
                        "Service": "ssm.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        },
        "MaxSessionDuration": 3600,
        "RoleLastUsed": {
            "LastUsedDate": "2024-08-19T14:30:44+00:00",
            "Region": "us-east-2"
        }
    }
}
```

4. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob die vom Kunden verwaltete Richtlinie der Rolle angefügt wurde.

```
aws iam list-attached-role-policies --role-name my-maintenance-window-role
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden:

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "mw-role-custom-policy",
            "PolicyArn": "arn:aws:iam::123456789012:policy/mw-role-custom-policy"
        }
    ]
}
```

Aufgabe 3: Bestimmten Benutzern die Berechtigung erteilen, Wartungsfensteraufgaben über die AWS CLI zu registrieren

Wenn Sie Benutzern Berechtigungen für den Zugriff auf die Servicerolle des benutzerdefinierten Wartungsfensters erteilen, können sie sie mit ihren Wartungsfenstern verwenden. Dies gilt zusätzlich zu den Berechtigungen, die Sie ihnen bereits zur Arbeit mit den Systems-Manager-API-Befehlen für das Tool Maintenance Windows erteilt haben. Diese IAM-Rolle vermittelt, dass Berechtigungen zum Ausführen einer Wartungsfenster-Aufgabe erforderlich sind. Infolgedessen kann ein Benutzer einem Wartungsfenster mithilfe Ihrer benutzerdefinierten Servicerolle keine Aufgaben registrieren, ohne diese IAM-Berechtigungen übergeben zu können.

Wenn Sie eine Aufgabe bei einem Wartungsfenster registrieren, geben Sie eine Servicerolle an, um die eigentlichen Aufgabenvorgänge auszuführen. Hierbei handelt es sich um die Rolle, die vom Service angenommen wird, wenn Aufgaben in Ihrem Namen ausgeführt werden. Um die Aufgabe selbst zu registrieren, weisen Sie zuvor die IAM-`PassRole`-Richtlinie einer IAM-Entität (z. B. einem Benutzer oder einer Gruppe) zu. Dadurch kann die IAM Entität als Teil der Registrierung dieser Aufgaben im Wartungsfenster die Rolle angeben, die beim Ausführen der Aufgaben verwendet werden soll. Weitere Informationen finden Sie unter Erteilen von Berechtigungen, mit denen ein Benutzer eine Rolle an einen AWS-Serviceübergeben kann im IAM-Benutzerhandbuch.

Um Berechtigungen für Benutzer zu konfigurieren, die Wartungsfensteraufgaben registrieren dürfen, verwenden Sie den AWS CLI

1. Kopieren Sie die folgende AWS Identity and Access Management (IAM-) Richtlinie, fügen Sie sie in einen Texteditor ein und speichern Sie sie mit dem folgenden Namen und der folgenden Dateierweiterung:`mw-passrole-policy.json`.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/my-maintenance-
window-role"
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "arn:aws:iam::111122223333:role/"
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/
ssm.amazonaws.com/"
        }
    ]
}
```

*my-maintenance-window-role*Ersetzen Sie es durch den Namen der benutzerdefinierten Rolle im Wartungsfenster, die Sie zuvor erstellt haben.

*account-id*Ersetzen Sie durch die ID Ihres AWS-Konto. Wenn Sie diese Berechtigung für die Ressource `arn:aws:iam::`*account-id*`:role/` hinzufügen, können Benutzer in der Gruppe Kundenrollen in der Konsole anzeigen und auswählen, wenn sie eine Wartungsfensteraufgabe erstellen. Wenn Sie diese Berechtigung für `arn:aws:iam::`*account-id*`:role/aws-service-role/ssm.amazonaws.com/` hinzufügen, können Benutzer in der Gruppe die mit dem Systems Manager-Service verknüpfte Rolle in der Konsole auswählen, wenn sie eine Wartungsfensteraufgabe erstellen.

2. Öffne das AWS CLI.
3. Je nachdem, ob Sie die Berechtigung einer IAM-Entität (Benutzer oder Gruppe) zuweisen, führen Sie einen der folgenden Befehle aus.

   - Für eine IAM-Entität:

     Linux & macOS

     ```
     aws iam put-user-policy \
         --user-name "user-name" \
         --policy-name "policy-name" \
         --policy-document file://path-to-document
     ```

     Windows

     ```
     aws iam put-user-policy ^
         --user-name "user-name" ^
         --policy-name "policy-name" ^
         --policy-document file://path-to-document
     ```

     Geben Sie für *user-name* den Benutzer an, der Wartungsfenstern Aufgaben zuweist. Geben Sie für den Namen an*policy-name*, den Sie zur Identifizierung der Richtlinie verwenden möchten, z. B. **my-iam-passrole-policy** Geben Sie als *path-to-document* den Pfad zur in Schritt 1 gespeicherten Datei an. Beispiel: `file://C:\Temp\mw-passrole-policy.json`

> **Note**
>
> Um einem Benutzer Zugriff zum Registrieren von Aufgaben für Wartungsfenster über die Systems-Manager-Konsole zu gewähren, müssen Sie auch die Richtlinie `AmazonSSMFullAccess` Ihrem Benutzer zuweisen (oder eine IAM-Richtlinie, die eine kleinere Gruppe von Zugriffsberechtigungen für Systems Manager bereitstellt, die die Aufgaben im Wartungsfenster abdeckt). Führen Sie den folgenden Befehl aus, um Ihrem Benutzer die Richtlinie `AmazonSSMFullAccess` zuzuweisen.
>
> Linux & macOS
>
> ```
> aws iam attach-user-policy \
>     --policy-arn "arn:aws:iam::aws:policy/AmazonSSMFullAccess" \
>     --user-name "user-name"
> ```
>
> Windows
>
> ```
> aws iam attach-user-policy ^
>     --policy-arn "arn:aws:iam::aws:policy/AmazonSSMFullAccess" ^
>     --user-name "user-name"
> ```

- Für eine IAM-Gruppe:

  Linux & macOS

  ```
  aws iam put-group-policy \
      --group-name "group-name" \
      --policy-name "policy-name" \
      --policy-document file://path-to-document
  ```

  Windows

  ```
  aws iam put-group-policy ^
      --group-name "group-name" ^
      --policy-name "policy-name" ^
      --policy-document file://path-to-document
  ```

  Geben Sie für *group-name* die Gruppe an, deren Mitglieder Wartungsfenstern Aufgaben zuweisen. Geben Sie für den Namen an*policy-name*, den Sie zur Identifizierung der

Richtlinie verwenden möchten, z. **`my-iam-passrole-policy`** B. Geben Sie als *`path-to-document`* den Pfad zur in Schritt 1 gespeicherten Datei an. Zum Beispiel: `file://C:\Temp\mw-passrole-policy.json`

> **Note**
>
> Um Mitgliedern einer Gruppe Zugriff zum Registrieren von Aufgaben für Wartungsfenster über die Systems Manager-Konsole zu gewähren, müssen Sie die Richtlinie `AmazonSSMFullAccess` auch Ihrer Gruppe zuweisen. Führen Sie den folgenden Befehl aus, um Ihrer Gruppe diese Richtlinie zuzuweisen.
>
> Linux & macOS
>
> ```
> aws iam attach-group-policy \
>     --policy-arn "arn:aws:iam::aws:policy/AmazonSSMFullAccess" \
>     --group-name "group-name"
> ```
>
> Windows
>
> ```
> aws iam attach-group-policy ^
>     --policy-arn "arn:aws:iam::aws:policy/AmazonSSMFullAccess" ^
>     --group-name "group-name"
> ```

4. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob die Richtlinie der Gruppe zugewiesen wurde.

   Linux & macOS

   ```
   aws iam list-group-policies \
       --group-name "group-name"
   ```

   Windows

   ```
   aws iam list-group-policies ^
       --group-name "group-name"
   ```

Aufgabe 4: Verhindern Sie, dass bestimmte Benutzer Aufgaben im Wartungsfenster registrieren, indem Sie AWS CLI

Sie können Benutzern in Ihrem Umfeld, die Sie nicht möchten AWS-Konto , die `ssm:RegisterTaskWithMaintenanceWindow` Erlaubnis verweigern, Aufgaben in Wartungsfenstern zu registrieren. Dies bietet eine zusätzliche Verhinderungsebene für Benutzer, die keine Wartungsfenster-Aufgaben registrieren sollten.

Je nachdem, ob Sie die `ssm:RegisterTaskWithMaintenanceWindow`-Berechtigung für einen einzelnen Benutzer oder eine Gruppe verweigern, verwenden Sie eines der folgenden Verfahren, um zu verhindern, dass Benutzer Aufgaben mit einem Wartungsfenster registrieren können.

Um Berechtigungen für Benutzer zu konfigurieren, die keine Wartungsfensteraufgaben registrieren dürfen, verwenden Sie den AWS CLI

1. Kopieren Sie die folgende IAM-Richtlinie und fügen Sie sie in einen Text-Editor ein und speichern Sie sie mit dem folgenden Namen und Dateierweiterung: **`deny-mw-tasks-policy.json`**.

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": [
           {
               "Effect": "Deny",
               "Action": "ssm:RegisterTaskWithMaintenanceWindow",
               "Resource": "*"
           }
       ]
   }
   ```

2. Öffnen Sie das AWS CLI.
3. Je nachdem, ob Sie die Berechtigung einer IAM-Entität (Benutzer oder Gruppe) zuweisen, führen Sie einen der folgenden Befehle aus.

   - Für einen Benutzer:

     Linux & macOS

     ```
     aws iam put-user-policy \
         --user-name "user-name" \
     ```

```
    --policy-name "policy-name" \
    --policy-document file://path-to-document
```

Windows

```
aws iam put-user-policy ^
    --user-name "user-name" ^
    --policy-name "policy-name" ^
    --policy-document file://path-to-document
```

Geben Sie für den Benutzer an*user-name*, der verhindern soll, dass Wartungsfenstern Aufgaben zugewiesen werden. Geben Sie für den Namen an*policy-name*, den Sie zur Identifizierung der Richtlinie verwenden möchten, z. B. **my-deny-mw-tasks-policy** Geben Sie als *path-to-document* den Pfad zur in Schritt 1 gespeicherten Datei an. Beispiel: `file://C:\Temp\deny-mw-tasks-policy.json`

- Für eine Gruppe:

  Linux & macOS

```
aws iam put-group-policy \
    --group-name "group-name" \
    --policy-name "policy-name" \
    --policy-document file://path-to-document
```

  Windows

```
aws iam put-group-policy ^
    --group-name "group-name" ^
    --policy-name "policy-name" ^
    --policy-document file://path-to-document
```

  Geben Sie für die Gruppe an*group-name*, die daran gehindert werden soll, Wartungsfenstern Aufgaben zuzuweisen. Geben Sie für den Namen an*policy-name*, den Sie zur Identifizierung der Richtlinie verwenden möchten, z. B. **my-deny-mw-tasks-policy** Geben Sie als *path-to-document* den Pfad zur in Schritt 1 gespeicherten Datei an. Zum Beispiel: `file://C:\Temp\deny-mw-tasks-policy.json`

4. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob die Richtlinie der Gruppe zugewiesen wurde.

Linux & macOS

```
aws iam list-group-policies \
    --group-name "group-name"
```

Windows

```
aws iam list-group-policies ^
    --group-name "group-name"
```

Wartungsfenster mit der Konsole erstellen und verwalten

In diesem Abschnitt wird beschrieben, wie Sie Wartungsfenster mithilfe der AWS Systems Manager-Konsole erstellen, konfigurieren, aktualisieren und löschen. Dieser Abschnitt enthält auch Informationen zum Verwalten der Ziele und Aufgaben eines Wartungsfensters.

> **Important**
>
> Wir empfehlen, dass Sie Wartungsfenster anfänglich in einer Testumgebung erstellen und konfigurieren.

Bevor Sie beginnen

Bevor Sie ein Wartungsfenster erstellen, müssen Sie den Zugriff auf Maintenance Windows, ein Tool in AWS Systems Manager, konfigurieren. Weitere Informationen finden Sie unter Einrichten von Maintenance Windows.

Themen

- Erstellen eines Wartungsfensters mit der Konsole
- Ziele zu einem Wartungsfenster mit der Konsole zuweisen
- Aufgaben zu einem Wartungsfenster mit der Konsole zuweisen
- Deaktivieren oder Aktivieren eines Wartungsfensters mithilfe der Konsole
- Ressourcen für das Wartungsfenster mithilfe der Konsole aktualisieren oder löschen

Erstellen eines Wartungsfensters mit der Konsole

In diesem Verfahren erstellen Sie ein Wartungsfenster in Maintenance Windows, einem Tool in AWS Systems Manager. Sie können die grundlegenden Optionen, wie Name, Zeitplan und Dauer, festlegen. In späteren Schritten wählen Sie die Ziele oder Ressourcen aus, die damit aktualisiert werden sollen, sowie die Aufgaben, die während der Ausführung des Wartungsfensters ausgeführt werden.

> **Note**
>
> Eine Beschreibung des Verhältnisses zwischen den verschiedenen zeitplanbezogenen Optionen für Wartungsfenster finden Sie unter Wartungsfenster-Optionen für Planung und aktive Zeiträume.
> Weitere Informationen zum Arbeiten mit der `--schedule`-Option finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

So erstellen Sie ein Wartungsfenster mit der Konsole

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie Create maintenance window (Wartungsfenster erstellen) aus.
4. Geben Sie im Feld Name einen aussagekräftigen Namen ein, an dem Sie dieses Wartungsfenster erkennen können.
5. (Optional) Geben Sie unter Description (Beschreibung) eine Beschreibung ein, um anzugeben, wie dieses Wartungsfenster verwendet werden soll.
6. Wenn eine Wartungsfenster-Aufgabe auf verwalteten Knoten ausgeführt werden soll, obwohl diese Knoten nicht als Ziele registriert wurden, wählen Sie Allow unregistered targets (Nicht registrierte Ziele erlauben) aus.

   Falls Sie diese Option wählen, können Sie die nicht registrierten Knoten (nach Knoten-ID) auswählen, wenn Sie eine Aufgabe für das Wartungsfenster registrieren.

   Sollten Sie diese Option nicht wählen, müssen Sie die zuvor registrierten Ziele auswählen, wenn Sie eine Aufgabe für das Wartungsfenster registrieren.
7. Geben Sie mithilfe einer der drei Planungsoptionen einen Zeitplan für das Wartungsfenster an.

Hinweise zum Erstellen von cron/rate Ausdrücken finden Sie unterReferenz: Cron- und Rate-Ausdrücke für System Manager.

8. Geben Sie unter Duration (Dauer) die Anzahl der Stunden ein, die das Wartungsfenster ausgeführt wird. Der Wert, den Sie angeben, bestimmt die spezifische Endzeit für das Wartungsfenster basierend auf dem Zeitpunkt, an dem es beginnt. Nach der resultierenden Endzeit dürfen keine Wartungsfenster-Aufgaben gestartet werden, abzüglich der Anzahl der Stunden, die Sie für Stop initiating tasks (Initiieren von Aufgaben beenden) im nächsten Schritt angeben.

   Beispiel: Wenn das Wartungsfenster um 15:00 Uhr beginnt, die Dauer drei Stunden beträgt und der Wert Stop initiating tasks (Initiieren von Aufgaben beenden) eine Stunde beträgt, können nach 17:00 Uhr keine Wartungsfenster-Aufgaben gestartet werden.

9. Geben Sie unter Stop initiating tasks (Initiieren von Aufgaben beenden) die Anzahl der Stunden für den Zeitpunkt vor dem Ende des Wartungsfensters an, ab dem vom System keine neuen auszuführenden Aufgaben mehr geplant werden sollen.

10. (Optional) Geben Sie unter Window start date (Startzeit des Fensters) ein Datum und eine Uhrzeit im erweiterten ISO-8601-Format an, zu dem bzw. der das Wartungsfenster aktiviert werden soll. Auf diese Weise können Sie die Aktivierung des Wartungsfensters bis zum angegebenen künftigen Zeitpunkt verzögern.

    > **Note**
    >
    > Sie können kein Startdatum und keine Startzeit angeben, die in der Vergangenheit liegen.

11. (Optional) Geben Sie unter Window end date (Enddatum des Fensters) ein Datum und eine Uhrzeit im erweiterten ISO-8601-Format an, zu dem bzw. der das Wartungsfenster deaktiviert werden soll. Auf diese Weise können Sie ein in der Zukunft liegendes Datum sowie eine Uhrzeit festlegen, nach dem das Wartungsfenster nicht mehr ausgeführt wird.

12. (Optional) Geben Sie unter Schedule time zone (Zeitzone des Zeitplans) die Zeitzone im IANA-Format (Internet Assigned Numbers Authority) an, die als Grundlage für die Ausführung der geplanten Wartungsfenster verwendet werden soll. Zum Beispiel: "America/Los_Angeles", "etc/UTC", or "Asia/Seoul“.

    Weitere Informationen zu gültigen Formaten finden Sie unter Time Zone Database (Zeitzonendatenbank) auf der IANA-Website.

13. (Optional) Geben Sie unter Schedule offset (Zeitplanversatz) die Anzahl der Tage an, die nach dem durch einen Cron- oder Rate-Ausdruck angegebenen Datum und der angegebenen Uhrzeit gewartet werden soll, bevor das Wartungsfenster ausgeführt wird. Sie können ein bis sechs Tage angeben.

    > **Note**
    >
    > Diese Option ist nur verfügbar, wenn Sie einen Zeitplan durch manuelle Eingabe eines Cron- oder Rate-Ausdrucks angegeben haben.

14. (Optional) Wenden Sie im Bereich „Tags verwalten“ ein oder mehrere name/value Tag-Schlüsselpaare auf das Wartungsfenster an.

    Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Sie können beispielsweise ein Wartungsfenster mit Tags versehen, um die Aufgabentypen, die darüber ausgeführt werden, die Arten der Ziele sowie die Umgebung, in der es ausgeführt wird, zu identifizieren. In diesem Fall könnten Sie die folgenden name/value Schlüsselpaare angeben:

    - `Key=TaskType,Value=AgentUpdate`
    - `Key=OS,Value=Windows`
    - `Key=Environment,Value=Production`

15. Wählen Sie Create maintenance window (Wartungsfenster erstellen) aus. Das System leitet Sie zur Seite „Maintenance Window“ (Wartungsfenster) zurück. Der Status des soeben erstellten Wartungsfensters lautet Enabled (Aktiviert).

Ziele zu einem Wartungsfenster mit der Konsole zuweisen

In diesem Verfahren registrieren Sie ein Ziel für ein Wartungsfenster. Mit anderen Worten: Geben Sie an, für welche Ressourcen das Wartungsfenster Aktionen durchführt.

> **Note**
>
> Wenn eine einzelne Wartungsfenster-Aufgabe mit mehreren Zielen registriert ist, werden ihre Aufrufe sequenziell und nicht parallel ausgeführt. Wenn Ihre Aufgabe gleichzeitig auf

mehreren Zielen ausgeführt werden muss, registrieren Sie eine Aufgabe für jedes Ziel einzeln, und weisen Sie jeder Aufgabe dieselbe Prioritätsstufe zu.

So weisen Sie Ziele zu einem Wartungsfenster mit der Konsole zu

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie in der Wartungsfensterliste die Wartungsfenster aus, dem Ziele hinzugefügt werden sollen.
4. Wählen Sie Actions (Aktionen) und anschließend Register targets (Ziele registrieren) aus.
5. (Optional) Geben Sie im Feld Target Name (Zielname) einen Namen für die Ziele ein.
6. (Optional) Geben Sie unter Description (Beschreibung) eine Beschreibung ein.
7. (Optional) Geben Sie für Eigentümerinformationen Informationen an, die in jedes EventBridge Amazon-Ereignis aufgenommen werden sollen, das während der Ausführung von Aufgaben für diese Ziele in diesem Wartungsfenster ausgelöst wird.

   Informationen EventBridge zur Überwachung von Systems Manager Manager-Ereignissen finden Sie unterÜberwachung von Systems Manager-Ereignissen mit Amazon EventBridge.
8. Wählen Sie im Bereich Targets (Ziele) eine der in der folgenden Tabelle beschriebenen Optionen.

Option	Description
Specify instance tags (Instance-Tags angeben)	Geben Sie unter Specify instance tags (Instance-Tags angeben) einen oder mehrere Tag-Schlüssel und (optional) Werte an, die den verwalteten Knoten in Ihrem Konto hinzugefügt wurden oder werden. Wenn das Wartungsfenster ausgeführt wird, versucht das Programm, Aufgaben auf allen verwaltet en Knoten auszuführen, denen diese Tags hinzugefügt wurden.

Option	Description
	Wenn Sie mehr als einen Tag-Schlüssel angeben, muss ein Knoten mit allen Tag-Schlüsseln und -Werten markiert werden, die Sie für die Aufnahme in die Zielgruppe angeben.
Choose instances manually (Instances manuell auswählen)	Aktivieren Sie in der Liste das Kontrollk ästchen für jeden Knoten, den Sie für das Wartungsfenster-Ziel aufnehmen möchten. Die Liste enthält alle Knoten in Ihrem Konto, die für die Verwendung mit Systems Manager konfiguriert sind. Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembe handlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung. Informationen zu Edge-Geräten und lokalen Servern und virtuellen Maschinen (VMs) finden Sie unter Verwalten von Knoten in Hybrid- und Multi-Cloud-Umgebungen mit Systems Manager

Option	Description
Eine Ressourcengruppe auswählen	Wählen Sie für Resource group (Ressourcengruppe) den Namen einer vorhandenen Ressourcengruppe in Ihrem Konto aus der Liste aus. Weitere Informationen zum Erstellen von und Arbeiten mit Ressourcengruppen finden Sie unter den folgenden Themen: • Was sind Ressourcengruppen? im AWS -Ressourcengruppen -Benutzerhandbuch • Ressourcengruppen und Tagging für AWS im AWS News Blog (Optional) Wählen Sie unter Resource types (Ressourcentypen) bis zu fünf verfügbare Ressourcentypen aus oder wählen Sie All resource types (Alle Ressourcentypen) aus. Wenn die Aufgaben, die Sie dem Wartungsfenster zugeordnet haben, für einen der dem Ziel hinzugefügten Ressourcentypen nicht für geeignet sind, meldet das System möglicherweise einen Fehler. Auch wenn ein solcher Fehler gemeldet wird, werden Aufgaben, für die ein unterstützter Ressourcentyp gefunden wurde, dennoch ausgeführt. Nehmen Sie beispielsweise an, Sie fügen diesem Ziel die folgenden Ressourcentypen hinzu: • `AWS::S3::Bucket` • `AWS::DynamoDB::Table` • `AWS::EC2::Instance`

Option	Description
	Wenn Sie dem Wartungsfenster später Aufgaben hinzufügen, nehmen Sie nur Aufgaben auf, die Aktionen für Knoten durchführen, wie z. B. das Anwenden einer Patch-Baseline oder das Neustarte n eines Knotens. Möglicherweise wird im Protokoll Wartungsfensterprotokoll ein Fehler gemeldet, dass keine Amazon Simple Storage Service (Amazon S3)-Buckets oder Amazon DynamoDB-Tabellen gefunden wurden. Das Wartungsfenster führt jedoch weiterhin Aufgaben auf den Knoten in Ihrer Ressourcengruppe aus.

9. Wählen Sie Register target.

Wenn Sie diesem Wartungsfenster mehrere Ziele zuweisen möchten, wählen Sie die Registerkarte Targets (Ziele) und anschließend Register target (Ziel registrieren) aus. Mit dieser Option können Sie eine andere Auswahlmethode festlegen. Wenn Sie beispielsweise zuvor Ziel-Knoten nach Knoten-ID ausgewählt haben, können Sie neue Ziele und Ziel-Knoten registrieren, indem Sie für verwaltete Knoten Tags angeben oder Ressourcentypen aus einer Ressourcengruppe auswählen.

Aufgaben zu einem Wartungsfenster mit der Konsole zuweisen

In diesem Verfahren fügen Sie eine Aufgabe zu einem Wartungsfenster hinzu. Aufgaben sind die Aktionen, die während der Ausführung eines Wartungsfensters durchgeführt werden.

Die folgenden vier Aufgabentypen können zu einem Wartungsfenster hinzugefügt werden:

- AWS Systems Manager Run Command-Befehle
- Systems Manager Automation-Workflows
- AWS Step Functions Aufgaben
- AWS Lambda Funktionen

> **Important**
>
> Die IAM-Richtlinie für Maintenance Windows erfordert, dass Sie den Namen von Lambda-Funktionen (oder Aliasen) das Präfix `SSM` hinzufügen. Bevor Sie mit der Registrierung dieser Art von Aufgabe fortfahren, aktualisieren Sie ihren Namen so, dass er AWS Lambda einschließt`SSM`. Beispiel: Wenn Ihr Lambda-Funktionsname `MyLambdaFunction` lautet, ändern Sie ihn in `SSMMyLambdaFunction`.

So weisen Sie einem Wartungsfenster Aufgaben zu

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie ein Wartungsfenster aus der Wartungsfensterliste aus.
4. Wählen Sie Actions (Aktionen) und anschließend die Option für den Aufgabentyp aus, den Sie für das Wartungsfenster registrieren möchten.

   - Register Run command task (Run Command-Aufgabe registrieren)
   - Register Automation task (Automatisierungsaufgabe registrieren)
   - Register Lambda task (Lambda-Aufgabe registrieren)
   - Register Step Functions task (Step Functions-Aufgabe registrieren)

   > **Note**
   >
   > Aufgaben im Wartungsfenster unterstützen nur Zustandsmaschinen-Workflows von Step Functions Standard. Sie unterstützen keine Express-Zustandsmaschinen-Workflows. Informationen zu Workflowtypen für Zustandsmaschinen finden Sie unter Standard- gegenüber Express-Workflows im AWS Step Functions - Entwicklerhandbuch.

5. (Optional) Geben Sie unter Name einen Namen für die Aufgabe ein.
6. (Optional) Geben Sie unter Description (Beschreibung) eine Beschreibung ein.

7. Wählen Sie für New task invocation cutoff (Cutoff für den Aufruf neuer Aufgaben) Enabled (Aktiviert), wenn Sie nicht möchten, dass neue Aufgabenaufrufe nach Erreichen der Grenzzeit des Wartungsfensters gestartet werden.

   Wenn diese Option nicht aktiviert ist, wird die Aufgabe weiter ausgeführt, wenn die Grenzzeit erreicht ist, und startet neue Aufgabenaufrufe bis zum Abschluss.

   > **Note**
   >
   > Der Status für Aufgaben, die beim Aktivieren dieser Option nicht abgeschlossen sind, lautet `TIMED_OUT`.

8. Folgen Sie für diesen Schritten den Unterschritten für den ausgewählten Aufgabentyp.

   Run Command

   1. Wählen Sie in der Liste Command document Befehlsdokument das Systems-Manager-Befehlsdokument (SSM-Dokument) aus, das die auszuführenden Aufgaben definiert.
   2. Wählen Sie für Document version (Dokumentversion) die zu verwendende Dokumentversion aus.
   3. Geben Sie für Task priority (Aufgabenpriorität) eine Priorität für diese Aufgabe an. Null (`0`) ist die höchste Priorität. Aufgaben in einem Wartungsfenster werden in Reihenfolge der Priorität geplant. Dabei werden Aufgaben mit derselben Priorität parallel ausgeführt.

   Automation

   1. Wählen Sie in der Liste Automation-Dokument das Automation-Runbook aus, das die auszuführende Aufgabe definiert.
   2. Wählen Sie für Document version (Dokumentversion) die zu verwendende Runbook-Version aus.
   3. Geben Sie für Task priority (Aufgabenpriorität) eine Priorität für diese Aufgabe an. Null (`0`) ist die höchste Priorität. Aufgaben in einem Wartungsfenster werden in Reihenfolge der Priorität geplant. Dabei werden Aufgaben mit derselben Priorität parallel ausgeführt.

   Lambda

   1. Wählen Sie im Bereich Lambda-Parameter eine Lambda-Funktion aus der Liste aus.

2. (Optional) Geben Sie einzubeziehende Inhalte für Payload (Nutzlast), Client Context (Client-Kontext) oder Qualifier (Qualifizierer) an.

> **Note**
>
> In einigen Fällen können Sie einen Pseudo-Parameter als Teil Ihres `Payload`-Werts verwenden. Während der Ausführung übergibt die Wartungsfenster-Aufgabe dann anstelle der Pseudoparameter-Platzhalter richtige Werte. Weitere Informationen finden Sie unter Verwendung von Pseudo-Parametern bei der Registrierung von Aufgaben im Wartungsfenster.

3. Geben Sie für Task priority (Aufgabenpriorität) eine Priorität für diese Aufgabe an. Null (`0`) ist die höchste Priorität. Aufgaben in einem Wartungsfenster werden in Reihenfolge der Priorität geplant. Dabei werden Aufgaben mit derselben Priorität parallel ausgeführt.

Step Functions

1. Wählen Sie im Bereich Step-Functions-Parameter eine Zustandsmaschine aus der Liste aus.
2. (Optional) Geben Sie einen Namen für die Ausführung des Zustandsautomaten und einzubeziehende Inhalte für Input (Eingabe) an.

> **Note**
>
> In einigen Fällen können Sie einen Pseudo-Parameter als Teil Ihres `Input`-Werts verwenden. Während der Ausführung übergibt die Wartungsfenster-Aufgabe dann anstelle der Pseudoparameter-Platzhalter richtige Werte. Weitere Informationen finden Sie unter Verwendung von Pseudo-Parametern bei der Registrierung von Aufgaben im Wartungsfenster.

3. Geben Sie für Task priority (Aufgabenpriorität) eine Priorität für diese Aufgabe an. Null (`0`) ist die höchste Priorität. Aufgaben in einem Wartungsfenster werden in Reihenfolge der Priorität geplant. Dabei werden Aufgaben mit derselben Priorität parallel ausgeführt.

9. Wählen Sie im Bereich Targets (Ziele) eine der folgenden Optionen aus:

- Auswahl registrierter Zielgruppen: Wählen Sie ein oder mehrere Wartungsfensterziele aus, die Sie im aktuellen Wartungsfenster registriert haben.

- Auswählen von nicht registrierten Zielen: Wählen Sie nacheinander verfügbare Ressourcen als Ziele für den Vorgang aus.

  Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

- Aufgabenziel nicht erforderlich: Ziele für die Aufgabe werden möglicherweise bereits in anderen Funktionen für alle Run Command-Typ-Aufgaben angegeben.

  Geben Sie ein oder mehrere Ziele für Wartungsfenster Run Command-Typ-Aufgaben an. Je nach Aufgabe sind Ziele für andere Aufgabentypen im Wartungsfenster (Automatisierung AWS Lambda, und AWS Step Functions) optional. Weitere Informationen zur Ausführung von Aufgaben, die keine Ziele angeben, finden Sie unter Wartungsfenster-Tasks ohne Ziele registrieren.

  > **Note**
  >
  > In vielen Fällen müssen Sie ein Ziel für eine Automatisierungsaufgabe nicht explizit angeben. Angenommen, Sie erstellen beispielsweise eine Automation-Aufgabe, um eine Amazon Machine Image (AMI) für Linux mit dem `AWS-UpdateLinuxAmi`-Runbook zu aktualisieren. Wenn die Aufgabe ausgeführt wird, wird AMI mit den neuesten verfügbaren Linux-Verteilungspaketen und Amazon-Software aktualisiert. Neue Instances, die aus der AMI erstellt wurden, haben diese Updates bereits installiert. Da die ID des AMI in den Eingabeparametern für das Runbook angegeben ist, muss in der Wartungsfenster-Aufgabe kein Ziel erneut angegeben werden.

10. Nur für Automation-Aufgaben:

    Geben Sie im Bereich Input parameters (Eingabeparameter) Werte für erforderliche oder optionale Parameter an, die zum Ausführen der Aufgabe notwendig sind.

    > **Note**
    >
    > In einigen Fällen können Sie einen Pseudoparameter für bestimmte Eingabeparameterwerte verwenden. Während der Ausführung übergibt die Wartungsfenster-Aufgabe dann anstelle der Pseudoparameter-Platzhalter richtige Werte.

Weitere Informationen finden Sie unter Verwendung von Pseudo-Parametern bei der Registrierung von Aufgaben im Wartungsfenster.

11. Für Ratenregelung:

    - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

      > **Note**
      >
      > Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, auf wie viele verwaltete Knoten es sich handelt, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

    - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

12. (Optional) Wählen Sie für IAM-Servicerolle eine Rolle aus, um Systems Manager Berechtigungen zur Übernahme zum Ausführen von Wartungsfenster-Aufgaben zu erteilen.

    Wenn Sie keinen ARN für die Servicerolle angeben, verwendet Systems Manager eine serviceverknüpfte Rolle in Ihrem Konto. Diese Rolle ist nicht im Dropdown-Menü aufgeführt. Wenn in Ihrem Konto keine geeignete serviceverknüpfte Rolle für Systems Manager vorhanden ist, wird sie erstellt, wenn die Aufgabe erfolgreich registriert wurde.

    > **Note**
    >
    > Um die Sicherheitslage zu verbessern, empfehlen wir dringend, eine benutzerdefinierte Richtlinie und eine benutzerdefinierte Servicerolle für die Ausführung Ihrer Aufgaben im Wartungsfenster zu erstellen. Die Richtlinie kann so gestaltet werden, dass sie nur die Berechtigungen gewährt, die für Ihre speziellen Wartungsfensteraufgaben erforderlich sind. Weitere Informationen finden Sie unter Einrichten von Maintenance Windows.

13. Nur für Run Command-Aufgaben:

(Optional) Gehen Sie für Output options (Ausgabeoptionen) wie folgt vor:

- Aktivieren Sie das Kontrollkästchen Enable writing to S3 (Schreiben in S3 aktivieren), um die Befehlsausgabe in einer Datei zu speichern. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.
- Aktivieren Sie das Kontrollkästchen CloudWatch Ausgabe, um die vollständige Ausgabe in Amazon CloudWatch Logs zu schreiben. Geben Sie den Namen einer CloudWatch Logs-Protokollgruppe ein.

> **Note**
>
> Die Berechtigungen, die das Schreiben von Daten in einen S3-Bucket oder in CloudWatch Logs ermöglichen, entsprechen denen des Instanzprofils, das dem Knoten zugewiesen ist, und nicht denen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Erforderliche Instance-Berechtigungen für Systems Manager konfigurieren. Wenn sich der angegebene S3-Bucket oder die angegebene Protokollgruppe in einem anderen befindet, stellen Sie außerdem sicher AWS-Konto, dass das dem Knoten zugeordnete Instanzprofil über die erforderlichen Berechtigungen verfügt, um in diesen Bucket zu schreiben.

14. Nur für Run Command-Aufgaben:

    Aktivieren Sie das Kontrollkästchen SNS-Benachrichtigungen aktivieren im Abschnitt SNS-Benachrichtigungen, wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon-SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

15. Nur für Run Command-Aufgaben:

    Geben Sie im Abschnitt Parameters (Parameter) die Parameter für das Dokument an.

> **Note**
>
> In einigen Fällen können Sie einen Pseudoparameter für bestimmte Eingabeparameterwerte verwenden. Während der Ausführung übergibt die Wartungsfenster-Aufgabe dann anstelle der Pseudoparameter-Platzhalter richtige Werte.

Weitere Informationen finden Sie unter Verwendung von Pseudo-Parametern bei der Registrierung von Aufgaben im Wartungsfenster.

16. Nur für Run Command und Automation-Aufgaben:

    (Optional) Wählen Sie im CloudWatch Alarmbereich unter Alarmname einen vorhandenen CloudWatch Alarm aus, der auf Ihre zu überwachende Aufgabe angewendet werden soll.

    Die Aufgabe wird gestoppt, wenn Ihr Alarm aktiviert wird.

    > **Note**
    >
    > Um Ihrer Aufgabe einen CloudWatch Alarm zuzuweisen, muss der IAM-Prinzipal, der die Aufgabe ausführt, über die entsprechenden Berechtigungen verfügen`iam:createServiceLinkedRole`. Weitere Informationen zu CloudWatch Alarmen finden Sie unter CloudWatch Amazon-Alarme verwenden.

17. Wählen Sie je nach Aufgabentyp eine der folgenden Optionen:

    - Register Run command task (Run Command-Aufgabe registrieren)
    - Register Automation task (Automatisierungsaufgabe registrieren)
    - Register Lambda task (Lambda-Aufgabe registrieren)
    - Register Step Functions task (Step Functions-Aufgabe registrieren)

Deaktivieren oder Aktivieren eines Wartungsfensters mithilfe der Konsole

Sie können ein Wartungsfenster in Maintenance Windows deaktivieren oder aktivieren, einem Tool in AWS Systems Manager. Sie können jeweils ein Wartungsfenster auswählen, um die Ausführung des Wartungsfensters entweder zu deaktivieren oder zu aktivieren. Sie können auch mehrere oder alle Wartungsfenster zum Aktivieren und Deaktivieren auswählen.

In diesem Abschnitt wird beschrieben, wie Sie ein Wartungsfenster mit Hilfe der Systems-Manager-Konsole deaktivieren oder aktivieren können. Beispiele dafür, wie dies mit der AWS Command Line Interface (AWS CLI) möglich ist, finden Sie unter Tutorial: Aktualisieren Sie ein Wartungsfenster mit dem AWS CLI.

Themen

- Deaktivieren eines Wartungsfensters mithilfe der Konsole

- Aktivieren eines Wartungsfensters mit der Konsole

Deaktivieren eines Wartungsfensters mithilfe der Konsole

Sie können ein Wartungsfenster deaktivieren, um eine Aufgabe für einen bestimmten Zeitraum anzuhalten, so dass sie später wieder aktiviert werden kann.

Um ein Wartungsfenster zu deaktivieren

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie mit Hilfe des Kontrollkästchens neben dem Wartungsfenster, das Sie deaktivieren möchten, ein oder mehrere Wartungsfenster aus.
4. Wählen Sie Wartungsfenster deaktivieren im Menü Aktionen. Sie werden aufgefordert, Ihre Aktionen zu bestätigen.

Aktivieren eines Wartungsfensters mit der Konsole

Sie können ein Wartungsfenster aktivieren, um eine Aufgabe wieder aufzunehmen.

> **Note**
>
> Wenn das Wartungsfenster eine Tariftabelle verwendet und das Startdatum derzeit auf ein Datum und eine vergangene Uhrzeit festgelegt ist, werden das aktuelle Datum und die aktuelle Uhrzeit als Startdatum für das Wartungsfenster verwendet. Sie können das Startdatum des Wartungsfensters vor oder nach der Aktivierung ändern. Weitere Informationen finden Sie unter Ressourcen für das Wartungsfenster mithilfe der Konsole aktualisieren oder löschen.

Ein Wartungsfenster aktivieren

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie das Kontrollkästchen neben dem Wartungsfenster, um es zu aktivieren.

4. Wählen Sie Aktionen, Wartungsfenster aktivieren. Sie werden aufgefordert, Ihre Aktionen zu bestätigen.

Ressourcen für das Wartungsfenster mithilfe der Konsole aktualisieren oder löschen

Sie können in Maintenance Windows, ein Tool in AWS Systems Manager, ein Wartungsfenster aktualisieren oder löschen. Sie können auch die Ziele oder Aufgaben eines Wartungsfensters aktualisieren oder löschen. Wenn Sie die Details eines Wartungsfensters bearbeiten, können Sie den Zeitplan, die Ziele und die Aufgaben ändern. Sie können auch Namen und Beschreibungen für Fenster, Ziele und Aufgaben angeben. Auf diese Weise erhalten Sie einen besseren Eindruck des Zwecks und vereinfachen die Verwaltung Ihrer Fensterwarteschlange.

In diesem Abschnitt wird beschrieben, wie Sie ein Wartungsfenster, Ziele und Aufgaben über die Systems Manager-Konsole aktualisieren oder löschen. Beispiele dafür, wie dies mit der AWS Command Line Interface (AWS CLI) möglich ist, finden Sie unter Tutorial: Aktualisieren Sie ein Wartungsfenster mit dem AWS CLI.

Themen

- Aktualisieren oder Löschen eines Wartungsfensters mithilfe der Konsole
- Aktualisieren oder Abmelden von Wartungsfenster-Zielen mithilfe der Konsole
- Aktualisieren oder Abmelden von Wartungsfenster-Aufgaben mithilfe der Konsole

Aktualisieren oder Löschen eines Wartungsfensters mithilfe der Konsole

Sie können ein Wartungsfenster aktualisieren, um den Namen, die Beschreibung und den Zeitplan des Wartungsfensters zu ändern und festzulegen, ob das Wartungsfenster nicht registrierte Ziele zulassen soll.

So aktualisieren oder löschen Sie ein Wartungsfenster

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie die Schaltfläche neben dem Wartungsfenster aus, das Sie aktualisieren oder löschen möchten, und führen Sie dann einen der folgenden Schritte aus:

   - Wählen Sie Löschen aus. Sie werden aufgefordert, Ihre Aktionen zu bestätigen.

- Wählen Sie Bearbeiten aus. Ändern Sie auf der Seite Edit maintenance window (Wartungsfenster bearbeiten) die Werte und Optionen nach Bedarf und wählen Sie dann Save changes (Änderungen speichern) aus.

  Weitere Informationen zu den Konfigurationsoptionen, die Sie ausführen können, finden Sie unter Erstellen eines Wartungsfensters mit der Konsole.

Aktualisieren oder Abmelden von Wartungsfenster-Zielen mithilfe der Konsole

Sie können die Ziele eines Wartungsfensters aktualisieren oder abmelden. Wenn Sie das Ziel eines Wartungsfensters aktualisieren möchten, können Sie einen neuen Zielnamen, eine Beschreibung und einen Eigentümer angeben. Sie können auch verschiedene Ziele auswählen.

So aktualisieren oder löschen Sie die Ziele eines Wartungsfensters

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie den Namen des zu aktualisierenden Wartungsfensters und danach die Registerkarte Targets (Ziele) aus und führen Sie dann einen der folgenden Schritte aus:

   - Um Ziele zu aktualisieren, klicken Sie auf die Schaltfläche neben dem zu aktualisierenden Ziel und wählen Sie dann Edit (Bearbeiten) aus.
   - Um Ziele abzumelden, klicken Sie auf die Schaltfläche neben dem abzumeldenden Ziel und wählen Sie dann Deregister target (Ziel abmelden) aus. Wählen Sie im Dialogfenster Deregister maintenance windows target (Deregistrierung des Wartungsfensterziels) die Option Deregistrierung.

Aktualisieren oder Abmelden von Wartungsfenster-Aufgaben mithilfe der Konsole

Sie können die Aufgaben eines Wartungsfensters aktualisieren oder abmelden. Wenn Sie eine Aktualisierung durchführen möchten, können Sie einen neuen Aufgabennamen, eine Beschreibung und einen Eigentümer angeben. Bei Run Command und Automation-Aufgaben können Sie ein anderes SSM-Dokument für die Aufgaben wählen. Sie können allerdings den Typ einer Aufgabe nicht durch Bearbeitung ändern. Beispiel: Wenn Sie eine Automatisierungsaufgabe erstellt haben, können Sie die Aufgabe nicht bearbeiten und in eine Run Command-Aufgabe ändern.

So aktualisieren oder löschen Sie die Aufgaben eines Wartungsfensters mithilfe der Konsole

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie den Namen des zu aktualisierenden Wartungsfensters aus.
4. Wählen Sie die Registerkarte Tasks (Aufgaben) aus und klicken Sie dann auf die Schaltfläche neben der zu aktualisierenden Aufgabe.
5. Führen Sie eine der folgenden Aktionen aus:
   - Um eine Aufgabe abzumelden, wählen Sie Deregister task (Aufgabe abmelden) aus.
   - Um die Aufgabe zu bearbeiten, wählen Sie Edit (Bearbeiten) aus. Ändern Sie die Werte und Optionen wie gewünscht und wählen Sie dann Edit Task (Aufgabe bearbeiten) aus.

Tutorials

Die Tutorials in diesem Abschnitt veranschaulichen, wie Sie mit Wartungsfenstern gängige Aufgaben durchführen.

Erfüllen der Voraussetzungen

Erfüllen Sie die folgenden Voraussetzungen, bevor Sie diese Tutorials ausführen.

- Konfigurieren der AWS CLI auf Ihrem lokalen Computer: Bevor Sie AWS CLI-Befehle ausführen können, müssen Sie die CLI auf Ihrem lokalen Computer installieren und konfigurieren. Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.
- Überprüfen der Wartungsfenster-Rollen und -Berechtigungen: Ein AWS-Administrator in Ihrem Konto muss Ihnen die AWS Identity and Access Management (IAM)-Berechtigungen gewähren, die Sie zur Verwaltung von Wartungsfenstern mithilfe der CLI benötigen. Weitere Informationen finden Sie unter Einrichten von Maintenance Windows.
- Erstellen oder Konfigurieren einer mit Systems Manager kompatiblen Instance Sie benötigen mindestens eine Amazon Elastic Compute Cloud (Amazon EC2)-Instance, die für die Verwendung mit Systems Manager konfiguriert ist, um die Tutorials abzuschließen. Dies bedeutet, dass SSM Agent auf der Instance installiert ist und der Instance ein IAM-Instance-Profil für Systems Manager zugeordnet ist.

Wir empfehlen, eine Instance von einem AWS verwalteten Amazon Machine Image (AMI) mit vorinstalliertem Agent zu starten. Weitere Informationen finden Sie unter Finden Sie AMIs mit dem vorinstallierten SSM Agent.

Weitere Informationen zum Installieren von SSM Agent auf einer Instance finden Sie in den folgenden Themen:

- Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Windows Server
- Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux

Informationen zum Konfigurieren von IAM-Berechtigungen für Systems Manager für Ihre Instance finden Sie unter Konfigurieren von erforderlichen Instance-Berechtigungen für Systems Manager.

- Erstellen zusätzlicher Ressourcen nach Bedarf: Run Command, ein Tool in Systems Manager, enthält viele Aufgaben, für die Sie keine anderen Ressourcen erstellen müssen als die, die in diesem Thema über die Voraussetzungen aufgeführt sind. Aus diesem Grund stellen wir Ihnen eine einfache Run Command-Aufgabe bereit, die Sie verwenden können, wenn Sie sich das erste Mal durch die Tutorials arbeiten. Sie benötigen außerdem eine EC2-Instance, die für die Verwendung mit Systems Manager konfiguriert ist, wie weiter oben in diesem Thema beschrieben. Nachdem Sie diese Instance konfiguriert haben, können Sie eine einfache Run Command-Aufgabe registrieren.

  Das Systems-Manager-Tool Maintenance Windows unterstützt vier Arten von Aufgaben:

  - Run Command-Befehle
  - Systems Manager Automation-Workflows
  - AWS Lambda-Funktionen
  - AWS Step Functions-Aufgaben

  Wenn eine Wartungsfensteraufgabe, die Sie ausführen möchten, zusätzliche Ressourcen erfordert, gilt im Allgemeinen, dass Sie diese zuerst erstellen sollten. Beispiel: Wenn Sie in einem Wartungsfenster eine AWS Lambda-Funktion ausführen möchten, erstellen Sie zuerst die Lambda-Funktion, bevor Sie beginnen. Erstellen Sie für eine Run Command-Aufgabe zuerst den S3-Bucket zum Speichern der Befehlsausgabe (sofern Sie dies vorhaben) usw.

Tutorials

- Tutorials: Wartungsfenster mit der AWS CLI erstellen und verwalten
- Tutorial: Erstellen Sie ein Wartungsfenster zum Patchen über die Konsole

Tutorials: Wartungsfenster mit der AWS CLI erstellen und verwalten

Dieser Abschnitt enthält Tutorials, die Sie mit der Verwendung der AWS Command Line Interface (AWS CLI) zur Ausführung der folgenden Schritte vertraut machen:

- Erstellen und Konfigurieren eines Wartungsfensters
- Anzeigen von Informationen zu Wartungsfenstern
- Anzeigen von Informationen über Wartungsfenster-Aufgaben und Aufgabenausführungen
- Aktualisieren eines Wartungsfensters
- Löschen eines Wartungsfensters

Verfolgen der Ressourcen-IDs

Wenn Sie die Aufgaben in diesen AWS CLI-Tutorials ausführen, verfolgen Sie die Ressource-IDs, die von den ausgeführten Befehlen generiert werden. Sie können viele davon als Eingabe für nachfolgende Befehle verwenden. Wenn Sie beispielsweise das Wartungsfenster erstellen, stellt das System eine Wartungsfenster-ID im folgenden Format für Sie bereit.

```
{
   "WindowId":"mw-0c50858d01EXAMPLE"
}
```

Notieren Sie sich die folgenden systemgenerierten IDs, da sie in den Tutorials in diesem Abschnitt verwendet werden:

- `WindowId`
- `WindowTargetId`
- `WindowTaskId`
- `WindowExecutionId`
- `TaskExecutionId`
- `InvocationId`
- `ExecutionId`

Sie benötigen außerdem die ID der EC2-Instance, die Sie in den Tutorials verwenden möchten. Beispiel: `i-02573cafcfEXAMPLE`

Tutorials

- Tutorial: Erstellen und konfigurieren Sie ein Wartungsfenster mit dem AWS CLI
- Tutorial: Informationen zu Wartungsfenstern anzeigen mit dem AWS CLI
- Tutorial: Informationen zu Aufgaben und Aufgabenausführungen anzeigen mit dem AWS CLI
- Tutorial: Aktualisieren Sie ein Wartungsfenster mit dem AWS CLI
- Tutorial: Löschen eines Wartungsfensters mithilfe von AWS CLI

Tutorial: Erstellen und konfigurieren Sie ein Wartungsfenster mit dem AWS CLI

In diesem Tutorial wird gezeigt, wie Sie mit AWS Command Line Interface (AWS CLI) ein Wartungsfenster sowie dessen Ziele und Aufgaben erstellen und konfigurieren. Der Hauptpfad durch das Tutorial besteht aus einfachen Schritten. Sie erstellen ein einziges Wartungsfenster, identifizieren ein einziges Ziel und richten eine einfache Aufgabe ein, die im Wartungsfenster ausgeführt werden soll. Entlang des Pfades stellen wir Informationen bereit, die Sie beim Ausprobieren komplexerer Szenarien unterstützen.

Wenn Sie die Schritte in diesem Tutorial befolgen, ersetzen Sie die Werte in kursivem *`red`* Text durch Ihre eigenen Optionen und. IDs Ersetzen Sie beispielsweise die ID des Wartungsfensters *`mw-0c50858d01EXAMPLE`* und die Instanz-ID durch die IDs von *`i-02573cafcfEXAMPLE`* Ihnen erstellten Ressourcen.

Inhalt

- Schritt 1: Erstellen Sie das Wartungsfenster mit dem AWS CLI
- Schritt 2: Ein Zielknoten mit dem Wartungsfenster mithilfe von AWS CLI registrieren
- Schritt 3: Registrieren einer Aufgaben für das Wartungsfenster mithilfe der AWS CLI

Schritt 1: Erstellen Sie das Wartungsfenster mit dem AWS CLI

In diesem Schritt erstellen Sie ein Wartungsfenster und geben die grundlegenden Optionen, wie z. B. Name, Zeitplan und Dauer, an. In späteren Schritten, wählen Sie die Instance aus, die aktualisiert werden soll, und legen die Aufgabe fest, die ausgeführt wird.

In unserem Beispiel erstellen Sie ein Wartungsfenster, das alle fünf Minuten ausgeführt wird. Normalerweise würden Sie ein Wartungsfenster nicht so häufig ausführen. Mit dieser Rate werden Ihre Ergebnisse des Tutorials schneller ersichtlich. Wir zeigen Ihnen, wie Sie zu einer weniger häufigen Rate wechseln, nachdem die Aufgabe erfolgreich ausgeführt wurde.

> **Note**
>
> Eine Beschreibung des Verhältnisses zwischen den verschiedenen zeitplanbezogenen Optionen für Wartungsfenster finden Sie unter Wartungsfenster-Optionen für Planung und aktive Zeiträume.
> Weitere Informationen zum Arbeiten mit der `--schedule`-Option finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

Um ein Wartungsfenster zu erstellen, verwenden Sie AWS CLI

1. Öffnen Sie AWS Command Line Interface (AWS CLI) und führen Sie den folgenden Befehl auf Ihrem lokalen Computer aus, um ein Wartungsfenster zu erstellen, das Folgendes bewirkt:

   - Es wird (je nach Bedarf) über bis zu zwei Stunden hinweg alle fünf Minuten ausgeführt.
   - Verhindert, dass bis zu einer Stunde nach der Ausführung des Wartungsfensters neue Aufgaben gestartet werden.
   - Es ermöglicht nicht zugeordnete Ziele (Instances, die Sie nicht beim Wartungsfenster registriert haben).
   - Es gibt durch die Verwendung von benutzerdefinierten Tags an, dass sein Ersteller beabsichtigt, es in einem Tutorial zu verwenden.

   Linux & macOS

   ```
   aws ssm create-maintenance-window \
       --name "My-First-Maintenance-Window" \
       --schedule "rate(5 minutes)" \
       --duration 2 \
       --cutoff 1 \
       --allow-unassociated-targets \
       --tags "Key=Purpose,Value=Tutorial"
   ```

   Windows

   ```
   aws ssm create-maintenance-window ^
       --name "My-First-Maintenance-Window" ^
       --schedule "rate(5 minutes)" ^
       --duration 2 ^
   ```

```
    --cutoff 1 ^
    --allow-unassociated-targets ^
    --tags "Key"="Purpose","Value"="Tutorial"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
   "WindowId":"mw-0c50858d01EXAMPLE"
}
```

2. Führen Sie jetzt den folgenden Befehl aus, um Details zu diesem und allen anderen Wartungsfenstern anzuzeigen, die Ihrem Konto bereits zugeordnet sind.

```
aws ssm describe-maintenance-windows
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
   "WindowIdentities":[
      {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "Enabled": true,
            "Duration": 2,
            "Cutoff": 1,
            "NextExecutionTime": "2019-05-11T16:46:16.991Z"
      }
   ]
}
```

Fahren Sie fort mit Schritt 2: Ein Zielknoten mit dem Wartungsfenster mithilfe von AWS CLI registrieren.

Schritt 2: Ein Zielknoten mit dem Wartungsfenster mithilfe von AWS CLI registrieren

In diesem Schritt registrieren Sie ein Ziel für Ihr neues Wartungsfenster. In diesem Fall geben Sie an, welcher Knoten aktualisiert werden soll, wenn das Wartungsfenster ausgeführt wird.

Ein Beispiel für die gleichzeitige Registrierung von mehr als einem Knoten mit Knoten-IDs, Beispiele zur Verwendung von Tags zur Identifizierung mehrerer Knoten und Beispiele für die Angabe von Ressourcengruppen als Ziele finden Sie unter Beispiele: Ziele für ein Wartungsfenster registrieren.

> **Note**
>
> Sie sollten bereits eine in diesem Schritt zu verwendende Amazon Elastic Compute Cloud (Amazon EC2)-Instance nach der Beschreibung in den Voraussetzungen für das Maintenance Windows-Tutorial erstellt haben.

So melden Sie einen Ziel-Knoten mit einem Wartungsfenster mithilfe von AWS CLI an

1. Führen Sie den folgenden Befehl auf Ihrem lokalen Computer aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

   Linux & macOS

   ```
   aws ssm register-target-with-maintenance-window \
       --window-id "mw-0c50858d01EXAMPLE" \
       --resource-type "INSTANCE" \
       --target "Key=InstanceIds,Values=i-02573cafcfEXAMPLE"
   ```

   Windows

   ```
   aws ssm register-target-with-maintenance-window ^
       --window-id "mw-0c50858d01EXAMPLE" ^
       --resource-type "INSTANCE" ^
       --target "Key=InstanceIds,Values=i-02573cafcfEXAMPLE"
   ```

   Die vom System zurückgegebenen Informationen ähneln den Folgenden.

   ```
   {
      "WindowTargetId":"e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
   }
   ```

2. Jetzt führen Sie den folgenden Befehl auf Ihrem lokalen Computer aus, um Details zu Ihrem Wartungsfensterziel anzuzeigen.

Linux & macOS

```
aws ssm describe-maintenance-window-targets \
    --window-id "mw-0c50858d01EXAMPLE"
```

Windows

```
aws ssm describe-maintenance-window-targets ^
    --window-id "mw-0c50858d01EXAMPLE"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "Targets": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowTargetId": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE",
            "ResourceType": "INSTANCE",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-02573cafcfEXAMPLE"
                    ]
                }
            ]
        }
    ]
}
```

Fahren Sie fort mit Schritt 3: Registrieren einer Aufgaben für das Wartungsfenster mithilfe der AWS CLI.

Beispiele: Ziele für ein Wartungsfenster registrieren

Sie können einen einzelnen Knoten mithilfe der Knoten-ID als Ziel registrieren. Die Anleitung dazu finden Sie unter Schritt 2: Ein Zielknoten mit dem Wartungsfenster mithilfe von AWS CLI registrieren. Darüber hinaus haben Sie die Möglichkeit, einen oder mehrere Knoten mithilfe der Befehlsformate auf dieser Seite als Ziele zu registrieren.

Im Allgemeinen gibt es zwei Methoden, um Knoten als Ziele für das Wartungsfenster zu identifizieren: durch das Festlegen einzelner Knoten und mithilfe von Ressourcen-Tags. Die Ressourcen-Tags-Methode bietet weitere Optionen, wie in den Beispielen 2 bis 3 gezeigt.

Sie können auch eine oder mehrere Ressourcengruppen als Ziel eines Wartungsfensters angeben. Eine Ressourcengruppe kann Knoten einschließen und viele andere Arten von unterstützten AWS-Ressourcen. Die Beispiele 4 und 5 demonstrieren nun, wie Sie Ihren Zielen für das Wartungsfenster Ressourcengruppen hinzufügen.

> **Note**
>
> Wenn eine einzelne Wartungsfenster-Aufgabe mit mehreren Zielen registriert ist, werden ihre Aufrufe sequenziell und nicht parallel ausgeführt. Wenn Ihre Aufgabe gleichzeitig auf mehreren Zielen ausgeführt werden muss, registrieren Sie eine Aufgabe für jedes Ziel einzeln, und weisen Sie jeder Aufgabe dieselbe Prioritätsstufe zu.

Weitere Informationen zum Erstellen und Verwalten von Ressourcengruppen finden Sie unter Was sind Ressourcengruppen? im AWS -Ressourcengruppen-Benutzerhandbuch und unter Resource Groups and Tagging for AWS im AWS News Blog.

Informationen zu Kontingenten für Maintenance Windows, ein Tool in AWS Systems Manager, zusätzlich zu den in den folgenden Beispielen angegebenen, finden Sie unter Service Quotas für Systems Manager im Allgemeine Amazon Web Services-Referenz.

Beispiel 1: Registrieren mehrerer Ziele mithilfe von Knoten-IDs

Führen Sie den folgenden Befehl auf Ihrer lokalen Maschine aus, um mehrere Knoten anhand ihrer Knoten-IDs als Ziele anzumelden. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

Linux & macOS

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "INSTANCE" \
    --target
 "Key=InstanceIds,Values=i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE,i-07782c72faEXAMPLE"
```

Windows

```
aws ssm register-target-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE ^
    --resource-type "INSTANCE" ^
    --target
 "Key=InstanceIds,Values=i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE,i-07782c72faEXAMPLE
```

Empfohlene Verwendung: Besonders nützlich bei der erstmaligen Registrierung einer eindeutigen Gruppe von Knoten bei einem beliebigen Wartungsfenster, wenn die Knoten nicht über ein gemeinsames Knoten-Tag verfügen.

Kontingente: Sie können insgesamt bis zu 50 Knoten für jedes Wartungsfenster-Ziel angeben.

Beispiel 2: Registrieren von Zielen mithilfe von Ressourcen-Tags, die auf Knoten angewendet werden

Führen Sie den folgenden Befehl auf Ihrer lokalen Maschine aus, um alle Knoten anzumelden, die bereits mit einem von Ihnen zugewiesenen Schlüssel-Wert-Paar markiert wurden. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen.

Linux & macOS

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "INSTANCE" \
    --target "Key=tag:Region,Values=East"
```

Windows

```
aws ssm register-target-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --resource-type "INSTANCE" ^
    --target "Key=tag:Region,Values=East"
```

Empfohlene Verwendung: Besonders nützlich bei der erstmaligen Registrierung einer eindeutigen Gruppe von Knoten bei einem beliebigen Wartungsfenster, wenn die Knoten schon über ein gemeinsames Knoten-Tag verfügen.

Kontingente: Sie können insgesamt bis zu fünf Schlüssel-Wert-Paare für jedes Ziel festlegen. Wenn Sie mehr als ein Schlüssel-Wert-Paar angeben, muss ein Knoten mit allen Tag-Schlüsseln und Werten gekennzeichnet werden, die Sie für die Aufnahme in die Zielgruppe angeben.

> **Note**
>
> Sie können eine Gruppe von Knoten mit dem Tag-Schlüssel `Patch Group` oder `PatchGroup` markieren und die Knoten einem gemeinsamen Schlüsselwert zuweisen, z. B. `my-patch-group`. (Sie müssen `PatchGroup` ohne Leerzeichen verwenden, wenn Sie Tags in EC2-Instance-Metadaten zugelassen haben.) Patch Manager, ein Tool in Systems Manager, wertet den Schlüssel `Patch Group` oder `PatchGroup` auf Knoten aus, um zu bestimmen, welche Patch-Baseline für sie gilt. Wenn Ihre Aufgabe das `AWS-RunPatchBaseline`-SSM-Dokument (oder das Legacy-`AWS-ApplyPatchBaseline`-SSM-Dokument) ausführt, können Sie das gleiche `Patch Group`- oder `PatchGroup`-Schlüssel-Wert-Paar angeben, wenn Sie Ziele für ein Wartungsfenster registrieren. Beispiel: `--target "Key=tag:PatchGroup,Values=`*`my-patch-group`*. Auf diese Weise können Sie Patches für eine Gruppe von Knoten, die schon der gleichen Patch-Baseline zugeordnet sind, mithilfe eines Wartungsfensters aktualisieren. Weitere Informationen finden Sie unter Patch-Gruppen.

Beispiel 3: Registrieren von Zielen mithilfe einer Gruppe von Tag-Schlüsseln (ohne Tag-Werte)

Führen Sie den folgenden Befehl auf Ihrer lokalen Maschine aus, um Knoten anzumelden, denen alle ein oder mehrere Tag-Schlüssel zugeordnet wurden, unabhängig von ihren Schlüsselwerten. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

Linux & macOS

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "INSTANCE" \
    --target "Key=tag-key,Values=Name,Instance-Type,CostCenter"
```

Windows

```
aws ssm register-target-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --resource-type "INSTANCE" ^
```

```
    --target "Key=tag-key,Values=Name,Instance-Type,CostCenter"
```

Empfohlene Verwendung: Diese Option ist hilfreich, wenn Sie Knoten als Ziele verwenden möchten, indem Sie mehrere Tag-Schlüssel (ohne Werte) angeben und nicht nur einen Tag-Schlüssel oder ein Tag-Schlüssel-Wert-Paar.

Kontingente: Sie können insgesamt bis zu fünf Tag-Schlüssel für jedes Ziel festlegen. Wenn Sie mehr als einen Tag-Schlüssel angeben, muss ein Knoten mit allen Tag-Schlüsseln und Werten markiert werden, die Sie für die Aufnahme in die Zielgruppe angeben.

Beispiel 4: Registrieren von Zielen unter Verwendung eines Ressourcengruppennamens

Führen Sie den folgenden Befehl zum Registrieren einer bestimmten Ressourcengruppe auf Ihrem lokalen Computer aus, unabhängig von der Art der Ressourcen, die sie enthält. Ersetzen Sie *mw-0c50858d01EXAMPLE* durch Ihre eigenen Informationen. Wenn die Aufgaben, die Sie dem Wartungsfenster zuweisen, auf eine Art von Ressource in dieser Ressourcengruppe nicht angewendet werden kann, meldet das System möglicherweise einen Fehler. Auch wenn ein solcher Fehler gemeldet wird, werden Aufgaben, für die ein unterstützter Ressourcentyp gefunden wurde, dennoch ausgeführt.

Linux & macOS

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "RESOURCE_GROUP" \
    --target "Key=resource-groups:Name,Values=MyResourceGroup"
```

Windows

```
aws ssm register-target-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --resource-type "RESOURCE_GROUP" ^
    --target "Key=resource-groups:Name,Values=MyResourceGroup"
```

Empfohlene Verwendung: Diese Vorgehensweise ist hilfreich, wenn Sie schnell eine Ressourcengruppe als Ziel angeben möchten, ohne auszuwerten, ob alle Ressourcentypen Ziel des Wartungsfensters sind, oder wenn Sie wissen, dass die Ressourcengruppe nur solche Ressourcentypen enthält, über denen Ihre Aufgaben Aktionen durchführen können.

Kontingente: Sie können nur eine Ressourcengruppe als Ziel angeben.

Beispiel 5: Registrieren von Zielen durch Filtern von Ressourcentypen in einer Ressourcengruppe

Führen Sie den folgenden Befehl auf Ihrem lokalen Computer aus, um nur bestimmte Ressourcentypen zu registrieren, die einer Ressourcengruppe angehören, die Sie angeben. Ersetzen Sie *mw-0c50858d01EXAMPLE* durch Ihre eigenen Informationen. Bei dieser Vorgehensweise wird eine Aufgabe, selbst wenn Sie sie für einen Ressourcentyp hinzufügen, der der Ressourcengruppe angehört, nicht ausgeführt, wenn Sie den Filter nicht explizit auf den Ressourcentyp setzen

Linux & macOS

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "RESOURCE_GROUP" \
    --target "Key=resource-groups:Name,Values=MyResourceGroup" \
    "Key=resource-
groups:ResourceTypeFilters,Values=AWS::EC2::Instance,AWS::ECS::Cluster"
```

Windows

```
aws ssm register-target-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --resource-type "RESOURCE_GROUP" ^
    --target "Key=resource-groups:Name,Values=MyResourceGroup" ^
    "Key=resource-
groups:ResourceTypeFilters,Values=AWS::EC2::Instance,AWS::ECS::Cluster"
```

Empfohlene Verwendung: Verwenden Sie diese Vorgehensweise, wenn Sie genau kontrollieren möchten, über welchen Typen von AWS-Ressourcen Ihr Wartungsfensters Aktionen ausführen kann, oder wenn Ihre Ressourcengruppe eine große Anzahl von Ressourcentypen enthält und Sie vermeiden möchten, dass unnötig viele Fehler im Wartungsfensterprotokoll aufgenommen werden.

Kontingente: Sie können nur eine Ressourcengruppe als Ziel angeben.

Schritt 3: Registrieren einer Aufgaben für das Wartungsfenster mithilfe der AWS CLI

In diesem Schritt des Tutorials registrieren Sie eine AWS Systems Manager Run Command-Aufgabe, die den `df`-Befehl auf Ihrer Amazon Elastic Compute Cloud (Amazon EC2)-Instance für Linux ausführt. Die Ergebnisse dieses Standard-Linux-Befehls zeigen, wie viel Speicherplatz frei ist und wie viel Speicherplatz auf dem Festplatten-Dateisystem Ihrer Instance belegt wird.

–oder–

Wenn Sie die Anweisungen an eine Amazon-EC2-Instance für Windows Server anstatt für Linux richten, ersetzen Sie df im folgenden Befehl durch ipconfig. Die Ausgabe dieses Befehls enthält Details über die IP-Adresse, die Subnetzmaske und das Standard-Gateway für Adapter auf der Ziel-Instance.

Wenn Sie zum Registrieren anderer Aufgabentypen oder zur Verwendung weiterer verfügbarer Run Command Systems Manager-Optionen bereit sind, finden Sie unter Beispiele: Registrieren von Aufgaben für ein Wartungsfenster weitere Informationen. Dort befinden sich weitere Informationen zu allen vier Aufgabentypen und einigen ihrer wichtigsten Optionen, die Sie beim Planen umfassenderer realer Szenarien unterstützen.

So registrieren Sie eine Aufgabe für ein Wartungsfenster

1. Führen Sie den folgenden Befehl auf Ihrem lokalen Computer aus. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen. Die Version, die von einem lokalen Windows-Computer aus ausgeführt werden soll, enthält die Escape-Zeichen („/“), die Sie zum Ausführen des Befehls über Ihr Befehlszeilen-Tool benötigen.

   Linux & macOS

   ```
   aws ssm register-task-with-maintenance-window \
       --window-id mw-0c50858d01EXAMPLE \
       --task-arn "AWS-RunShellScript" \
       --max-concurrency 1 --max-errors 1 \
       --priority 10 \
       --targets "Key=InstanceIds,Values=i-0471e04240EXAMPLE" \
       --task-type "RUN_COMMAND" \
       --task-invocation-parameters '{"RunCommand":{"Parameters":{"commands":
   ["df"]}}}'
   ```

   Windows

   ```
   aws ssm register-task-with-maintenance-window ^
       --window-id mw-0c50858d01EXAMPLE ^
       --task-arn "AWS-RunShellScript" ^
       --max-concurrency 1 --max-errors 1 ^
       --priority 10 ^
       --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE" ^
       --task-type "RUN_COMMAND" ^
   ```

```
    --task-invocation-parameters={\"RunCommand\":{\"Parameters\":{\"commands\":
[\"df\"]}}}
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden:

```
{
    "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE"
}
```

2. Führen Sie nun den folgenden Befehl aus, um Details zu der von Ihnen erstellten Wartungsfensteraufgabe anzuzeigen.

   Linux & macOS

   ```
   aws ssm describe-maintenance-window-tasks \
       --window-id mw-0c50858d01EXAMPLE
   ```

   Windows

   ```
   aws ssm describe-maintenance-window-tasks ^
       --window-id mw-0c50858d01EXAMPLE
   ```

3. Die vom System zurückgegebenen Informationen ähneln den Folgenden.

   ```
   {
       "Tasks": [
           {
               "WindowId": "mw-0c50858d01EXAMPLE",
               "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
               "TaskArn": "AWS-RunShellScript",
               "Type": "RUN_COMMAND",
               "Targets": [
                   {
                       "Key": "InstanceIds",
                       "Values": [
                           "i-02573cafcfEXAMPLE"
                       ]
                   }
               ],
               "TaskParameters": {},
               "Priority": 10,
   ```

```
            "ServiceRoleArn": "arn:aws:iam::123456789012:role/
MyMaintenanceWindowServiceRole",
            "MaxConcurrency": "1",
            "MaxErrors": "1"
        }
    ]
}
```

4. Räumen Sie basierend auf dem von Ihnen in Schritt 1: Erstellen Sie das Wartungsfenster mit dem AWS CLI angegebenen Zeitplan genügend Zeit für die Aufgabenausführung ein. Wenn Sie **`--schedule "rate(5 minutes)"`** angegeben haben, warten Sie beispielsweise fünf Minuten. Führen Sie dann den folgenden Befehl aus, um Informationen über alle Ausführungen anzuzeigen, die für diese Aufgabe aufgetreten sind.

   Linux & macOS

   ```
   aws ssm describe-maintenance-window-executions \
       --window-id mw-0c50858d01EXAMPLE
   ```

   Windows

   ```
   aws ssm describe-maintenance-window-executions ^
       --window-id mw-0c50858d01EXAMPLE
   ```

   Die vom System zurückgegebenen Informationen ähneln den Folgenden.

   ```
   {
       "WindowExecutions": [
           {
               "WindowId": "mw-0c50858d01EXAMPLE",
               "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
               "Status": "SUCCESS",
               "StartTime": 1557593493.096,
               "EndTime": 1557593498.611
           }
       ]
   }
   ```

Tip

Nachdem die Aufgabe erfolgreich durchgeführt wurde, können Sie die Rate verringern, mit der das Wartungsfenster ausgeführt wird. Führen Sie beispielsweise den folgenden Befehl aus, um die Häufigkeit auf einmal pro Woche zu verringern. Ersetzen Sie *mw-0c50858d01EXAMPLE* durch Ihre eigenen Informationen.

Linux & macOS

```
aws ssm update-maintenance-window \
    --window-id mw-0c50858d01EXAMPLE \
    --schedule "rate(7 days)"
```

Windows

```
aws ssm update-maintenance-window ^
    --window-id mw-0c50858d01EXAMPLE ^
    --schedule "rate(7 days)"
```

Weitere Informationen zum Verwalten von Wartungsfenster-Zeitplänen finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager und Wartungsfenster-Optionen für Planung und aktive Zeiträume.
Weitere Informationen zur Verwendung der AWS Command Line Interface (AWS CLI) zum Ändern eines Wartungsfensters finden Sie unter Tutorial: Aktualisieren Sie ein Wartungsfenster mit dem AWS CLI.

Um sich in der Ausführung von AWS CLI-Befehlen für die Ansicht weiterer Details zu Ihrer Wartungsfenster-Aufgabe und deren Ausführungen zu üben, fahren Sie mit Tutorial: Informationen zu Aufgaben und Aufgabenausführungen anzeigen mit dem AWS CLI fort.

Zugreifen auf die Tutorial-Befehlsausgabe

Es geht über den Rahmen dieses Tutorials hinaus, mithilfe der AWS CLI die Ausgabe des Run Command-Befehls für Ihre Wartungsfensteraufgabe-Ausführungen anzuzeigen.

Sie könnten diese Daten jedoch mithilfe der AWS CLI anzeigen. (Sie können die Ausgabe auch in der Systems Manager-Konsole oder in einer in einem Amazon Simple Storage Service (Amazon S3)-

Bucket gespeicherten Protokolldatei anzeigen, sofern Sie das Wartungsfenster zur Befehlsausgabe an dieser Stelle konfiguriert haben.) In diesem Fall würden Sie feststellen, dass die Ausgabe des Befehls df auf einer EC2-Instance für Linux ähnlich der folgenden ist.

```
Filesystem 1K-blocks Used Available Use% Mounted on

devtmpfs 485716 0 485716 0% /dev

tmpfs 503624 0 503624 0% /dev/shm

tmpfs 503624 328 503296 1% /run

tmpfs 503624 0 503624 0% /sys/fs/cgroup

/dev/xvda1 8376300 1464160 6912140 18% /
```

Die Ausgabe des Befehls ipconfig auf einer EC2-Instance für Windows Server sieht in etwa wie folgt aus:

```
Windows IP Configuration


Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . : example.com
   IPv4 Address. . . . . . . . . . . : 10.24.34.0/23
   Subnet Mask . . . . . . . . . . . : 255.255.255.255
   Default Gateway . . . . . . . . . : 0.0.0.0

Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : abc1.wa.example.net

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::100b:c234:66d6:d24f%4
```

```
   IPv4 Address. . . . . . . . . . . : 192.0.2.0
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.0.2.0

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

Beispiele: Registrieren von Aufgaben für ein Wartungsfenster

Sie können eine Aufgabe inRun Command, ein Tool in AWS Systems Manager, mit einem Wartungsfenster registrieren, indem Sie die AWS Command Line Interface (AWS CLI) verwenden, wie unter Aufgaben mit dem Wartungsfenster registrieren gezeigt. Sie können auch Aufgaben für Workflows, AWS Lambda Funktionen und AWS Step Functions Aufgaben von Systems Manager Automation registrieren, wie weiter unten in diesem Thema gezeigt wird.

> **Note**
>
> Geben Sie ein oder mehrere Ziele für Wartungsfenster Run Command-Typ-Aufgaben an. Je nach Aufgabe sind Ziele für andere Aufgabentypen im Wartungsfenster (Automatisierung AWS Lambda, und AWS Step Functions) optional. Weitere Informationen zur Ausführung von Aufgaben, die keine Ziele angeben, finden Sie unter Wartungsfenster-Tasks ohne Ziele registrieren.

In diesem Thema finden Sie Beispiele für die Verwendung des Befehls AWS Command Line Interface (AWS CLI)`register-task-with-maintenance-window`, um jeden der vier unterstützten Aufgabentypen in einem Wartungsfenster zu registrieren. Die Beispiele dienen nur zur Veranschaulichung. Sie können sie abwandeln, um funktionsfähige Befehle zur Aufgabenregistrierung zu erstellen.

Verwenden der cli-input-json Option --

Zur besseren Verwaltung Ihrer Aufgabenoptionen können Sie die Befehlsoption `--cli-input-json` mit in einer JSON-Datei referenzierten Optionswerten verwenden.

Um den Inhalt der JSON-Beispieldatei zu verwenden, den wir in den folgenden Beispielen bereitgestellt haben, führen Sie auf Ihrem lokalen Computer die die folgenden Schritte aus:

1. Erstellen Sie eine Datei mit einem Namen wie z. B. `MyRunCommandTask.json`, `MyAutomationTask.json` oder einem anderen von Ihnen bevorzugten Namen.
2. Kopieren Sie den Inhalt der JSON-Beispieldatei in die Datei.
3. Ändern Sie den Inhalt der Datei für Ihre Aufgabenregistrierung ab und speichern Sie dann die Datei.
4. Führen Sie in demselben Verzeichnis, in dem Sie die Datei gespeichert haben, den folgenden Befehl aus. Ersetzen Sie Ihren Dateinamen durch*`MyFile.json`*.

   Linux & macOS

   ```
   aws ssm register-task-with-maintenance-window \
       --cli-input-json file://MyFile.json
   ```

   Windows

   ```
   aws ssm register-task-with-maintenance-window ^
       --cli-input-json file://MyFile.json
   ```

Pseudo-Parameter in Wartungsfenster-Aufgaben

In einigen Beispielen verwenden wir Pseudoparameter als Methode zur Übergabe von ID-Informationen an Ihre Aufgaben. Zum Beispiel `{{TARGET_ID}}` und `{{RESOURCE_ID}}` kann verwendet werden, um AWS Ressourcen an Automation-, Lambda- und Step Functions Functions-Aufgaben weiterzugeben IDs . Weitere Informationen zu Pseudoparametern im `--task-invocation-parameters`-Inhalt finden Sie unter Verwendung von Pseudo-Parametern bei der Registrierung von Aufgaben im Wartungsfenster.

Weitere Informationen

- Parameteroptionen für den Befehl register-task-with-maintenance -windows.
- register-task-with-maintenance-window in der AWS CLI Befehlsreferenz
- RegisterTaskWithMaintenanceWindow in der AWS Systems Manager -API-Referenz

Beispiele der Aufgabenregistrierung

Die folgenden Abschnitte enthalten einen AWS CLI Beispielbefehl für die Registrierung eines unterstützten Aufgabentyps und ein JSON-Beispiel, das mit der `--cli-input-json` Option verwendet werden kann.

Registrieren einer Systems Manager Run Command-Aufgabe

Die folgenden Beispiele veranschaulichen, wie Sie Aufgaben von Systems Manager Run Command-Aufgaben mithilfe der AWS CLI bei einem Wartungsfenster registrieren.

Linux & macOS

```
aws ssm register-task-with-maintenance-window \
    --window-id mw-0c50858d01EXAMPLE \
    --task-arn "AWS-RunShellScript" \
    --max-concurrency 1 --max-errors 1 --priority 10 \
    --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE" \
    --task-type "RUN_COMMAND" \
    --task-invocation-parameters '{"RunCommand":{"Parameters":{"commands":["df"]}}}'
```

Windows

```
aws ssm register-task-with-maintenance-window ^
    --window-id mw-0c50858d01EXAMPLE ^
    --task-arn "AWS-RunShellScript" ^
    --max-concurrency 1 --max-errors 1 --priority 10 ^
    --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE" ^
    --task-type "RUN_COMMAND" ^
    --task-invocation-parameters "{\"RunCommand\":{\"Parameters\":{\"commands\":
[\"df\"]}}}"
```

JSON-Inhalt für die Verwendung mit der Dateioption **`--cli-input-json`**:

```
{
    "TaskType": "RUN_COMMAND",
    "WindowId": "mw-0c50858d01EXAMPLE",
    "Description": "My Run Command task to update SSM Agent on an instance",
    "MaxConcurrency": "1",
    "MaxErrors": "1",
    "Name": "My-Run-Command-Task",
    "Priority": 10,
```

```
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": [
                "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
            ]
        }
    ],
    "TaskArn": "AWS-UpdateSSMAgent",
    "TaskInvocationParameters": {
        "RunCommand": {
            "Comment": "A TaskInvocationParameters test comment",
            "NotificationConfig": {
                "NotificationArn": "arn:aws:sns:region:123456789012:my-sns-topic-name",
                "NotificationEvents": [
                    "All"
                ],
                "NotificationType": "Invocation"
            },
            "OutputS3BucketName": "amzn-s3-demo-bucket",
            "OutputS3KeyPrefix": "S3-PREFIX",
            "TimeoutSeconds": 3600
        }
    }
}
```

Registrieren einer Systems Manager Automation-Aufgabe

Die folgenden Beispiele veranschaulichen, wie Systems Manager Automation-Aufgaben mithilfe der bei einem Wartungsfenster registriert werden AWS CLI:

AWS CLI Befehl:

Linux & macOS

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --task-arn "AWS-RestartEC2Instance" \
    --service-role-arn arn:aws:iam::123456789012:role/MyMaintenanceWindowServiceRole
 \
    --task-type AUTOMATION \
    --task-invocation-parameters
 "Automation={DocumentVersion=5,Parameters={InstanceId='{{RESOURCE_ID}}'}}" \
```

```
    --priority 0 --name "My-Restart-EC2-Instances-Automation-Task" \
    --description "Automation task to restart EC2 instances"
```

Windows

```
aws ssm register-task-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --task-arn "AWS-RestartEC2Instance" ^
    --service-role-arn arn:aws:iam::123456789012:role/MyMaintenanceWindowServiceRole
 ^
    --task-type AUTOMATION ^
    --task-invocation-parameters
 "Automation={DocumentVersion=5,Parameters={InstanceId='{{TARGET_ID}}'}}" ^
    --priority 0 --name "My-Restart-EC2-Instances-Automation-Task" ^
    --description "Automation task to restart EC2 instances"
```

JSON-Inhalt für die Verwendung mit der Dateioption **`--cli-input-json`**:

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
        "TaskArn": "AWS-PatchInstanceWithRollback",
    "TaskType": "AUTOMATION","TaskInvocationParameters": {
        "Automation": {
            "DocumentVersion": "1",
            "Parameters": {
                "instanceId": [
                    "{{RESOURCE_ID}}"
                ]
            }
        }
    }
}
```

Registrieren einer AWS Lambda -Aufgabe

Die folgenden Beispiele veranschaulichen, wie Lambda-Funktionsaufgaben mithilfe der AWS CLI bei einem Wartungsfenster registriert werden.

Bei diesen Beispielen hat der Benutzer, der die Lambda-Funktion erstellt hat, ihr den Namen `SSMrestart-my-instances` gegeben und zwei Parameter mit dem Namen `instanceId` und `targetType` erstellt.

> **Important**
>
> Die IAM-Richtlinie für Maintenance Windows erfordert, dass Sie den Namen von Lambda-Funktionen (oder Aliasen) das Präfix `SSM` hinzufügen. Bevor Sie mit der Registrierung dieser Art von Aufgabe fortfahren, aktualisieren Sie ihren Namen so, dass er AWS Lambda einschließt`SSM`. Beispiel: Wenn Ihr Lambda-Funktionsname `MyLambdaFunction` lautet, ändern Sie ihn in `SSMMyLambdaFunction`.

AWS CLI Befehl:

Linux & macOS

> **Important**
>
> Wenn Sie Version 2 von verwenden AWS CLI, müssen Sie die Option `--cli-binary-format raw-in-base64-out` in den folgenden Befehl aufnehmen, wenn Ihre Lambda-Payload nicht base64-codiert ist. Die Option `cli_binary_format` ist nur in Version 2 verfügbar. Informationen zu diesen und anderen AWS CLI *`config`* Dateieinstellungen finden Sie im Benutzerhandbuch unter Unterstützte *`config`* Dateieinstellungen.AWS Command Line Interface

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" \
    --priority 2 --max-concurrency 10 --max-errors 5 --name "My-Lambda-Example" \
    --description "A description for my LAMBDA example task" --task-type "LAMBDA" \
    --task-arn "arn:aws:lambda:region:123456789012:function:serverlessrepo-
SSMrestart-my-instances-C4JF9EXAMPLE" \
    --task-invocation-parameters '{"Lambda":{"Payload":"{\"InstanceId\":
\"{{RESOURCE_ID}}\",\"targetType\":\"{{TARGET_TYPE}}\"}","Qualifier": "$LATEST"}}'
```

PowerShell

> **Important**
>
> Wenn Sie Version 2 von verwenden AWS CLI, müssen Sie die Option `--cli-binary-format raw-in-base64-out` in den folgenden Befehl aufnehmen, wenn Ihre Lambda-

Payload nicht base64-codiert ist. Die Option `cli_binary_format` ist nur in Version 2 verfügbar. Informationen zu diesen und anderen AWS CLI *`config`* Dateieinstellungen finden Sie im Benutzerhandbuch unter Unterstützte *`config`* Dateieinstellungen.AWS Command Line Interface

```
aws ssm register-task-with-maintenance-window `
    --window-id "mw-0c50858d01EXAMPLE" `
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" `
    --priority 2 --max-concurrency 10 --max-errors 5 --name "My-Lambda-Example" `
    --description "A description for my LAMBDA example task" --task-type "LAMBDA" `
    --task-arn "arn:aws:lambda:region:123456789012:function:serverlessrepo-
SSMrestart-my-instances-C4JF9EXAMPLE" `
    --task-invocation-parameters '{\"Lambda\":{\"Payload\":\"{\\\"InstanceId\\\":\\
\"{{RESOURCE_ID}}\\\",\\\"targetType\\\":\\\"{{TARGET_TYPE}}\\\"}\",\"Qualifier\":
 \"$LATEST\"}}'
```

JSON-Inhalt für die Verwendung mit der Dateioption **`--cli-input-json`**:

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": [
                "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
            ]
        }
    ],
    "TaskArn": "SSM_RestartMyInstances",
    "TaskType": "LAMBDA",
    "MaxConcurrency": "10",
    "MaxErrors": "10",
    "TaskInvocationParameters": {
        "Lambda": {
            "ClientContext": "ew0KICAi--truncated--0KIEXAMPLE",
            "Payload": "{ \"instanceId\": \"{{RESOURCE_ID}}\", \"targetType\":
 \"{{TARGET_TYPE}}\" }",
            "Qualifier": "$LATEST"
        }
    },
```

```
    "Name": "My-Lambda-Task",
    "Description": "A description for my LAMBDA task",
    "Priority": 5
}
```

Register a Step Functions task (Eine Step Functions-Aufgabe registrieren)

Die folgenden Beispiele veranschaulichen, wie Sie Aufgaben von Step Functions-Zustandsautomaten mithilfe der AWS CLI bei einem Wartungsfenster registrieren.

> **Note**
>
> Aufgaben im Wartungsfenster unterstützen nur Zustandsmaschinen-Workflows von Step Functions Standard. Sie unterstützen keine Express-Zustandsmaschinen-Workflows. Informationen zu Workflowtypen für Zustandsmaschinen finden Sie unter Standard- gegenüber Express-Workflows im AWS Step Functions -Entwicklerhandbuch.

In diesen Beispielen erstellte der Benutzer, der den Step Functions-Zustandsautomaten erstellt hatte, einen Zustandsautomaten mit dem Namen „`SSMMyStateMachine`“ und dem Parameter „`instanceId`“.

> **Important**
>
> Die AWS Identity and Access Management (IAM-) Richtlinie für Maintenance Windows erfordert, dass Sie Step Functions Functions-Zustandsmaschinen das Präfix voranstellen. `SSM` Bevor Sie mit dem Registrieren dieser Art von Aufgabe fortfahren, müssen Sie ihren Namen in AWS Step Functions so aktualisieren, dass in ihm `SSM` enthalten ist. Beispiel: Wenn der Name des Zustandsautomaten `MyStateMachine` lautet, ändern Sie ihn in `SSMMyStateMachine`.

AWS CLI Befehl:

Linux & macOS

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" \
```

```
    --task-arn arn:aws:states:region:123456789012:stateMachine:SSMMyStateMachine-
MggiqEXAMPLE \
    --task-type STEP_FUNCTIONS \
    --task-invocation-parameters '{"StepFunctions":{"Input":"{\"InstanceId\":
\"{{RESOURCE_ID}}\"}", "Name":"{{INVOCATION_ID}}"}}' \
    --priority 0 --max-concurrency 10 --max-errors 5 \
    --name "My-Step-Functions-Task" --description "A description for my Step
 Functions task"
```

PowerShell

```
aws ssm register-task-with-maintenance-window `
    --window-id "mw-0c50858d01EXAMPLE" `
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" `
    --task-arn arn:aws:states:region:123456789012:stateMachine:SSMMyStateMachine-
MggiqEXAMPLE `
    --task-type STEP_FUNCTIONS `
    --task-invocation-parameters '{\"StepFunctions\":{\"Input\":\"{\\\"InstanceId\\
\":\\\"{{RESOURCE_ID}}\\\"}\", \"Name\":\"{{INVOCATION_ID}}\"}}' `
    --priority 0 --max-concurrency 10 --max-errors 5 `
    --name "My-Step-Functions-Task" --description "A description for my Step
 Functions task"
```

JSON-Inhalt für die Verwendung mit der Dateioption **`--cli-input-json`**:

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": [
                "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
            ]
        }
    ],
    "TaskArn": "SSM_MyStateMachine",
    "TaskType": "STEP_FUNCTIONS",
    "MaxConcurrency": "10",
    "MaxErrors": "10",
    "TaskInvocationParameters": {
        "StepFunctions": {
            "Input": "{ \"instanceId\": \"{{TARGET_ID}}\" }",
```

```
            "Name": "{{INVOCATION_ID}}"
        }
    },
    "Name": "My-Step-Functions-Task",
    "Description": "A description for my Step Functions task",
    "Priority": 5
}
```

Parameteroptionen für den Befehl register-task-with-maintenance -windows

Der Befehl register-task-with-maintenance-window bietet mehrere Optionen für die Konfiguration einer Aufgabe entsprechend Ihren Anforderungen. Einige sind erforderlich, einige sind optional und einige gelten nur für einen einzigen Wartungsfenster-Aufgabentyp.

In diesem Thema erhalten Sie Informationen zu einigen dieser Optionen, um Sie bei der Arbeit mit Beispielen in diesem Abschnitt des Tutorials zu unterstützen. Informationen über alle Befehlsoptionen finden Sie unter register-task-with-maintenance-window in der AWS CLI Command Reference.

Befehlsoption: **`--task-arn`**

Die Option `--task-arn` wird verwendet, um die Ressource anzugeben, die von der Aufgabe ausgeführt wird. Der von Ihnen angegebene Wert hängt wie in der folgenden Tabelle beschrieben, davon ab, welche Art von Aufgabe Sie registrieren möchten.

TaskArn Formate für Aufgaben im Wartungsfenster

Wartungsfenster-Aufgabentyp	TaskArn Wert
`RUN_COMMAND` und **`AUTOMATION`**	`TaskArn` ist der SSM-Dokumentname oder der Amazon-Ressourcenname (ARN). Zum Beispiel: `AWS-RunBatchShellScript` –oder– `arn:aws:ssm:` *`region`*`:11112222 3333:document/My-Document` .
`LAMBDA`	`TaskArn` ist der Funktionsname oder -ARN. Zum Beispiel:

Wartungsfenster-Aufgabentyp	TaskArn Wert
	`SSMMy-Lambda-Function` –oder– `arn:aws:lambda: region:11112222 3333:function:SSMMyLambdaFu nction` . **Important** Die IAM-Richtlinie für Maintenance Windows erfordert, dass Sie den Namen von Lambda-Funktionen (oder Aliasen) das Präfix `SSM` hinzufügen. Bevor Sie mit der Registrierung dieser Art von Aufgabe fortfahren, aktualisieren Sie ihren Namen so, dass er AWS Lambda einschließt`SSM`. Beispiel: Wenn Ihr Lambda-Funktionsname `MyLambdaFunction` lautet, ändern Sie ihn in `SSMMyLambdaFunction` .

Wartungsfenster-Aufgabentyp	TaskArn Wert
STEP_FUNCTIONS	TaskArn ist der ARN des Zustandsautomaten. Zum Beispiel: arn:aws:states:us-east-2:111122223333:stateMachine:SSMMyStateMachine . **Important** Die IAM-Richtlinie für Wartungsfenster erfordert, dass Sie Step Functions -Zustandsautomaten-Namen das Präfix SSM geben. Bevor Sie diese Art der Aufgabe registrieren können, müssen Sie ihren Namen in AWS Step Functions so aktualisieren, dass in ihm SSM enthalten ist. Beispiel: Wenn der Name des Zustandsautomaten MyStateMachine lautet, ändern Sie ihn in SSMMyStateMachine .

Befehlsoption: **--service-role-arn**

Die Rolle AWS Systems Manager , die bei der Ausführung der Wartungsfensteraufgabe übernommen werden soll.

Weitere Informationen finden Sie unter Einrichten von Maintenance Windows.

Befehlsoption: **--task-invocation-parameters**

Die Option --task-invocation-parameters wird dazu verwendet, jene Parameter anzugeben, die nur für die vier Aufgabentypen gelten. Die unterstützten Parameter für jede der vier Arten von Aufgaben werden in der folgenden Tabelle beschrieben.

> **Note**
>
> Weitere Informationen über die Verwendung von Pseudoparametern in `--task-invocation-parameters`-Inhalten, z. B. {{TARGET_ID}}, finden Sie unter Verwendung von Pseudo-Parametern bei der Registrierung von Aufgaben im Wartungsfenster.

Aufgabenaufruf-Parameteroptionen für Wartungsfenster-Aufgaben

Wartungsfenster-Aufgabentyp	Verfügbare Parameter	Beispiel

RUN_COMMAND	Comment
	DocumentHash
	DocumentHashType
	NotificationConfig
	OutputS3BucketName
	OutPutS3KeyPrefix
	Parameters
	ServiceRoleArn
	TimeoutSeconds
		<pre>"TaskInvocationParameters": {
		"RunCommand": {
		"Comment": "My Run Command task comment",
		"DocumentHash": "6554ed3d--truncated--5EXAMPLE",
		"DocumentHashType": "Sha256",
		"NotificationConfig": {
		"NotificationArn": "arn:aws:sns: region:123456789012:my-sns-topic-name",
		"NotificationEvents": [
		"FAILURE"
		],
		"NotificationType": "Invocation"
		},</pre>

Wartungsfenster-Aufgabentyp	Verfügbare Parameter	Beispiel

```
            "OutputS3
BucketName": "amzn-s3-
demo-bucket",
            "OutputS3
KeyPrefix": "  S3-PREFIX
 ",
            "Paramete
rs": {

 "commands": [

 "Get-ChildItem$env:
 temp-Recurse|Remove-
Item-Recurse-force"
                ]
            },
            "ServiceR
oleArn": "arn:aws:
iam::123456789012:
role/MyMaintenance
WindowServiceRole",
            "TimeoutS
econds": 3600
        }
    }
```

Wartungsfenster-Aufgabentyp	Verfügbare Parameter	Beispiel
AUTOMATION	DocumentVersion Parameters	"TaskInvocationParameters": { "Automation": { "DocumentVersion": "3", "Parameters": { "instanceid": ["{{TARGET_ID}}"] } } }
LAMBDA	ClientContext Payload Qualifier	"TaskInvocationParameters": { "Lambda": { "ClientContext": "ew0KICAi--truncated--0KIEXAMPLE", "Payload": "{ \"targetId\": \"{{TARGET_ID}}\", \"targetType\": \"{{TARGET_TYPE}}\" }", "Qualifier": "$LATEST" } }

Wartungsfenster-Aufgabentyp	Verfügbare Parameter	Beispiel

STEP_FUNCTIONS	Input
	
	Name
		"TaskInvocationParameters": {
		"StepFunctions": {
		"Input": "{ \"targetId\": \"{{TARGET_ID}}\" }",
		"Name": "{{INVOCATION_ID}}"
		}
		}

Tutorial: Informationen zu Wartungsfenstern anzeigen mit dem AWS CLI

In diesem Tutorial sind Befehle enthalten, mit denen Sie Ihre Wartungsfenster, Aufgaben, Ausführungen und Aufrufe aktualisieren oder Informationen darüber abrufen können. Die Beispiele sind nach Befehl geordnet, um zu zeigen, wie Befehlsoptionen verwendet werden, um nach der Art von Details zu filtern, die Sie anzeigen möchten.

Wenn Sie die Schritte in diesem Tutorial befolgen, ersetzen Sie die Werte in kursivem *red* Text durch Ihre eigenen Optionen und. IDs Ersetzen Sie beispielsweise die ID des Wartungsfensters *mw-0c50858d01EXAMPLE* und die Instanz-ID durch die IDs von *i-02573cafcfEXAMPLE* Ihnen erstellten Ressourcen.

Informationen zur Einrichtung und Konfiguration von AWS Command Line Interface (AWS CLI) finden Sie unter Installation, Aktualisierung und Deinstallation von AWS CLI und Konfiguration von. AWS CLI

Befehlsbeispiele

- Beispiele für " describe-maintenance-windows
- Beispiele für 'describe-maintenance-window-targets'
- Beispiele für 'describe-maintenance-window-tasks'
- Beispiele für 'describe-maintenance-windows-for-target'
- Beispiele für " describe-maintenance-window-executions
- Beispiele für 'describe-maintenance-window-schedule'

Beispiele für " describe-maintenance-windows

Listet alle Wartungsfenster in Ihrem auf AWS-Konto

Führen Sie den folgenden Befehl aus.

```
aws ssm describe-maintenance-windows
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
   "WindowIdentities":[
      {
         "WindowId":"mw-0c50858d01EXAMPLE",
         "Name":"My-First-Maintenance-Window",
         "Enabled":true,
         "Duration":2,
         "Cutoff":0,
         "NextExecutionTime": "2019-05-18T17:01:01.137Z"
      },
      {
         "WindowId":"mw-9a8b7c6d5eEXAMPLE",
         "Name":"My-Second-Maintenance-Window",
         "Enabled":true,
         "Duration":4,
         "Cutoff":1,
         "NextExecutionTime": "2019-05-30T03:30:00.137Z"
      },
   ]
}
```

Alle aktivierten Wartungsfenster aufführen

Führen Sie den folgenden Befehl aus.

```
aws ssm describe-maintenance-windows --filters "Key=Enabled,Values=true"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
   "WindowIdentities":[
```

```
      {
         "WindowId":"mw-0c50858d01EXAMPLE",
         "Name":"My-First-Maintenance-Window",
         "Enabled":true,
         "Duration":2,
         "Cutoff":0,
         "NextExecutionTime": "2019-05-18T17:01:01.137Z"
      },
      {
         "WindowId":"mw-9a8b7c6d5eEXAMPLE",
         "Name":"My-Second-Maintenance-Window",
         "Enabled":true,
         "Duration":4,
         "Cutoff":1,
         "NextExecutionTime": "2019-05-30T03:30:00.137Z"
      },
   ]
}
```

Alle deaktivierten Wartungsfenster aufführen

Führen Sie den folgenden Befehl aus.

```
aws ssm describe-maintenance-windows --filters "Key=Enabled,Values=false"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "WindowIdentities": [
        {
            "WindowId": "mw-6e5c9d4b7cEXAMPLE",
            "Name": "My-Disabled-Maintenance-Window",
            "Enabled": false,
            "Duration": 2,
            "Cutoff": 1
        }
    ]
}
```

Alle Wartungsfenster aufführen, deren Name mit einem bestimmten Präfix beginnt

Führen Sie den folgenden Befehl aus.

```
aws ssm describe-maintenance-windows --filters "Key=Name,Values=My"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "WindowIdentities": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "Enabled": true,
            "Duration": 2,
            "Cutoff": 0,
            "NextExecutionTime": "2019-05-18T17:01:01.137Z"
        },
        {
            "WindowId": "mw-9a8b7c6d5eEXAMPLE",
            "Name": "My-Second-Maintenance-Window",
            "Enabled": true,
            "Duration": 4,
            "Cutoff": 1,
            "NextExecutionTime": "2019-05-30T03:30:00.137Z"
        },
        {
            "WindowId": "mw-6e5c9d4b7cEXAMPLE",
            "Name": "My-Disabled-Maintenance-Window",
            "Enabled": false,
            "Duration": 2,
            "Cutoff": 1
        }
    ]
}
```

Beispiele für 'describe-maintenance-window-targets'

Die Ziele für ein Wartungsfenster anzeigen, das einem bestimmten Eigentümer-Informationswert entspricht

Führen Sie den folgenden Befehl aus.

Linux & macOS

```
aws ssm describe-maintenance-window-targets \
```

```
    --window-id "mw-6e5c9d4b7cEXAMPLE" \
    --filters "Key=OwnerInformation,Values=CostCenter1"
```

Windows

```
aws ssm describe-maintenance-window-targets ^
    --window-id "mw-6e5c9d4b7cEXAMPLE" ^
    --filters "Key=OwnerInformation,Values=CostCenter1"
```

> **Note**
>
> Die unterstützten Filterschlüssel sind `Type`, `WindowTargetId` und `OwnerInformation`.

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "Targets": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowTargetId": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE",
            "ResourceType": "INSTANCE",
            "Targets": [
                {
                    "Key": "tag:Name",
                    "Values": [
                        "Production"
                    ]
                }
            ],
            "OwnerInformation": "CostCenter1",
            "Name": "Target1"
        }
    ]
}
```

Beispiele für 'describe-maintenance-window-tasks'

Alle registrierten Aufgaben anzeigen, die das SSM-Befehlsdokument **`AWS-RunPowerShellScript`** aufrufen

Führen Sie den folgenden Befehl aus.

Linux & macOS

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-0c50858d01EXAMPLE" \
    --filters "Key=TaskArn,Values=AWS-RunPowerShellScript"
```

Windows

```
aws ssm describe-maintenance-window-tasks ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --filters "Key=TaskArn,Values=AWS-RunPowerShellScript"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
   "Tasks":[
      {
         "ServiceRoleArn": "arn:aws:iam::111122223333:role/
MyMaintenanceWindowServiceRole",
         "MaxErrors":"1",
         "TaskArn":"AWS-RunPowerShellScript",
         "MaxConcurrency":"1",
         "WindowTaskId":"4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
         "TaskParameters":{
            "commands":{
               "Values":[
                  "driverquery.exe"
               ]
            }
         },
         "Priority":3,
         "Type":"RUN_COMMAND",
         "Targets":[
            {
               "TaskTargetId":"i-02573cafcfEXAMPLE",
               "TaskTargetType":"INSTANCE"
            }
         ]
      },
      {
```

```
         "ServiceRoleArn":"arn:aws:iam::111122223333:role/
MyMaintenanceWindowServiceRole",
         "MaxErrors":"1",
         "TaskArn":"AWS-RunPowerShellScript",
         "MaxConcurrency":"1",
         "WindowTaskId":"4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
         "TaskParameters":{
            "commands":{
               "Values":[
                  "ipconfig"
               ]
            }
         },
         "Priority":1,
         "Type":"RUN_COMMAND",
         "Targets":[
            {
               "TaskTargetId":"i-02573cafcfEXAMPLE",
               "TaskTargetType":"WINDOW_TARGET"
            }
         ]
      }
   ]
}
```

Alle registrierten Aufgaben mit Priorität 3 anzeigen

Führen Sie den folgenden Befehl aus.

Linux & macOS

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-9a8b7c6d5eEXAMPLE" \
    --filters "Key=Priority,Values=3"
```

Windows

```
aws ssm describe-maintenance-window-tasks ^
    --window-id "mw-9a8b7c6d5eEXAMPLE" ^
    --filters "Key=Priority,Values=3"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
   "Tasks":[
      {
         "ServiceRoleArn":"arn:aws:iam::111122223333:role/
MyMaintenanceWindowServiceRole",
         "MaxErrors":"1",
         "TaskArn":"AWS-RunPowerShellScript",
         "MaxConcurrency":"1",
         "WindowTaskId":"4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
         "TaskParameters":{
            "commands":{
               "Values":[
                  "driverquery.exe"
               ]
            }
         },
         "Priority":3,
         "Type":"RUN_COMMAND",
         "Targets":[
            {
               "TaskTargetId":"i-02573cafcfEXAMPLE",
               "TaskTargetType":"INSTANCE"
            }
         ]
      }
   ]
}
```

Alle registrierten Aufgaben anzeigen, die Priorität "1" haben und Run Command verwenden

Führen Sie den folgenden Befehl aus.

Linux & macOS

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-0c50858d01EXAMPLE" \
    --filters "Key=Priority,Values=1" "Key=TaskType,Values=RUN_COMMAND"
```

Windows

```
aws ssm describe-maintenance-window-tasks ^
    --window-id "mw-0c50858d01EXAMPLE" ^
```

```
    --filters "Key=Priority,Values=1" "Key=TaskType,Values=RUN_COMMAND"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "Tasks": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
            "TaskArn": "AWS-RunShellScript",
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-02573cafcfEXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 1,
            "ServiceRoleArn": "arn:aws:iam::111122223333:role/
MyMaintenanceWindowServiceRole",
            "MaxConcurrency": "1",
            "MaxErrors": "1"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowTaskId": "8a5c4629-31b0-4edd-8aea-33698EXAMPLE",
            "TaskArn": "AWS-UpdateSSMAgent",
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-0471e04240EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 1,
```

```
            "ServiceRoleArn": "arn:aws:iam::111122223333:role/
MyMaintenanceWindowServiceRole",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "My-Run-Command-Task",
            "Description": "My Run Command task to update SSM Agent on an instance"
        }
    ]
}
```

Beispiele für 'describe-maintenance-windows-for-target'

Informationen über die Wartungsfensterziele oder -Aufgaben im Zusammenhang mit einem bestimmten Knoten aufführen

Führen Sie den folgenden Befehl aus.

Linux & macOS

```
aws ssm describe-maintenance-windows-for-target \
    --resource-type INSTANCE \
    --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE" \
    --max-results 10
```

Windows

```
aws ssm describe-maintenance-windows-for-target ^
    --resource-type INSTANCE ^
    --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE" ^
    --max-results 10
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "WindowIdentities": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window"
        },
        {
            "WindowId": "mw-9a8b7c6d5eEXAMPLE",
            "Name": "My-Second-Maintenance-Window"
```

```
        }
    ]
}
```

Beispiele für " describe-maintenance-window-executions

Alle Aufgaben aufführen, die vor einem bestimmten Datum ausgeführt wurden

Führen Sie den folgenden Befehl aus.

Linux & macOS

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-9a8b7c6d5eEXAMPLE" \
    --filters "Key=ExecutedBefore,Values=2019-05-12T05:00:00Z"
```

Windows

```
aws ssm describe-maintenance-window-executions ^
    --window-id "mw-9a8b7c6d5eEXAMPLE" ^
    --filters "Key=ExecutedBefore,Values=2019-05-12T05:00:00Z"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "WindowExecutions": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
            "Status": "FAILED",
            "StatusDetails": "The following SSM parameters are invalid: LevelUp",
            "StartTime": 1557617747.993,
            "EndTime": 1557617748.101
        },
        {
            "WindowId": "mw-9a8b7c6d5eEXAMPLE",
            "WindowExecutionId": "791b72e0-f0da-4021-8b35-f95dfEXAMPLE",
            "Status": "SUCCESS",
            "StartTime": 1557594085.428,
            "EndTime": 1557594090.978
        },
        {
```

```
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowExecutionId": "ecec60fa-6bb0-4d26-98c7-140308EXAMPLE",
            "Status": "SUCCESS",
            "StartTime": 1557593793.483,
            "EndTime": 1557593798.978
        }
    ]
}
```

Alle Aufgaben aufführen, die nach einem bestimmten Datum ausgeführt wurden

Führen Sie den folgenden Befehl aus.

Linux & macOS

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-9a8b7c6d5eEXAMPLE" \
    --filters "Key=ExecutedAfter,Values=2018-12-31T17:00:00Z"
```

Windows

```
aws ssm describe-maintenance-window-executions ^
    --window-id "mw-9a8b7c6d5eEXAMPLE" ^
    --filters "Key=ExecutedAfter,Values=2018-12-31T17:00:00Z"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "WindowExecutions": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
            "Status": "FAILED",
            "StatusDetails": "The following SSM parameters are invalid: LevelUp",
            "StartTime": 1557617747.993,
            "EndTime": 1557617748.101
        },
        {
            "WindowId": "mw-9a8b7c6d5eEXAMPLE",
            "WindowExecutionId": "791b72e0-f0da-4021-8b35-f95dfEXAMPLE",
            "Status": "SUCCESS",
            "StartTime": 1557594085.428,
```

```
            "EndTime": 1557594090.978
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowExecutionId": "ecec60fa-6bb0-4d26-98c7-140308EXAMPLE",
            "Status": "SUCCESS",
            "StartTime": 1557593793.483,
            "EndTime": 1557593798.978
        }
    ]
}
```

Beispiele für 'describe-maintenance-window-schedule'

Die nächsten zehn Wartungsfenster-Ausführungen, die für einen bestimmten Knoten geplant sind, anzeigen

Führen Sie den folgenden Befehl aus.

Linux & macOS

```
aws ssm describe-maintenance-window-schedule \
    --resource-type INSTANCE \
    --targets "Key=InstanceIds,Values=i-07782c72faEXAMPLE" \
    --max-results 10
```

Windows

```
aws ssm describe-maintenance-window-schedule ^
    --resource-type INSTANCE ^
    --targets "Key=InstanceIds,Values=i-07782c72faEXAMPLE" ^
    --max-results 10
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "ScheduledWindowExecutions": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2019-05-18T23:35:24.902Z"
        },
```

```
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2019-05-25T23:35:24.902Z"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2019-06-01T23:35:24.902Z"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2019-06-08T23:35:24.902Z"
        },
        {
            "WindowId": "mw-9a8b7c6d5eEXAMPLE",
            "Name": "My-Second-Maintenance-Window",
            "ExecutionTime": "2019-06-15T23:35:24.902Z"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2019-06-22T23:35:24.902Z"
        },
        {
            "WindowId": "mw-9a8b7c6d5eEXAMPLE",
            "Name": "My-Second-Maintenance-Window",
            "ExecutionTime": "2019-06-29T23:35:24.902Z"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2019-07-06T23:35:24.902Z"
        },
        {
            "WindowId": "mw-9a8b7c6d5eEXAMPLE",
            "Name": "My-Second-Maintenance-Window",
            "ExecutionTime": "2019-07-13T23:35:24.902Z"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "My-First-Maintenance-Window",
            "ExecutionTime": "2019-07-20T23:35:24.902Z"
```

```
        }
    ],
    "NextToken": "AAEABUXdceT92FvtKld/dGHELj5Mi+GKW/EXAMPLE"
}
```

Den Wartungsfenster-Zeitplan für Knoten anzeigen, die mit einem bestimmten Schlüssel-Wert-Paar markiert sind

Führen Sie den folgenden Befehl aus.

Linux & macOS

```
aws ssm describe-maintenance-window-schedule \
    --resource-type INSTANCE \
    --targets "Key=tag:prod,Values=rhel7"
```

Windows

```
aws ssm describe-maintenance-window-schedule ^
    --resource-type INSTANCE ^
    --targets "Key=tag:prod,Values=rhel7"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "ScheduledWindowExecutions": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "DemoRateStartDate",
            "ExecutionTime": "2019-10-20T05:34:56-07:00"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "DemoRateStartDate",
            "ExecutionTime": "2019-10-21T05:34:56-07:00"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "DemoRateStartDate",
            "ExecutionTime": "2019-10-22T05:34:56-07:00"
        },
```

```
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "DemoRateStartDate",
            "ExecutionTime": "2019-10-23T05:34:56-07:00"
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "Name": "DemoRateStartDate",
            "ExecutionTime": "2019-10-24T05:34:56-07:00"
        }
    ],
    "NextToken": "AAEABccwSXqQRGKiTZ1yzGELR6cxW4W/EXAMPLE"
}
```

Startzeiten für die nächsten vier Ausführungen eines Wartungsfensters anzeigen

Führen Sie den folgenden Befehl aus.

Linux & macOS

```
aws ssm describe-maintenance-window-schedule \
    --window-id "mw-0c50858d01EXAMPLE" \
    --max-results "4"
```

Windows

```
aws ssm describe-maintenance-window-schedule ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --max-results "4"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "WindowSchedule": [
        {
            "ScheduledWindowExecutions": [
                {
                    "ExecutionTime": "2019-10-04T10:10:10Z",
                    "Name": "My-First-Maintenance-Window",
                    "WindowId": "mw-0c50858d01EXAMPLE"
                },
                {
```

```
                    "ExecutionTime": "2019-10-11T10:10:10Z",
                    "Name": "My-First-Maintenance-Window",
                    "WindowId": "mw-0c50858d01EXAMPLE"
                },
                {
                    "ExecutionTime": "2019-10-18T10:10:10Z",
                    "Name": "My-First-Maintenance-Window",
                    "WindowId": "mw-0c50858d01EXAMPLE"
                },
                {
                    "ExecutionTime": "2019-10-25T10:10:10Z",
                    "Name": "My-First-Maintenance-Window",
                    "WindowId": "mw-0c50858d01EXAMPLE"
                }
            ]
        }
    ]
}
```

Tutorial: Informationen zu Aufgaben und Aufgabenausführungen anzeigen mit dem AWS CLI

In diesem Tutorial wird gezeigt, wie Sie mit AWS Command Line Interface (AWS CLI) Details zu Ihren abgeschlossenen Aufgaben im Wartungsfenster anzeigen können.

Wenn Sie direkt von Tutorial: Erstellen und konfigurieren Sie ein Wartungsfenster mit dem AWS CLI fortfahren, überprüfen Sie, dass genügend Zeit verstrichen ist, damit das Wartungsfenster mindestens einmal ausgeführt werden konnte, um die Ausführungsergebnisse anzuzeigen.

Wenn Sie die Schritte in diesem Tutorial befolgen, ersetzen Sie die Werte in kursivem *red* Text durch Ihre eigenen Optionen und. IDs Ersetzen Sie beispielsweise die ID des Wartungsfensters *mw-0c50858d01EXAMPLE* und die Instanz-ID durch die IDs von *i-02573cafcfEXAMPLE* Ihnen erstellten Ressourcen.

So zeigen Sie Informationen über Aufgaben und Aufgabenausführungen mithilfe der AWS CLI an

1. Führen Sie den folgenden Befe.l aus, um eine Liste der Aufgabenausführungen für ein bestimmtes Wartungsfenster anzuzeigen:

   Linux & macOS

   ```
   aws ssm describe-maintenance-window-executions \
       --window-id "mw-0c50858d01EXAMPLE"
   ```

Windows

```
aws ssm describe-maintenance-window-executions ^
    --window-id "mw-0c50858d01EXAMPLE"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "WindowExecutions": [
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
            "Status": "SUCCESS",
            "StartTime": 1557593793.483,
            "EndTime": 1557593798.978
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowExecutionId": "791b72e0-f0da-4021-8b35-f95dfEXAMPLE",
            "Status": "SUCCESS",
            "StartTime": 1557593493.096,
            "EndTime": 1557593498.611
        },
        {
            "WindowId": "mw-0c50858d01EXAMPLE",
            "WindowExecutionId": "ecec60fa-6bb0-4d26-98c7-140308EXAMPLE",
            "Status": "SUCCESS",
            "StatusDetails": "No tasks to execute.",
            "StartTime": 1557593193.309,
            "EndTime": 1557593193.334
        }
    ]
}
```

2. Führen Sie den folgenden Befehl aus, um Informationen zu der Aufgabenausführung eines Wartungsfensters abzurufen.

   Linux & macOS

   ```
   aws ssm get-maintenance-window-execution \
   ```

```
    --window-execution-id "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE"
```

Windows

```
aws ssm get-maintenance-window-execution ^
    --window-execution-id "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
    "TaskIds": [
        "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE"
    ],
    "Status": "SUCCESS",
    "StartTime": 1557593493.096,
    "EndTime": 1557593498.611
}
```

3. Führen Sie den folgenden Befehl aus, um eine Liste der Aufgabenausführungen als Teil einer Wartungsfenster-Ausführung anzuzeigen.

   Linux & macOS

   ```
   aws ssm describe-maintenance-window-execution-tasks \
       --window-execution-id "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE"
   ```

   Windows

   ```
   aws ssm describe-maintenance-window-execution-tasks ^
       --window-execution-id "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE"
   ```

   Die vom System zurückgegebenen Informationen ähneln den Folgenden.

   ```
   {
       "WindowExecutionTaskIdentities": [
           {
               "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
               "TaskExecutionId": "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE",
   ```

```
            "Status": "SUCCESS",
            "StartTime": 1557593493.162,
            "EndTime": 1557593498.57,
            "TaskArn": "AWS-RunShellScript",
            "TaskType": "RUN_COMMAND"
        }
    ]
}
```

4. Führen Sie den folgenden Befehl aus, um Details zu einer Aufgabenausführung abzurufen.

   Linux & macOS

   ```
   aws ssm get-maintenance-window-execution-task \
       --window-execution-id "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE" \
       --task-id "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE"
   ```

   Windows

   ```
   aws ssm get-maintenance-window-execution-task ^
       --window-execution-id "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE" ^
       --task-id "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE"
   ```

   Die vom System zurückgegebenen Informationen ähneln den Folgenden.

   ```
   {
       "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
       "TaskExecutionId": "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE",
       "TaskArn": "AWS-RunShellScript",
       "ServiceRole": "arn:aws:iam::111122223333:role/MyMaintenanceWindowServiceRole",
       "Type": "RUN_COMMAND",
       "TaskParameters": [
           {
               "aws:InstanceId": {
                   "Values": [
                       "i-02573cafcfEXAMPLE"
                   ]
               },
               "commands": {
                   "Values": [
                       "df"
   ```

```
                ]
            }
        }
    ],
    "Priority": 10,
    "MaxConcurrency": "1",
    "MaxErrors": "1",
    "Status": "SUCCESS",
    "StartTime": 1557593493.162,
    "EndTime": 1557593498.57
}
```

5. Führen Sie den folgenden Befehl aus, um die spezifischen Aufgabenaufrufe abzurufen, die bei einer Aufgabenausführung durchgeführt werden.

   Linux & macOS

   ```
   aws ssm describe-maintenance-window-execution-task-invocations \
       --window-execution-id "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE" \
       --task-id "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE"
   ```

   Windows

   ```
   aws ssm describe-maintenance-window-execution-task-invocations ^
       --window-execution-id "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE" ^
       --task-id "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE"
   ```

   Die vom System zurückgegebenen Informationen ähneln den Folgenden.

   ```
   {
       "WindowExecutionTaskInvocationIdentities": [
           {
               "WindowExecutionId": "14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
               "TaskExecutionId": "c9b05aba-197f-4d8d-be34-e73fbEXAMPLE",
               "InvocationId": "c336d2ab-09de-44ba-8f6a-6136cEXAMPLE",
               "ExecutionId": "76a5a04f-caf6-490c-b448-92c02EXAMPLE",
               "TaskType": "RUN_COMMAND",
               "Parameters": "{\"documentName\":\"AWS-RunShellScript\",\"instanceIds
   \":[\"i-02573cafcfEXAMPLE\"],\"maxConcurrency\":\"1\",\"maxErrors\":\"1\",
   \"parameters\":{\"commands\":[\"df\"]}}",
               "Status": "SUCCESS",
   ```

```
            "StatusDetails": "Success",
            "StartTime": 1557593493.222,
            "EndTime": 1557593498.466
        }
    ]
}
```

Tutorial: Aktualisieren Sie ein Wartungsfenster mit dem AWS CLI

Dieses Tutorial zeigt, wie Sie das AWS Command Line Interface (AWS CLI) verwenden, um ein Wartungsfenster zu aktualisieren. Es zeigt Ihnen auch, wie Sie verschiedene Aufgabentypen aktualisieren, einschließlich der Aufgabentypen für AWS Systems Manager Run Command und Automatisierung AWS Lambda, und AWS Step Functions.

In den Beispielen dieses Abschnitts werden die folgenden Systems Manager-Aktionen zum Aktualisieren eines Wartungsfensters verwendet:

- UpdateMaintenanceWindow
- UpdateMaintenanceWindowTarget
- UpdateMaintenanceWindowTask
- DeregisterTargetFromMaintenanceWindow

Weitere Informationen zum Aktualisieren eines Wartungsfensters über die Systems Manager-Konsole finden Sie unter Ressourcen für das Wartungsfenster mithilfe der Konsole aktualisieren oder löschen.

Wenn Sie die Schritte in diesem Tutorial befolgen, ersetzen Sie die Werte in kursivem *red* Text durch Ihre eigenen Optionen und. IDs Ersetzen Sie beispielsweise die ID des Wartungsfensters *mw-0c50858d01EXAMPLE* und die Instanz-ID durch die IDs von *i-02573cafcfEXAMPLE* Ihnen erstellten Ressourcen.

Um ein Wartungsfenster mit dem zu aktualisieren AWS CLI

1. Öffnen Sie das AWS CLI und führen Sie den folgenden Befehl aus, um ein Ziel so zu aktualisieren, dass es einen Namen und eine Beschreibung enthält.

   Linux & macOS

   ```
   aws ssm update-maintenance-window-target \
       --window-id "mw-0c50858d01EXAMPLE" \
   ```

```
    --window-target-id "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" \
    --name "My-Maintenance-Window-Target" \
    --description "Description for my maintenance window target"
```

Windows

```
aws ssm update-maintenance-window-target ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --window-target-id "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" ^
    --name "My-Maintenance-Window-Target" ^
    --description "Description for my maintenance window target"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "WindowTargetId": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE",
    "Targets": [
        {
            "Key": "InstanceIds",
            "Values": [
                "i-02573cafcfEXAMPLE"
            ]
        }
    ],
    "Name": "My-Maintenance-Window-Target",
    "Description": "Description for my maintenance window target"
}
```

2. Führen Sie den folgenden Befehl aus, um mit der `replace`-Option das Beschreibungsfeld zu entfernen und ein zusätzliches Ziel hinzuzufügen. Das Beschreibungsfeld wird gelöscht, da die Aktualisierung das Feld nicht enthält (NULL-Wert). Stellen Sie sicher, dass Sie einen zusätzlichen Knoten angeben, der für die Verwendung mit Systems Manager konfiguriert wurde.

   Linux & macOS

```
aws ssm update-maintenance-window-target \
    --window-id "mw-0c50858d01EXAMPLE" \
    --window-target-id "d208dedf-3f6b-41ff-ace8-8e751EXAMPLE" \
    --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE" \
    --name "My-Maintenance-Window-Target" \
```

```
    --replace
```

Windows

```
aws ssm update-maintenance-window-target ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --window-target-id "d208dedf-3f6b-41ff-ace8-8e751EXAMPLE" ^
    --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE,i-0471e04240EXAMPLE" ^
    --name "My-Maintenance-Window-Target" ^
    --replace
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "WindowTargetId": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE",
    "Targets": [
        {
            "Key": "InstanceIds",
            "Values": [
                "i-02573cafcfEXAMPLE",
                "i-0471e04240EXAMPLE"
            ]
        }
    ],
    "Name": "My-Maintenance-Window-Target"
}
```

3. Die Option `start-date` erlaubt Ihnen, die Aktivierung eines Wartungsfensters bis zu einem angegebenen künftigen Zeitpunkt zu verzögern. Die Option `end-date` erlaubt Ihnen, ein in der Zukunft liegendes Datum sowie eine Uhrzeit festzulegen, nach dem das Wartungsfenster nicht mehr ausgeführt wird. Geben Sie die Optionen im erweiterten ISO-8601-Format an.

   Führen Sie den folgenden Befehl aus, um ein Datum oder eine Zeitspanne für die regelmäßig geplanten Wartungsfenster-Ausführungen anzugeben.

   Linux & macOS

```
aws ssm update-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --start-date "2020-10-01T10:10:10Z" \
```

```
    --end-date "2020-11-01T10:10:10Z"
```

Windows

```
aws ssm update-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --start-date "2020-10-01T10:10:10Z" ^
    --end-date "2020-11-01T10:10:10Z"
```

4. Führen Sie den folgenden Befehl aus, um eine Run Command-Aufgabe zu aktualisieren.

> **Tip**
>
> Wenn Ihr Ziel eine Amazon Elastic Compute Cloud (Amazon EC2) -Instance für ist Windows Serveripconfig, ändern Sie df `AWS-RunPowerShellScript` im folgenden Befehl `AWS-RunShellScript` zu und zu.

Linux & macOS

```
aws ssm update-maintenance-window-task \
    --window-id "mw-0c50858d01EXAMPLE" \
    --window-task-id "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE" \
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
 \
    --task-arn "AWS-RunShellScript" \
    --service-role-arn "arn:aws:iam::account-id:role/MaintenanceWindowsRole" \
    --task-invocation-parameters "RunCommand={Comment=Revising my Run Command
 task,Parameters={commands=df}}" \
    --priority 1 --max-concurrency 10 --max-errors 4 \
    --name "My-Task-Name" --description "A description for my Run Command task"
```

Windows

```
aws ssm update-maintenance-window-task ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --window-task-id "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE" ^
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
 ^
    --task-arn "AWS-RunShellScript" ^
    --service-role-arn "arn:aws:iam::account-id:role/MaintenanceWindowsRole" ^
```

```
    --task-invocation-parameters "RunCommand={Comment=Revising my Run Command
task,Parameters={commands=df}}" ^
    --priority 1 --max-concurrency 10 --max-errors 4 ^
    --name "My-Task-Name" --description "A description for my Run Command task"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": [
                "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
            ]
        }
    ],
    "TaskArn": "AWS-RunShellScript",
    "ServiceRoleArn": "arn:aws:iam::111122223333:role/MaintenanceWindowsRole",
    "TaskParameters": {},
    "TaskInvocationParameters": {
        "RunCommand": {
            "Comment": "Revising my Run Command task",
            "Parameters": {
                "commands": [
                    "df"
                ]
            }
        }
    },
    "Priority": 1,
    "MaxConcurrency": "10",
    "MaxErrors": "4",
    "Name": "My-Task-Name",
    "Description": "A description for my Run Command task"
}
```

5. Passen Sie den folgenden Befehl aus und führen Sie ihn aus, um eine Lambda-Aufgabe zu aktualisieren.

Linux & macOS

```
aws ssm update-maintenance-window-task \
    --window-id mw-0c50858d01EXAMPLE \
    --window-task-id 4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE \
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
 \
    --task-arn "arn:aws:lambda:region:111122223333:function:SSMTestLambda" \
    --service-role-arn "arn:aws:iam:account-id:role/MaintenanceWindowsRole" \
    --task-invocation-parameters '{"Lambda":{"Payload":"{\"InstanceId\":
\"{{RESOURCE_ID}}\",\"targetType\":\"{{TARGET_TYPE}}\"}"}}' \
    --priority 1 --max-concurrency 10 --max-errors 5 \
    --name "New-Lambda-Task-Name" \
    --description "A description for my Lambda task"
```

Windows

```
aws ssm update-maintenance-window-task ^
    --window-id mw-0c50858d01EXAMPLE ^
    --window-task-id 4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE ^
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
 ^
    --task-arn --task-arn
 "arn:aws:lambda:region:111122223333:function:SSMTestLambda" ^
    --service-role-arn "arn:aws:iam:account-id:role/MaintenanceWindowsRole" ^
    --task-invocation-parameters '{"Lambda":{"Payload":"{\"InstanceId\":
\"{{RESOURCE_ID}}\",\"targetType\":\"{{TARGET_TYPE}}\"}"}}' ^
    --priority 1 --max-concurrency 10 --max-errors 5 ^
    --name "New-Lambda-Task-Name" ^
    --description "A description for my Lambda task"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
```

```
        }
    ],
    "TaskArn": "arn:aws:lambda:us-east-2:111122223333:function:SSMTestLambda",
    "ServiceRoleArn": "arn:aws:iam::111122223333:role/MaintenanceWindowsRole",
    "TaskParameters": {},
    "TaskInvocationParameters": {
        "Lambda": {
            "Payload": "e30="
        }
    },
    "Priority": 1,
    "MaxConcurrency": "10",
    "MaxErrors": "5",
    "Name": "New-Lambda-Task-Name",
    "Description": "A description for my Lambda task"
}
```

6. Wenn Sie eine Step Functions Functions-Aufgabe aktualisieren, passen Sie sie an und führen Sie den folgenden Befehl aus, um sie zu aktualisieren task-invocation-parameters.

   Linux & macOS

   ```
   aws ssm update-maintenance-window-task \
       --window-id "mw-0c50858d01EXAMPLE" \
       --window-task-id "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE" \
       --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
    \
       --task-arn "arn:aws:states:region:execution:SSMStepFunctionTest" \
       --service-role-arn "arn:aws:iam:account-id:role/MaintenanceWindowsRole" \
       --task-invocation-parameters '{"StepFunctions":{"Input":"{\"InstanceId\":
   \"{{RESOURCE_ID}}\"}"}}' \
       --priority 0 --max-concurrency 10 --max-errors 5 \
       --name "My-Step-Functions-Task" \
       --description "A description for my Step Functions task"
   ```

   Windows

   ```
   aws ssm update-maintenance-window-task ^
       --window-id "mw-0c50858d01EXAMPLE" ^
       --window-task-id "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE" ^
       --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
    ^
       --task-arn "arn:aws:states:region:execution:SSMStepFunctionTest" ^
   ```

```
    --service-role-arn "arn:aws:iam:account-id:role/MaintenanceWindowsRole" ^
    --task-invocation-parameters '{"StepFunctions":{"Input":"{\"InstanceId\":
\"{{RESOURCE_ID}}\"}"}}' ^
    --priority 0 --max-concurrency 10 --max-errors 5 ^
    --name "My-Step-Functions-Task" ^
    --description "A description for my Step Functions task"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": [
                "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
            ]
        }
    ],
    "TaskArn": "arn:aws:states:us-
east-2:111122223333:execution:SSMStepFunctionTest",
    "ServiceRoleArn": "arn:aws:iam::111122223333:role/MaintenanceWindowsRole",
    "TaskParameters": {},
    "TaskInvocationParameters": {
        "StepFunctions": {
            "Input": "{\"instanceId\":\"{{RESOURCE_ID}}\"}"
        }
    },
    "Priority": 0,
    "MaxConcurrency": "10",
    "MaxErrors": "5",
    "Name": "My-Step-Functions-Task",
    "Description": "A description for my Step Functions task"
}
```

7. Führen Sie den folgenden Befehl aus, um ein Ziel von einem Wartungsfenster abzumelden. In diesem Beispiel wird der `safe`-Parameter verwendet, um zu bestimmen, ob beliebige Aufgaben auf das Ziel verweisen und es sicher abgemeldet werden kann.

Linux & macOS

```
aws ssm deregister-target-from-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --window-target-id "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" \
    --safe
```

Windows

```
aws ssm deregister-target-from-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --window-target-id "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" ^
    --safe
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
An error occurred (TargetInUseException) when calling the
 DeregisterTargetFromMaintenanceWindow operation:
This Target cannot be deregistered because it is still referenced in Task:
 4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE
```

8. Führen Sie den folgenden Befehl aus, um ein Ziel auch dann von einem Wartungsfenster abzumelden, wenn eine Aufgabe auf das Ziel verweist. Sie können den Abmeldevorgang mit dem `no-safe`-Parameter erzwingen.

   Linux & macOS

   ```
   aws ssm deregister-target-from-maintenance-window \
       --window-id "mw-0c50858d01EXAMPLE" \
       --window-target-id "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" \
       --no-safe
   ```

   Windows

   ```
   aws ssm deregister-target-from-maintenance-window ^
       --window-id "mw-0c50858d01EXAMPLE" ^
       --window-target-id "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE" ^
       --no-safe
   ```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "WindowTargetId": "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
}
```

9. Führen Sie den folgenden Befehl aus, um eine Run Command-Aufgabe zu aktualisieren. Dieses Beispiel verwendet einen Parameter Store Systems Manager-Parameter mit dem Namen `UpdateLevel` und der folgenden Formatierung: `{{ssm:UpdateLevel}}'`

   Linux & macOS

   ```
   aws ssm update-maintenance-window-task \
       --window-id "mw-0c50858d01EXAMPLE" \
       --window-task-id "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE" \
       --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE"  \
       --task-invocation-parameters "RunCommand={Comment=A comment for my task
    update,Parameters={UpdateLevel='{{ssm:UpdateLevel}}'}}"
   ```

   Windows

   ```
   aws ssm update-maintenance-window-task ^
       --window-id "mw-0c50858d01EXAMPLE" ^
       --window-task-id "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE" ^
       --targets "Key=InstanceIds,Values=i-02573cafcfEXAMPLE"  ^
       --task-invocation-parameters "RunCommand={Comment=A comment for my task
    update,Parameters={UpdateLevel='{{ssm:UpdateLevel}}'}}"
   ```

   Die vom System zurückgegebenen Informationen ähneln den Folgenden.

   ```
   {
       "WindowId": "mw-0c50858d01EXAMPLE",
       "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
       "Targets": [
           {
               "Key": "InstanceIds",
               "Values": [
                   "i-02573cafcfEXAMPLE"
   ```

```
            ]
        }
    ],
    "TaskArn": "AWS-RunShellScript",
    "ServiceRoleArn": "arn:aws:iam::111122223333:role/
MyMaintenanceWindowServiceRole",
    "TaskParameters": {},
    "TaskInvocationParameters": {
        "RunCommand": {
            "Comment": "A comment for my task update",
            "Parameters": {
                "UpdateLevel": [
                    "{{ssm:UpdateLevel}}"
                ]
            }
        }
    },
    "Priority": 10,
    "MaxConcurrency": "1",
    "MaxErrors": "1"
}
```

10. Führen Sie den folgenden Befehl aus, um eine Automatisierungsaufgabe so zu aktualisieren, dass `WINDOW_ID`-Parameter und `WINDOW_TASK_ID`-Parameter als `task-invocation-parameters`-Parameter angegeben werden:

    Linux & macOS

    ```
    aws ssm update-maintenance-window-task \
        --window-id "mw-0c50858d01EXAMPLE" \
        --window-task-id "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE" \
        --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE \
        --task-arn "AutoTestDoc" \
        --service-role-arn "arn:aws:iam:account-id:role/
    MyMaintenanceWindowServiceRole \
        --task-invocation-parameters
     "Automation={Parameters={InstanceId='{{RESOURCE_ID}}',initiator='{{WINDOW_ID}}.Task-
    {{WINDOW_TASK_ID}}'}}" \
        --priority 3 --max-concurrency 10 --max-errors 5
    ```

    Windows

    ```
    aws ssm update-maintenance-window-task ^
    ```

```
    --window-id "mw-0c50858d01EXAMPLE" ^
    --window-task-id "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE" ^
    --targets "Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE ^
    --task-arn "AutoTestDoc" ^
    --service-role-arn "arn:aws:iam:account-id:role/
MyMaintenanceWindowServiceRole ^
    --task-invocation-parameters
 "Automation={Parameters={InstanceId='{{RESOURCE_ID}}',initiator='{{WINDOW_ID}}.Task-
{{WINDOW_TASK_ID}}'}}" ^
    --priority 3 --max-concurrency 10 --max-errors 5
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
    "WindowId": "mw-0c50858d01EXAMPLE",
    "WindowTaskId": "4f7ca192-7e9a-40fe-9192-5cb15EXAMPLE",
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": [
                "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
            ]
        }
    ],
    "TaskArn": "AutoTestDoc",
    "ServiceRoleArn": "arn:aws:iam::111122223333:role/
MyMaintenanceWindowServiceRole",
    "TaskParameters": {},
    "TaskInvocationParameters": {
        "Automation": {
            "Parameters": {
                "multi": [
                    "{{WINDOW_TASK_ID}}"
                ],
                "single": [
                    "{{WINDOW_ID}}"
                ]
            }
        }
    },
    "Priority": 0,
    "MaxConcurrency": "10",
```

```
    "MaxErrors": "5",
    "Name": "My-Automation-Task",
    "Description": "A description for my Automation task"
}
```

Tutorial: Löschen eines Wartungsfensters mithilfe von AWS CLI

Um ein in diesen Tutorials erstelltes Wartungsfenster zu löschen, führen Sie den folgenden Befehl aus.

```
aws ssm delete-maintenance-window --window-id "mw-0c50858d01EXAMPLE"
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
{
   "WindowId":"mw-0c50858d01EXAMPLE"
}
```

Tutorial: Erstellen Sie ein Wartungsfenster zum Patchen über die Konsole

> **Important**
>
> Sie können dieses ältere Thema weiterhin zum Erstellen eines Wartungsfensters zum Patchen verwenden. Wir empfehlen jedoch, stattdessen eine Patch-Richtlinie zu verwenden. Weitere Informationen erhalten Sie unter Patch-Richtlinienkonfigurationen in Quick Setup und Das Patchen für Instances in einer Organisation mithilfe einer Patch-Richtlinie für Quick Setup konfigurieren.

Um die Auswirkungen auf die Verfügbarkeit Ihres Servers zu minimieren, empfehlen wir, ein Wartungsfenster zu konfigurieren, um die Patches dann einzuspielen, wenn der Geschäftsbetrieb dadurch nicht unterbrochen wird.

Sie müssen Rollen und Berechtigungen für Maintenance Windows ein Tool in konfigurieren AWS Systems Manager, bevor Sie mit diesem Verfahren beginnen. Weitere Informationen finden Sie unter Einrichten von Maintenance Windows.

So erstellen Sie ein Wartungsfenster für das Einspielen von Patches

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie Create maintenance window (Wartungsfenster erstellen) aus.
4. Geben Sie im Feld Name einen Namen ein, aus dem hervorgeht, dass das Wartungsfenster für das Einspielen von kritischen und wichtigen Updates verwendet wird.
5. (Optional) Geben Sie unter Description (Beschreibung) eine Beschreibung ein.
6. Wählen Sie Allow unregistered targets (Nicht registrierte Ziele erlauben), wenn Sie erlauben möchten, dass eine Wartungsfensteraufgabe auf verwalteten Knoten ausgeführt wird, obwohl diese Knoten nicht als Ziele registriert wurden.

   Falls Sie diese Option wählen, können Sie die nicht registrierten Knoten (nach Knoten-ID) auswählen, wenn Sie eine Aufgabe für das Wartungsfenster registrieren.

   Sollten Sie diese Option nicht wählen, müssen Sie die zuvor registrierten Ziele auswählen, wenn Sie eine Aufgabe für das Wartungsfenster registrieren.
7. Geben Sie oben im Abschnitt Schedule (Zeitplan) einen Zeitplan für das Wartungsfenster an, indem Sie eine der drei Planungsoptionen verwenden.

   Hinweise zum Erstellen von cron/rate Ausdrücken finden Sie unterReferenz: Cron- und Rate-Ausdrücke für System Manager.
8. Geben Sie unter Duration (Dauer) die Anzahl der Stunden ein, die das Wartungsfenster ausgeführt wird. Der Wert, den Sie angeben, bestimmt die spezifische Endzeit für das Wartungsfenster basierend auf dem Zeitpunkt, an dem es beginnt. Nach der resultierenden Endzeit dürfen keine Wartungsfenster-Aufgaben gestartet werden, abzüglich der Anzahl der Stunden, die Sie für Stop initiating tasks (Initiieren von Aufgaben beenden) im nächsten Schritt angeben.

   Beispiel: Wenn das Wartungsfenster um 15:00 Uhr beginnt, die Dauer drei Stunden beträgt und der Wert Stop initiating tasks (Initiieren von Aufgaben beenden) eine Stunde beträgt, können nach 17:00 Uhr keine Wartungsfenster-Aufgaben gestartet werden.
9. Geben Sie unter Stop initiating tasks (Initiieren von Aufgaben beenden) die Anzahl der Stunden für den Zeitpunkt vor dem Ende des Wartungsfensters an, ab dem vom System keine neuen auszuführenden Aufgaben mehr geplant werden sollen.

10. (Optional) Geben Sie unter Window start date (Startzeit des Fensters) ein Datum und eine Uhrzeit im erweiterten ISO-8601-Format an, zu dem bzw. der das Wartungsfenster aktiviert werden soll. Auf diese Weise können Sie die Aktivierung des Wartungsfensters bis zum angegebenen künftigen Zeitpunkt verzögern.
11. (Optional) Geben Sie unter Window end date (Enddatum des Fensters) ein Datum und eine Uhrzeit im erweiterten ISO-8601-Format an, zu dem bzw. der das Wartungsfenster deaktiviert werden soll. Auf diese Weise können Sie ein in der Zukunft liegendes Datum sowie eine Uhrzeit festlegen, nach dem das Wartungsfenster nicht mehr ausgeführt wird.
12. (Optional) Geben Sie unter Zeitzone planen die Zeitzone im IANA-Format (Internet Assigned Numbers Authority) an, auf der die Ausführung geplanter Wartungsfenster basieren soll. Zum Beispiel: "America/Los_Angeles", "etc/UTC", or "Asia/Seoul“.

    Weitere Informationen zu gültigen Formaten finden Sie unter Time Zone Database (Zeitzonendatenbank) auf der IANA-Website.
13. (Optional) Wenden Sie im Bereich „Tags verwalten“ ein oder mehrere name/value Tag-Schlüsselpaare auf das Wartungsfenster an.

    Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Sie können beispielsweise dieses Wartungsfenster mit Tags versehen, um die Aufgabentypen, die darin ausgeführt werden, zu identifizieren. In diesem Fall könnten Sie das folgende name/value Schlüsselpaar angeben:

    - `Key=TaskType,Value=Patching`
14. Wählen Sie Create maintenance window (Wartungsfenster erstellen) aus.
15. Wählen Sie in der Liste mit den Wartungsfenstern das gerade erstellte Wartungsfenster aus und klicken Sie anschließend auf Actions (Aktionen), Register targets (Ziele registrieren).
16. (Optional) Geben Sie im Abschnitt Maintenance window target details einen Namen, eine Beschreibung und Eigentümerinformationen (Ihren Namen oder Alias) für dieses Ziel an.
17. Wählen Sie für die Zielauswahl die Option Instance-Tags festlegen aus.
18. Geben Sie im Feld Instance-Tags angeben einen Tag-Schlüssel und einen Tag-Wert ein, um die Knoten zu identifizieren, die beim Wartungsfenster angemeldet werden sollen, und wählen Sie dann Hinzufügen.
19. Wählen Sie Register target. Das System erstellt ein Ziel für das Wartungsfenster.

20. Wählen Sie auf der Detailseite des von Ihnen erstellten Wartungsfensters Actions (Aktionen), Register Run command task (Ausführungsbefehlaufgabe registrieren) aus.
21. (Optional) Geben Sie im Abschnitt Maintenance window task details (Aufgabendetails für Wartungszeitraum) einen Namen und eine Beschreibung für diese Aufgabe an.
22. Wählen Sie unter Command document (Befehlsdokument) die Option `AWS-RunPatchBaseline` aus.
23. Wählen Sie für Task priority (Aufgabenpriorität) eine Priorität aus. Null (`0`) ist die höchste Priorität.
24. Wählen Sie für Targets (Ziele) unter Target by (Auswahl nach) das Wartungsfensterziel aus, das Sie zuvor erstellt haben.
25. Für Ratenregelung:

    - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

      > **Note**
      >
      > Wenn Sie Ziele ausgewählt haben, indem Sie Tags angegeben haben, die auf verwaltete Knoten angewendet wurden, oder indem Sie AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

    - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.
26. (Optional) Wählen Sie für IAM-Servicerolle eine Rolle aus, um Systems Manager Berechtigungen zur Übernahme zum Ausführen von Wartungsfenster-Aufgaben zu erteilen.

    Wenn Sie keinen ARN für die Servicerolle angeben, verwendet Systems Manager eine serviceverknüpfte Rolle in Ihrem Konto. Wenn in Ihrem Konto keine geeignete serviceverknüpfte Rolle für Systems Manager vorhanden ist, wird sie erstellt, wenn die Aufgabe erfolgreich registriert wurde.

> **Note**
>
> Um die Sicherheitslage zu verbessern, empfehlen wir dringend, eine benutzerdefinierte Richtlinie und eine benutzerdefinierte Servicerolle für die Ausführung Ihrer Aufgaben im Wartungsfenster zu erstellen. Die Richtlinie kann so gestaltet werden, dass sie nur die Berechtigungen gewährt, die für Ihre speziellen Wartungsfensteraufgaben erforderlich sind. Weitere Informationen finden Sie unter Einrichten von Maintenance Windows.

27. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

Um die Ausgabe in eine Amazon CloudWatch Logs-Protokollgruppe zu streamen, wählen Sie das CloudWatch Ausgabefeld aus. Geben Sie den Namen der Protokollgruppe in das Feld ein.

28. Aktivieren Sie das Kontrollkästchen Enable SNS notifications (SNS-Benachrichtigungen aktivieren) im Abschnitt SNS notifications (SNS-Benachrichtigungen), wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

29. Für Parameters (Parameter):

- Wählen Sie in der Liste Operation (Vorgang) die Option Scan (Scannen), um nach fehlenden Patches zu suchen, oder wählen Sie Install (Installieren), um nach fehlenden Patches zu suchen und diese direkt zu installieren.
- Sie brauchen keine Angaben für das Feld Snapshot Id (Snapshot-ID) zu machen. Das System generiert diesen Parameter automatisch und stellt ihn bereit.
- Sie müssen nichts in das Feld Install Override List (Überschreibungsliste installieren) eingeben, es sei denn, Sie möchten, dass Patch Manager einen anderen Patch als für die Patch-Baseline angegeben verwenden soll. Weitere Informationen finden Sie unter Parametername: `InstallOverrideList`.
- Geben Sie für an RebootOption, ob Knoten neu gestartet werden sollen, wenn während des `Install` Vorgangs Patches installiert werden, oder ob andere Patches Patch Manager erkannt werden, die seit dem letzten Knotenneustart installiert wurden. Weitere Informationen finden Sie unter Parametername: `RebootOption`.
- (Optional) Geben Sie im Feld Comment (Kommentar) eine Verfolgungsnotiz oder Erinnerung zu diesem Befehl ein.
- Geben Sie im Feld Timeout (seconds) (Timeout (Sekunden)) die Anzahl der Sekunden ein, die das System warten soll, bis der Vorgang beendet ist, bevor er als nicht erfolgreich eingestuft wird.

30. Wählen Sie Register Run command task.

Nachdem die Wartungsfensteraufgabe abgeschlossen ist, können Sie Details zur Patch-Compliance in der Systems-Manager-Konsole im Tool Fleet Manager anzeigen.

Sie können Konformitätsinformationen auch im Tool Patch Manager auf der Registerkarte Konformitätsberichte einsehen.

Sie können auch das DescribePatchGroupStateund verwenden DescribeInstancePatchStatesForPatchGroup APIs , um Konformitätsdetails einzusehen. Weitere Informationen zu Patch-Compliance-Daten finden Sie unter Info zu Patch Compliance.

Patching von Zeitplänen mithilfe von Wartungsfenstern

Nach der Konfiguration einer Patch-Baseline (und optional einer Patch-Gruppe), können Sie Patches für Ihren Knoten mithilfe eines Wartungsfensters einspielen. Ein Wartungsfenster kann die Auswirkungen bei der Serververfügbarkeit verringern, da Sie die Möglichkeit haben, eine Uhrzeit für

das Einspielen der Patches festzulegen, sodass der Geschäftsbetrieb nicht unterbrochen werden muss. Wartungsfenster funktionieren wie folgt:

1. Sie erstellen ein Wartungsfenster mit einem Zeitplan für Ihre Patching-Operationen.
2. Wählen Sie die Ziele für das Wartungsfenster aus, indem Sie das `PatchGroup` Tag `Patch Group` oder für den Tag-Namen und einen beliebigen Wert angeben, für den Sie Amazon Elastic Compute Cloud (Amazon EC2) -Tags definiert haben, z. B. „Webserver“ oder „US-EAST-PROD“. (Sie müssen ohne Leerzeichen angeben`PatchGroup`, ob Sie Tags in den Instanz-Metadaten zugelassen haben. EC2
3. Sie erstellen eine neue Aufgabe für das Wartungsfenster und geben für diese Aufgabe das Dokument `AWS-RunPatchBaseline` an.

Wenn Sie die Aufgabe konfigurieren, können Sie entweder Knoten scannen oder Patches scannen und auf den Knoten installieren. Wenn Sie die Knoten scannen, scannt Patch Manager, ein Tool in AWS Systems Manager, jeden Knoten und generiert eine Liste der fehlenden Patches für Sie zur Überprüfung.

Wenn Sie Patches scannen und installieren, scannt Patch Manager jeden Knoten und vergleicht die Liste der installierten Patches mit der Liste der genehmigten Patches in der Baseline. Patch Manager identifiziert fehlende Patches und lädt dann alle fehlenden und genehmigten Patches herunter und installiert sie.

Wenn Sie einen einmaligen Scan oder eine einmalige Installation ausführen möchten, um ein Problem zu beheben, können Sie Run Command für den direkten Aufruf des Dokuments `AWS-RunPatchBaseline` verwenden.

> **Important**
>
> Nach dem Installieren von Patches führt Systems Manager einen Neustart eines jeden Knotens durch. Der Neustart ist erforderlich, um sicherzustellen, dass die Patches ordnungsgemäß installiert sind, und um sicherzustellen, dass das System den Knoten nach dem Einspielen der Patches nicht in einem potenziell fehlerhaften Zustand zurücklässt. (Ausnahme: Wenn der `RebootOption`-Parameter im `NoReboot`-Dokument auf `AWS-RunPatchBaseline` gesetzt ist, wird der verwaltete Knoten nach der Ausführung von Patch Manager nicht neu gestartet. Weitere Informationen finden Sie unter Parametername: `RebootOption`.)

Verwendung von Pseudo-Parametern bei der Registrierung von Aufgaben im Wartungsfenster

Beim Registrieren einer Aufgabe in Maintenance Windows, einem Tool in AWS Systems Manager, geben Sie die Parameter an, die für jede der vier Arten von Aufgaben eindeutig sind. (In CLI-Befehlen werden diese mit der `--task-invocation-parameters`-Option bereitgestellt.)

Sie können auch mithilfe der Pseudoparameter-Syntax wie `{{RESOURCE_ID}}`, `{{TARGET_TYPE}}` und `{{WINDOW_TARGET_ID}}` auf bestimmte Werte verweisen. Während der Ausführung übergibt die Wartungsfenster-Aufgabe anstelle der Pseudoparameter-Platzhalter richtige Werte. Die vollständige Liste der verwendbaren Pseudoparameter finden Sie weiter unten in diesem Thema unter Unterstützte Pseudoparameter.

Important

Je nach dem für die Aufgabe erforderlichen ID-Format können Sie für den Zieltyp `RESOURCE_GROUP` auswählen, ob Sie die `{{TARGET_ID}}` und `{{RESOURCE_ID}}` zum Verweisen verwenden möchten, wenn Ihre Aufgabe ausgeführt wird. `{{TARGET_ID}}` gibt den vollständigen ARN der Ressource zurück. `{{RESOURCE_ID}}` gibt wie in diesen Beispielen gezeigt nur einen kürzeren Namen oder eine kürzere ID der Ressource zurück.

- `{{TARGET_ID}}`-Format: `arn:aws:ec2:us-east-1:123456789012:instance/i-02573cafcfEXAMPLE`
- `{{RESOURCE_ID}}`-Format: `i-02573cafcfEXAMPLE`

Für Zieltyp `INSTANCE` ergeben die Parameter `{{TARGET_ID}}` und `{{RESOURCE_ID}}` nur die Instance-ID. Weitere Informationen finden Sie unter Unterstützte Pseudoparameter. `{{TARGET_ID}}` und `{{RESOURCE_ID}}` werden verwendet, um IDs von AWS-Ressourcen nur an Automation-, Lambda- und Step Functions-Aufgaben zu übergeben. Diese beiden Pseudo-Parameter können nicht mit Run Command-Aufgaben verwendet werden.

Beispiele für Pseudoparameter

Angenommen, Ihre Nutzlast für eine AWS Lambda-Aufgabe muss anhand ihrer ID auf eine Instance verweisen.

Unabhängig davon, ob Sie ein Wartungsfensterziel `INSTANCE` oder `RESOURCE_GROUP` verwenden, kann dies mit dem `{{RESOURCE_ID}}`-Pseudoparameter erreicht werden. Zum Beispiel:

```
"TaskArn": "arn:aws:lambda:us-east-2:111122223333:function:SSMTestFunction",
    "TaskType": "LAMBDA",
    "TaskInvocationParameters": {
        "Lambda": {
            "ClientContext": "ew0KICAi--truncated--0KIEXAMPLE",
            "Payload": "{ \"instanceId\": \"{{RESOURCE_ID}}\" }",
            "Qualifier": "$LATEST"
        }
    }
```

Wenn Ihre Lambda-Aufgabe zusätzlich zu Amazon Elastic Compute Cloud (Amazon EC2)-Instances für einen anderen unterstützten Zieltyp, z. B. eine Amazon DynamoDB-Tabelle, ausgeführt werden soll, kann dieselbe Syntax verwendet werden und `{{RESOURCE_ID}}` ergibt nur den Namen der Tabelle. Wenn Sie jedoch den vollständigen ARN der Tabelle benötigen, verwenden Sie `{{TARGET_ID}}`, wie im folgenden Beispiel gezeigt.

```
"TaskArn": "arn:aws:lambda:us-east-2:111122223333:function:SSMTestFunction",
    "TaskType": "LAMBDA",
    "TaskInvocationParameters": {
        "Lambda": {
            "ClientContext": "ew0KICAi--truncated--0KIEXAMPLE",
            "Payload": "{ \"tableArn\": \"{{TARGET_ID}}\" }",
            "Qualifier": "$LATEST"
        }
    }
```

Dieselbe Syntax funktioniert, wenn Sie auf Instances oder andere Ressourcentypen abzielen. Wenn einer Ressourcengruppe mehrere Ressourcentypen hinzugefügt wurden, wird die Aufgabe für jede der entsprechenden Ressourcen ausgeführt.

> **Important**
>
> Nicht alle Ressourcentypen, die möglicherweise in eine Ressourcengruppe einbezogen werden, ergeben einen Wert für den `{{RESOURCE_ID}}`-Parameter. Eine Liste der unterstützten Ressourcentypen finden Sie unter Unterstützte Pseudoparameter.

Ein weiteres Beispiel: Um eine Automation-Aufgabe auszuführen, die Ihre EC2-Instances beendet, geben Sie das Systems Manager-Dokument (SSM-Dokument) `AWS-StopEC2Instance` als `TaskArn`-Wert an und verwenden Sie den Pseudoparameter `{{RESOURCE_ID}}`:

```
"TaskArn": "AWS-StopEC2Instance",
    "TaskType": "AUTOMATION"
    "TaskInvocationParameters": {
        "Automation": {
            "DocumentVersion": "1",
            "Parameters": {
                "instanceId": [
                    "{{RESOURCE_ID}}"
                ]
            }
        }
    }
```

Um eine Automatisierungsaufgabe auszuführen, die einen Snapshot eines Amazon Elastic Block Store (Amazon EBS)-Volumes kopiert, geben Sie das `AWS-CopySnapshot`-SSM-Dokument als `TaskArn`-Wert an und verwenden den Pseudoparameter „`{{RESOURCE_ID}}`“:

```
"TaskArn": "AWS-CopySnapshot",
    "TaskType": "AUTOMATION"
    "TaskInvocationParameters": {
        "Automation": {
            "DocumentVersion": "1",
            "Parameters": {
                "SourceRegion": "us-east-2",
                "targetType":"RESOURCE_GROUP",
                "SnapshotId": [
                    "{{RESOURCE_ID}}"
                ]
            }
        }
    }
```

Unterstützte Pseudoparameter

Die folgende Liste beschreibt die Pseudoparameter, die Sie mit der `{{`*`PSEUDO_PARAMETER`*`}}`-Syntax in der `--task-invocation-parameters`-Option angeben können.

- **`WINDOW_ID`**: Die ID des Ziel-Wartungsfensters.

- **`WINDOW_TASK_ID`**: Die ID der Fensteraufgabe, die ausgeführt wird.
- **`WINDOW_TARGET_ID`**: Die ID des Fensterziels, die das Ziel (die Ziel-ID) umfasst.
- **`WINDOW_EXECUTION_ID`**: Die ID der aktuellen Fensterausführung.
- **`TASK_EXECUTION_ID`**: Die ID der aktuellen Aufgabenausführung.
- **`INVOCATION_ID`**: Die ID des aktuellen Aufrufs.
- **`TARGET_TYPE`**: Der Zieltyp. Unterstützte Typen sind u. a.: `RESOURCE_GROUP` und `INSTANCE`.
- **`TARGET_ID`**:

  Wenn der angegebene Zieltyp „`INSTANCE`“ lautet, wird der Pseudoparameter „`TARGET_ID`“ durch die ID der Instance ersetzt. Beispiel, `i-078a280217EXAMPLE`.

  Wenn der angegebene Zieltyp „`RESOURCE_GROUP`“ lautet, ist der für die Aufgabenausführung referenzierte Wert der vollständige ARN der Ressource. Beispiel: `arn:aws:ec2:us-east-1:123456789012:instance/i-078a280217EXAMPLE`. Die folgende Tabelle enthält `TARGET_ID`-Beispielwerte für bestimmte Ressourcentypen in einer Ressourcengruppe.

  > **Note**
  >
  > `TARGET_ID` wird nicht für Run Command-Aufgaben unterstützt.

Resource type (Ressourcentyp)	Beispiel-TARGET_ID
`AWS::CloudWatch::Alarm`	`arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyCloudWatchAlarm i-078a280217EXAMPLE`
`AWS::DynamoDB::Table`	`arn:aws:dynamodb:us-east-1:123456789012:table/MyTable`
`AWS::EC2::Instance`	`arn:aws:ec2:us-east-1:123456789012:i`

Resource type (Ressourc entyp)	Beispiel-TARGET_ID	
	nstance/ i-078a280 217EXAMPLE	
AWS::EC2::Image	arn:aws:ec2:us-eas t-1:123456789012:i mage/ami-02250b373 2EXAMPLE	
AWS::EC2::Security Group	arn:aws:ec2:us-eas t-1:123456789012:s ecurity-group/sg-c EXAMPLE	
AWS::EC2::Snapshot	arn:aws:ec2:us-eas t-1:123456789012:s napshot/snap-03866 bf003EXAMPLE	
AWS::EC2::Volume	arn:aws:ec2:us-eas t-1:123456789012:v olume/vol-0912e04d 78EXAMPLE	
AWS::ECS::Service	arn:aws:ecs:us-eas t-1:123456789012:s ervice/my-ecs-serv ice	
AWS::RDS::DBCluster	arn:aws:rds:us-eas t-2:123456789012:c luster:My-Cluster	
AWS::RDS::DBInstance	arn:aws:rds:us-eas t-1:123456789012:d b:My-SQL-Instance	

Resource type (Ressourc entyp)	Beispiel-TARGET_ID	
AWS::S3::Bucket	arn:aws:s3:::amzn-s3-demo-bucket	
AWS::SSM::ManagedInstance	arn:aws:ssm:us-east-1:123456789012:managed-instance/mi-0feadcf2d9EXAMPLE	

- **RESOURCE_ID**: Die kurze ID eines Ressourcentyps, der in einer Ressourcengruppe enthalten ist. Die folgende Tabelle enthält RESOURCE_ID-Beispielwerte für bestimmte Ressourcentypen in einer Ressourcengruppe.

> **Note**
>
> RESOURCE_ID wird nicht für Run Command-Aufgaben unterstützt.

Resource type (Ressourc entyp)	Beispiel-RESOURCE_ID	
AWS::CloudWatch::Alarm	MyCloudWatchAlarm	
AWS::DynamoDB::Table	MyTable	
AWS::EC2::Instance	i-078a280217EXAMPLE	
AWS::EC2::Image	ami-02250b3732EXAMPLE	
AWS::EC2::SecurityGroup	sg-cEXAMPLE	
AWS::EC2::Snapshot	snap-03866bf003EXAMPLE	

Resource type (Ressourc entyp)	Beispiel-RESOURCE_ID	
AWS::EC2::Volume	vol-0912e04d78EXAMPLE	
AWS::ECS::Service	my-ecs-service	
AWS::RDS::DBCluster	My-Cluster	
AWS::RDS::DBInstance	My-SQL-Instance	
AWS::S3::Bucket	amzn-s3-demo-bucket	
AWS::SSM::ManagedInstance	mi-0feadcf2d9EXAMPLE	

> **Note**
>
> Wenn die von Ihnen angegebene AWS-Ressourcengruppe Ressourcentypen enthält, die keinen RESOURCE_ID-Wert ergeben und in der vorangegangenen Tabelle nicht aufgeführt sind, wird der RESOURCE_ID-Parameter nicht ausgefüllt. Für diese Ressource wird weiterhin ein Ausführungsaufruf ausgeführt. Verwenden Sie in diesen Fällen stattdessen Pseudoparameter „TARGET_ID“, der durch den vollständigen ARN der Ressource ersetzt wird.

Wartungsfenster-Optionen für Planung und aktive Zeiträume

Wenn Sie ein Wartungsfenster erstellen, müssen Sie angeben, wie oft das Wartungsfenster ausgeführt werden soll. Verwenden Sie dazu einen Cron- oder Rate-Ausdruck. Optional können Sie einen Datumsbereich angeben, in dem das Wartungsfenster nach seinem regulären Zeitplan laufen kann, sowie eine Zeitzone, auf der dieser reguläre Zeitplan basieren soll.

Beachten Sie jedoch, dass die Zeitzonenoption und die Optionen für Start- und Enddatum voneinander unabhängig sind. Das von Ihnen angegebene Start- und Enddatum (mit oder ohne einen Versatz für Ihre Zeitzone) bestimmt ausschließlich den gültigen Zeitraum, während dem das Wartungsfenster entsprechend seinem Zeitplan ausgeführt werden kann. Die Zeitzonenoption

bestimmt die internationale Zeitzone, auf dessen Basis der Wartungsfenster-Zeitplan während seines gültigen Zeitraums ausgeführt wird.

> **Note**
>
> Sie geben das Start- und Enddatum im ISO-8601-Zeitstempelformat an. Beispiel: `2021-04-07T14:29:00-08:00`
> Sie geben Zeitzonen in Internet Assigned Numbers Authority (IANA)-Format an. Beispiel: `America/Chicago`, `Europe/Berlin` oder `Asia/Tokyo`.

Beispiele

- Beispiel 1: Angeben eines Startdatums für das Wartungsfenster
- Beispiel 2: Angeben eines Start- und Enddatums für das Wartungsfenster
- Beispiel 3: Erstellen eines Wartungsfensters, das nur einmal ausgeführt wird
- Beispiel 4: Angeben der Anzahl der Zeitplanversatztage für ein Wartungsfenster

Beispiel 1: Angeben eines Startdatums für das Wartungsfenster

Angenommen, Sie verwenden die AWS Command Line Interface (AWS CLI) zum Erstellen eines Wartungsfensters mit den folgenden Optionen:

- `--start-date 2021-01-01T00:00:00-08:00`
- `--schedule-timezone "America/Los_Angeles"`
- `--schedule "cron(0 09 ? * WED *)"`

Zum Beispiel:

Linux & macOS

```
aws ssm create-maintenance-window \
    --name "My-LAX-Maintenance-Window" \
    --allow-unassociated-targets \
    --duration 3 \
    --cutoff 1 \
    --start-date 2021-01-01T00:00:00-08:00 \
    --schedule-timezone "America/Los_Angeles" \
```

```
    --schedule "cron(0 09 ? * WED *)"
```

Windows

```
aws ssm create-maintenance-window ^
    --name "My-LAX-Maintenance-Window" ^
    --allow-unassociated-targets ^
    --duration 3 ^
    --cutoff 1 ^
    --start-date 2021-01-01T00:00:00-08:00 ^
    --schedule-timezone "America/Los_Angeles" ^
    --schedule "cron(0 09 ? * WED *)"
```

Das bedeutet, dass der erste Durchlauf des Wartungsfensters erst nach dem angegebenen Startdatum und -zeitpunkt, d. h. am Freitag, dem 1. Januar 2021, um 12:00 Uhr US-Pazifikzeit, stattfinden wird. (Diese Zeitzone liegt acht Stunden hinter der UTC-Zeit.) In diesem Fall entsprechen das Startdatum und die Startzeit des Zeitfensters nicht dem Zeitpunkt, zu dem das Wartungsfenster zum ersten Mal läuft. Zusammen betrachtet bedeuten die Werte `--schedule-timezone` und `--schedule`, dass das Wartungsfenster jeden Mittwoch um 9:00 Uhr in der US Pacific-Zeitzone ausgeführt wird (angegeben durch "Amerika/Los Angeles" im IANA-Format). Die erste Ausführung im aktivierten Zeitraum erfolgt Mittwoch, 4. Januar 2021, um 9.00 Uhr US Pacific-Zeitzone.

Beispiel 2: Angeben eines Start- und Enddatums für das Wartungsfenster

In diesem Beispiel gehen wir davon aus, dass Sie als Nächstes ein Wartungsfenster mit diesen Optionen erstellen:

- `--start-date 2019-01-01T00:03:15+09:00`
- `--end-date 2019-06-30T00:06:15+09:00`
- `--schedule-timezone "Asia/Tokyo"`
- `--schedule "rate(7 days)"`

Zum Beispiel:

Linux & macOS

```
aws ssm create-maintenance-window \
    --name "My-NRT-Maintenance-Window" \
```

```
    --allow-unassociated-targets \
    --duration 3 \
    --cutoff 1 \
    --start-date 2019-01-01T00:03:15+09:00 \
    --end-date 2019-06-30T00:06:15+09:00 \
    --schedule-timezone "Asia/Tokyo" \
    --schedule "rate(7 days)"
```

Windows

```
aws ssm create-maintenance-window ^
    --name "My-NRT-Maintenance-Window" ^
    --allow-unassociated-targets ^
    --duration 3 ^
    --cutoff 1 ^
    --start-date 2019-01-01T00:03:15+09:00 ^
    --end-date 2019-06-30T00:06:15+09:00 ^
    --schedule-timezone "Asia/Tokyo" ^
    --schedule "rate(7 days)"
```

Der aktivierte Zeitraum für dieses Wartungsfenster beginnt am 1. Januar 2019 um 3:15 Uhr japanische Standardzeit. Der gültige Zeitraum für dieses Wartungsfenster endet am Sonntag, 30. Juni 2019 um 6:15 Uhr japanische Standardzeit. (Diese Zeitzone liegt neun Stunden vor der UTC-Zeit.) Zusammen betrachtet bedeuten die Werte `--schedule-timezone` und `--schedule`, dass das Wartungsfenster jeden Dienstag um 3:15 Uhr in der japanischen Standardzeitzone ausgeführt wird (angegeben durch "Asien/Tokio" im IANA-Format). Der Grund hierfür ist, dass das Wartungsfenster alle sieben Tage ausgeführt wird und am Dienstag, 1. Januar um 3:15 Uhr aktiv wird. Die letzte Ausführung erfolgt am Dienstag, 25. Juni 2019 um 3:15 Uhr japanische Standardzeit. Dies ist der letzte Dienstag bevor der aktivierte Zeitraum für das Wartungsfenster fünf Tage später endet.

Beispiel 3: Erstellen eines Wartungsfensters, das nur einmal ausgeführt wird

Jetzt erstellen Sie ein Wartungsfenster mit dieser Option:

- `--schedule "at(2020-07-07T15:55:00)"`

Zum Beispiel:

Linux & macOS

```
aws ssm create-maintenance-window \
    --name "My-One-Time-Maintenance-Window" \
    --schedule "at(2020-07-07T15:55:00)" \
    --duration 5 \
    --cutoff 2 \
    --allow-unassociated-targets
```

Windows

```
aws ssm create-maintenance-window ^
    --name "My-One-Time-Maintenance-Window" ^
    --schedule "at(2020-07-07T15:55:00)" ^
    --duration 5 ^
    --cutoff 2 ^
    --allow-unassociated-targets
```

Dieses Wartungsfenster wird nur einmal ausgeführt und zwar am 7. Juli 2020 um 15:55 Uhr UTC-Zeit. Das Wartungsfenster wurde aktiviert, um bei Bedarf bis zu fünf Stunden ausgeführt zu werden, jedoch können zwei Stunden vor dem Ende des Wartungsfensters keine neuen Aufgaben mehr gestartet werden.

Beispiel 4: Angeben der Anzahl der Zeitplanversatztage für ein Wartungsfenster

Jetzt erstellen Sie ein Wartungsfenster mit dieser Option:

```
--schedule-offset 2
```

Zum Beispiel:

Linux & macOS

```
aws ssm create-maintenance-window \
    --name "My-Cron-Offset-Maintenance-Window" \
    --schedule "cron(0 30 23 ? * TUE#3 *)" \
    --duration 4 \
    --cutoff 1 \
    --schedule-offset 2 \
    --allow-unassociated-targets
```

Windows

```
aws ssm create-maintenance-window ^
    --name "My-Cron-Offset-Maintenance-Window" ^
    --schedule "cron(0 30 23 ? * TUE#3 *)" ^
    --duration 4 ^
    --cutoff 1 ^
    --schedule-offset 2 ^
    --allow-unassociated-targets
```

Ein Zeitplanversatz ist die Anzahl der Tage, die nach dem über einen CRON-Ausdruck angegebenen Datum und der angegebenen Uhrzeit gewartet werden soll, bevor das Wartungsfenster ausgeführt wird.

Im vorhergegangenen Beispiel wird mit dem CRON-Ausdruck die Ausführung eines Wartungsfensters um 23.30 Uhr am dritten Dienstag jedes Monats geplant:

```
--schedule "cron(0 30 23 ? * TUE#3 *)
```

Die Einbeziehung von `--schedule-offset 2` bedeutet allerdings, dass das Wartungsfenster erst um 23.30 Uhr zwei Tage nach dem dritten Dienstag jedes Monats ausgeführt wird.

Zeitplanversätze werden nur für CRON Ausdrücke unterstützt.

Weitere Informationen

- Referenz: Cron- und Rate-Ausdrücke für System Manager
- Erstellen eines Wartungsfensters mit der Konsole
- Tutorial: Erstellen und konfigurieren Sie ein Wartungsfenster mit dem AWS CLI
- CreateMaintenanceWindow in der AWS Systems Manager-API-Referenz
- create-maintenance-window im Abschnitt AWS Systems Manager der AWS CLI-Befehlsreferenz
- Zeitzonendatenbank auf der IANA-Website

Wartungsfenster-Tasks ohne Ziele registrieren

Für jedes von Ihnen erstellte Wartungsfenster können Sie eine oder mehrere Aufgaben angeben, die beim Ausführen des Wartungsfensters ausgeführt werden sollen. In den meisten Fällen müssen

Sie die Ressourcen oder Ziele angeben, für die Aufgabe ausgeführt werden soll. In einigen Fällen müssen Sie Ziele jedoch nicht explizit in der Aufgabe angeben.

Ein oder mehrere Ziele für Wartungsfenster Systems Manager Run Command-Typ-Aufgaben müssen angegeben werden. Je nach Art der Aufgabe sind Ziele für andere Aufgabentypen im Wartungsfenster (Systems Manager Automation AWS Lambda, und AWS Step Functions) optional.

Bei den Aufgabentypen Lambda und Step Functions hängt es vom Inhalt der von Ihnen erstellten Funktion oder des Zustandsautomaten ab, ob ein Ziel erforderlich ist.

> **Note**
>
> Wenn eine Aufgabe registrierte Ziele hat, lösen Automation und AWS Step Functions Aufgaben die Ziele aus Ressourcengruppen und Tags auf und senden einen Aufruf pro aufgelöster Ressource, was zu mehreren Aufgabenaufrufen führt. AWS Lambda Aber angenommen, Sie möchten nur einen Aufruf für eine Lambda-Aufgabe, die bei einer Ressourcengruppe mit mehr als einer Instance registriert ist. Wenn Sie in diesem Fall in der AWS-Managementkonsole arbeiten, wählen Sie auf der Seite Lambda-Aufgabe registrieren oder Lambda-Aufgabe bearbeiten die Option Aufgabenziel nicht erforderlich aus. Wenn Sie den AWS CLI Befehl verwenden, geben Sie bei der Ausführung des Befehls oder register-task-with-maintenance-windowBefehls keine Ziele mithilfe des `--targets` Parameters an. update-maintenance-window-task

In vielen Fällen müssen Sie ein Ziel für eine Automatisierungsaufgabe nicht explizit angeben. Angenommen, Sie erstellen beispielsweise eine Automation-Aufgabe, um eine Amazon Machine Image (AMI) für Linux mit dem `AWS-UpdateLinuxAmi`-Runbook zu aktualisieren. Wenn die Aufgabe ausgeführt wird, wird AMI mit den neuesten verfügbaren Linux-Verteilungspaketen und Amazon-Software aktualisiert. Neue Instances, die aus der AMI erstellt wurden, haben diese Updates bereits installiert. Da die ID des AMI in den Eingabeparametern für das Runbook angegeben ist, muss in der Wartungsfenster-Aufgabe kein Ziel erneut angegeben werden.

Angenommen, Sie verwenden AWS Command Line Interface (AWS CLI), um eine Automatisierungsaufgabe im Wartungsfenster zu registrieren, die das `AWS-RestartEC2Instance` Runbook verwendet. Da der neu zu startende Knoten im `--task-invocation-parameters`-Argument angegeben wird, müssen Sie nicht auch eine `--targets`-Option angeben.

> **Note**
>
> Bei Wartungsfensteraufgaben ohne festgelegtes Ziel können Sie keine Werte für `--max-errors` und `--max-concurrency` bereitstellen. Stattdessen fügt das System den Platzhalterwert 1 ein, der in der Antwort auf Befehle wie describe-maintenance-window-tasks und get-maintenance-window-task gemeldet wird. Diese Werte wirken sich nicht auf die Ausführung Ihrer Aufgabe aus und können ignoriert werden.

Das folgende Beispiel zeigt auch, dass die `--targets`, `--max-errors` und `--max-concurrency`-Optionen für eine ziellose Wartungsfensteraufgabe weggelassen werden.

Linux & macOS

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --service-role-arn "arn:aws:iam::123456789012:role/
MaintenanceWindowAndAutomationRole" \
    --task-type "AUTOMATION" \
    --name "RestartInstanceWithoutTarget" \
    --task-arn "AWS-RestartEC2Instance" \
    --task-invocation-parameters "{\"Automation\":{\"Parameters\":{\"InstanceId\":
[\"i-02573cafcfEXAMPLE\"]}}}" \
    --priority 10
```

Windows

```
aws ssm register-task-with-maintenance-window ^
    --window-id "mw-ab12cd34eEXAMPLE" ^
    --service-role-arn "arn:aws:iam::123456789012:role/
MaintenanceWindowAndAutomationRole" ^
    --task-type "AUTOMATION" ^
    --name "RestartInstanceWithoutTarget" ^
    --task-arn "AWS-RestartEC2Instance" ^
    --task-invocation-parameters "{\"Automation\":{\"Parameters\":{\"InstanceId\":
[\"i-02573cafcfEXAMPLE\"]}}}" ^
    --priority 10
```

> **Note**
>
> Für Aufgaben im Wartungsfenster, die vor dem 23. Dezember 2020 registriert wurden: Wenn Sie Ziele für die Aufgabe angegeben haben und eines nicht mehr erforderlich ist, können Sie diese Aufgabe aktualisieren, um die Ziele mithilfe der Systems Manager Manager-Konsole oder des update-maintenance-window-task AWS CLI Befehls zu entfernen.

Weitere Informationen

- Fehlermeldungen: „Wartungsfensteraufgaben ohne Ziele unterstützen keine Werte“ und „Wartungsfensteraufgaben ohne Ziele unterstützen keine Werte“ MaxConcurrency MaxErrors

Fehlerbehebung bei Wartungsfenstern

Im Folgenden finden Sie Informationen zur Behandlung von Problemen mit Wartungsfenstern.

Themen

- Aufgabenfehler bearbeiten: Auf der Seite zur Bearbeitung einer Wartungsfensteraufgabe gibt die IAM-Rollenliste eine Fehlermeldung aus: „Wir konnten die für diese Aufgabe spezifizierte IAM-Wartungsfensterrolle nicht finden. Sie wurde möglicherweise gelöscht oder noch nicht erstellt.“
- Nicht alle Wartungsfensterziele werden aktualisiert
- Die Aufgabe schlägt mit dem Aufrufstatus der Aufgabe fehl: „Die bereitgestellte Rolle enthält nicht die richtigen SSM-Berechtigungen.“
- Aufgabe schlägt mit der Fehlermeldung „Step fails when it is validating and resolving the step inputs (Schritt schlägt fehl, wenn die Schritteingaben überprüft und gelöst werden)“ fehl.
- Fehlermeldungen: „Wartungsfensteraufgaben ohne Ziele unterstützen keine Werte“ und „Wartungsfensteraufgaben ohne Ziele unterstützen keine Werte“ MaxConcurrency MaxErrors

Aufgabenfehler bearbeiten: Auf der Seite zur Bearbeitung einer Wartungsfensteraufgabe gibt die IAM-Rollenliste eine Fehlermeldung aus: „Wir konnten die für diese Aufgabe spezifizierte IAM-Wartungsfensterrolle nicht finden. Sie wurde möglicherweise gelöscht oder noch nicht erstellt.“

Problem 1: Die ursprünglich angegebene AWS Identity and Access Management (IAM-) Wartungsfensterrolle wurde gelöscht, nachdem Sie die Aufgabe erstellt haben.

Mögliche Lösung: 1) Wählen Sie eine andere IAM-Wartungsfenster-Rolle aus, falls eine solche in Ihrem Konto vorhanden ist, oder erstellen Sie eine neue und wählen Sie sie für die Aufgabe aus.

Problem 2: Wenn die Aufgabe mit dem AWS Command Line Interface (AWS CLI), oder einem AWS SDK erstellt wurde AWS Tools for Windows PowerShell, hätte ein Rollenname für das IAM-Wartungsfenster angegeben werden können, der nicht existiert. Beispielsweise könnte die IAM-Wartungsfensterrolle gelöscht worden sein, bevor Sie die Aufgabe erstellt haben, oder der Rollenname könnte falsch eingegeben worden sein, z. B. **`myrole`** anstelle von **`my-role`**.

Mögliche Lösung: Wählen Sie den richtigen Namen der IAM-Wartungsfensterrolle aus, die Sie verwenden möchten, oder erstellen Sie eine neue, die Sie für die Aufgabe angeben können.

Nicht alle Wartungsfensterziele werden aktualisiert

Problem: Sie stellen fest, dass die Wartungsfensteraufgaben nicht auf allen Ressourcen ausgeführt wurden, auf die Ihr Wartungsfenster abzielt. Beispiel: In den Ausführungsergebnissen des Wartungsfensters wird die Aufgabe für diese Ressource beispielsweise als fehlgeschlagen oder zeitlich abgelaufen markiert.

Lösung: Die häufigsten Gründe für das Nicht-Ausführen einer Wartungsfensteraufgabe auf einer Zielressource, sind Konnektivität und Verfügbarkeit. Zum Beispiel:

- Systems Manager hat die Verbindung zur Ressource vor oder während des Wartungsfenstervorgangs unterbrochen.
- Die Ressource war offline oder wurde während des Wartungsfenstervorgangs beendet.

Sie können warten, bis die Zeit der nächsten geplanten Wartungsfensteraufgabe für die Ressourcen ausgeführt wird. Sie können die Wartungsfensteraufgabe manuell für die Ressourcen ausführen, die nicht verfügbar waren oder offline waren.

Die Aufgabe schlägt mit dem Aufrufstatus der Aufgabe fehl: „Die bereitgestellte Rolle enthält nicht die richtigen SSM-Berechtigungen.“

Problem: Sie haben eine Wartungsfenster-Servicerolle für eine Aufgabe angegeben, aber die Aufgabe wird nicht erfolgreich ausgeführt, und der Aufgabenaufrufstatus meldet, dass „die bereitgestellte Rolle nicht die richtigen SSM-Berechtigungen enthält.“

- Solution (Lösung): In Aufgabe 1: Erstellen einer benutzerdefinierten Servicerolle für Wartungsfenster-Aufgaben stellen wir eine grundlegende Richtlinie zur Verfügung, die Sie an Ihre

benutzerdefinierte Wartungsfenster-Servicerolle anhängen können. Die Richtlinie enthält die für viele Aufgabenszenarien erforderlichen Berechtigungen. Aufgrund der Vielzahl von Aufgaben, die Sie ausführen können, müssen Sie jedoch möglicherweise zusätzliche Berechtigungen in der Richtlinie für Ihre Wartungsfenster-Rolle angeben.

Beispielsweise funktionieren einige Automatisierungsaktionen mit AWS CloudFormation Stacks. Daher müssen Sie möglicherweise die zusätzlichen Berechtigungen `cloudformation:CreateStack`, `cloudformation:DescribeStacks` und `cloudformation:DeleteStack` in die Richtlinie für Ihre Wartungsfenster-Servicerolle aufnehmen.

Als weiteres Beispiel benötigt das Automation-Runbook `AWS-CopySnapshot` Berechtigungen zum Erstellen eines Amazon Elastic Block Store (Amazon EBS)-Snapshots. Daher müssen Sie möglicherweise die `ec2:CreateSnapshot`-Berechtigung hinzufügen.

Informationen zu den Rollenberechtigungen, die für ein AWS verwaltetes Automation-Runbook erforderlich sind, finden Sie in den Runbook-Beschreibungen in der AWS Systems Manager Automation Runbook Reference.

Informationen zu den Rollenberechtigungen, die für ein AWS verwaltetes SSM-Dokument erforderlich sind, finden Sie im Inhalt des Dokuments im Bereich Dokumente in der Systems Manager Manager-Konsole.

Informationen zu den Rollenberechtigungen, die für Step-Functions-Aufgaben, Lambda-Aufgaben, benutzerdefinierte Automation-Runbooks und SSM-Dokumente erforderlich sind, erhalten Sie beim Autor dieser Ressourcen über die Berechtigungsanforderungen.

Aufgabe schlägt mit der Fehlermeldung „Step fails when it is validating and resolving the step inputs (Schritt schlägt fehl, wenn die Schritteingaben überprüft und gelöst werden)“ fehl.

Problem: Ein Automation-Runbook oder Systems Manager-Befehlsdokument, das Sie in einer Aufgabe verwenden, erfordert, dass Sie Eingaben wie `InstanceId` oder `SnapshotId` angeben, aber ein Wert wird nicht angegeben oder nicht korrekt angegeben.

- Lösung 1: Wenn Ihr Vorgang auf eine einzelne Ressource ausgerichtet ist, z. B. ein einzelner Knoten oder ein einzelner Snapshot, geben Sie die ID in die Eingabeparameter für die Aufgabe ein.
- Lösung 2: Wenn Ihre Aufgabe auf mehrere Ressourcen abzielt, z. B. das Erstellen von Bildern aus mehreren Knoten, wenn Sie das Runbook verwenden`AWS-CreateImage`, können Sie in den

Eingabeparametern einen der Pseudoparameter verwenden, die für Wartungsfensteraufgaben unterstützt werden, um den Knoten IDs im Befehl darzustellen.

Die folgenden Befehle registrieren eine Systems Manager Automation-Aufgabe mit einem Wartungsfenster unter Verwendung der Option AWS CLI. Der `--targets`-Wert gibt eine Ziel-ID für das Wartungsfenster an. Auch wenn der `--targets`-Parameter eine Ziel-ID des Fensters angibt, erfordern Parameter des Automatisierung-Runbooks, dass eine Knoten-ID angegeben wird. In diesem Fall verwendet der Befehl den Pseudo-Parameter `{{RESOURCE_ID}}` als `InstanceId` Wert.

AWS CLI Befehl:

Linux & macOS

Mit dem folgenden Beispielbefehl werden Amazon Elastic Compute Cloud (Amazon EC2) -Instances neu gestartet, die zur Zielgruppe Maintenance Window mit der ID E32EECB2-646C-4F4B-8ED1-205FBExample gehören.

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --targets Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE \
    --task-arn "AWS-RestartEC2Instance" \
    --service-role-arn arn:aws:iam::123456789012:role/
MyMaintenanceWindowServiceRole \
    --task-type AUTOMATION \
    --task-invocation-parameters
 "Automation={DocumentVersion=5,Parameters={InstanceId='{{RESOURCE_ID}}'}}" \
    --priority 0 --max-concurrency 10 --max-errors 5 --name "My-Restart-EC2-
Instances-Automation-Task" \
    --description "Automation task to restart EC2 instances"
```

Windows

```
aws ssm register-task-with-maintenance-window ^
    --window-id "mw-0c50858d01EXAMPLE" ^
    --targets Key=WindowTargetIds,Values=e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE ^
    --task-arn "AWS-RestartEC2Instance" ^
    --service-role-arn arn:aws:iam::123456789012:role/
MyMaintenanceWindowServiceRole ^
    --task-type AUTOMATION ^
    --task-invocation-parameters
 "Automation={DocumentVersion=5,Parameters={InstanceId='{{RESOURCE_ID}}'}}" ^
```

```
    --priority 0 --max-concurrency 10 --max-errors 5 --name "My-Restart-EC2-
Instances-Automation-Task" ^
    --description "Automation task to restart EC2 instances"
```

Weitere Informationen zur Arbeit mit Pseudoparametern für Wartungsfensteraufgaben finden Sie unter Verwendung von Pseudo-Parametern bei der Registrierung von Aufgaben im Wartungsfenster und Beispiele der Aufgabenregistrierung.

Fehlermeldungen: „Wartungsfensteraufgaben ohne Ziele unterstützen keine Werte“ und „Wartungsfensteraufgaben ohne Ziele unterstützen keine Werte“ MaxConcurrency MaxErrors

Problem: Wenn Sie eine Run Command-Typ-Aufgabe registrieren, müssen Sie mindestens ein Ziel angeben, auf dem die Aufgabe ausgeführt werden soll. Bei anderen Aufgabentypen (Automatisierung AWS Lambda, und AWS Step Functions) sind Ziele je nach Art der Aufgabe optional. Die Optionen `MaxConcurrency` (die Anzahl der Ressourcen, auf denen eine Aufgabe gleichzeitig ausgeführt werden soll) und `MaxErrors` (die Anzahl der Fehlschläge, nach denen die Aufgabe auf den Zielressourcen ausgeführt werden soll, bevor die Aufgabe fehlschlägt) sind für Wartungsfensteraufgaben, die keine Ziele angeben, nicht erforderlich oder werden nicht unterstützt. Das System generiert diese Fehlermeldungen, wenn für eine dieser Optionen Werte angegeben werden, wenn kein Aufgabenziel angegeben ist.

Lösung: Wenn einer dieser Fehler angezeigt wird, entfernen Sie die Werte für Parallelität und Fehlerschwellenwert, bevor Sie mit der Registrierung oder Aktualisierung der Wartungsfensteraufgabe fortfahren.

Weitere Informationen zur Ausführung von Aufgaben, die keine Ziele angeben, finden Sie unter Wartungsfenster-Tasks ohne Ziele registrieren im AWS Systems Manager -Benutzerhandbuch.

AWS Systems Manager Quick Setup

Verwenden SieQuick Setup, ein Tool in AWS Systems Manager, um häufig verwendete Amazon Web Services und -Funktionen schnell mit empfohlenen Best Practices zu konfigurieren. Quick Setupvereinfacht die Einrichtung von Diensten, einschließlich Systems Manager, durch die Automatisierung häufiger oder empfohlener Aufgaben. Zu diesen Aufgaben gehören beispielsweise die Erstellung der erforderlichen Instanzprofilrollen AWS Identity and Access Management (IAM) und die Einrichtung betrieblicher Best Practices wie regelmäßige Patchscans und Inventarerfassung. Bei der Nutzung des Quick Setup-Service fallen keine Kosten an. Abhängig von der Art der von Ihnen eingerichteten Services und den Nutzungsbeschränkungen können jedoch Kosten anfallen,

ohne dass Gebühren für die Services anfallen, die für die Einrichtung Ihres Services verwendet werden. Um mit Quick Setup zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Quick Setup aus.

> **Note**
>
> Wenn Sie zu Quick Setup weitergeleitet wurden, um Ihre Instances für die Verwaltung durch Systems Manager zu konfigurieren, führen Sie das Verfahren in Amazon-EC2-Host-Verwaltung mit Quick Setup einrichten aus.

Was sind die Vorteile von Quick Setup?

Quick Setup bietet folgende Vorteile:

- Vereinfachte Service- und Featurekonfiguration

  Quick Setup führt Sie durch die Konfiguration betrieblicher bewährter Methoden und stellt diese Konfigurationen automatisch bereit. Das Quick Setup-Dashboard zeigt eine Echtzeitansicht Ihres Konfigurationsbereitstellungsstatus an.

- Automatisches Bereitstellen von Konfigurationen über mehrere Konten hinweg

  Sie können es einzeln AWS-Konto oder Quick Setup in mehreren verwenden AWS-Konten und AWS-Regionen durch Integration mit AWS Organizations. Durch Verwenden von Quick Setup über mehrere Konten hinweg können Sie sicherstellen, dass Ihre Organisation konsistente Konfigurationen aufrechterhält.

- Beseitigen von Konfigurationsabweichungen

  Die Konfigurationsabweichung tritt auf, wenn ein Benutzer Änderungen an einem Service oder Feature vornimmt, die mit der Auswahl über Quick Setup im Konflikt stehen. Quick Setup überprüft regelmäßig auf Konfigurationsabweichung und versucht, diese zu beheben.

An wen richtet sich Quick Setup?

Quick Setup ist besonders vorteilhaft für Kunden, die bereits Erfahrung mit den von ihnen eingerichteten Services und Features haben und ihren Einrichtungsprozess vereinfachen möchten. Wenn Sie mit dem, mit dem AWS-Service Sie die Konfiguration durchführen, nicht vertraut sindQuick

Setup, empfehlen wir Ihnen, mehr über den Service zu erfahren. Überprüfen Sie den Inhalt im entsprechenden Benutzerhandbuch, bevor Sie eine Konfiguration mit Quick Setuperstellen.

Verfügbarkeit von Quick Setup in AWS-Regionen

Im Folgenden AWS-Regionen können Sie alle Quick Setup Konfigurationstypen für eine gesamte Organisation, wie unter konfiguriert AWS Organizations, oder nur für die von Ihnen ausgewählten Organisationskonten und Regionen verwenden. Sie können Quick Setup in diesen Regionen auch mit nur einem einzigen Konto verwenden.

- US East (Ohio)
- USA Ost (Nord-Virginia)
- USA West (Nordkalifornien)
- USA West (Oregon)
- Asia Pacific (Mumbai)
- Asia Pacific (Seoul)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)
- Canada (Central)
- Europa (Frankfurt)
- Europa (Stockholm)
- Europa (Irland)
- Europa (London)
- Europa (Paris)
- Südamerika (São Paulo)

In den folgenden Regionen ist nur der Konfigurationstyp Host-Verwaltung für einzelne Konten verfügbar:

- Europa (Milan)
- Asien-Pazifik (Hongkong)
- Middle East (Bahrain)
- China (Peking)

- China (Ningxia)
- AWS GovCloud (USA-Ost)
- AWS GovCloud (US-West)

Eine Liste aller von Systems Manager unterstützten Regionen finden Sie im Allgemeine Amazon Web Services-Referenz unter Service-Endpunkte von Systems Manager in der Spalte Region.

Erste Schritte mit Quick Setup

Verwenden Sie die Informationen in diesem Thema, um sich auf die Verwendung von Quick Setup vorzubereiten.

Themen

- IAM-Rollen und -Berechtigungen für das Quick Setup-Onboarding
- Manuelles Onboarding für die programmatische Arbeit mit der Quick Setup API

IAM-Rollen und -Berechtigungen für das Quick Setup-Onboarding

Quick Setup hat ein neues Konsolenerlebnis und eine neue API eingeführt. Jetzt können Sie mit dieser API über die Konsole, AWS CLI CloudFormation, und interagieren SDKs. Wenn Sie sich für das neue Erlebnis entscheiden, werden Ihre vorhandenen Konfigurationen mithilfe der neuen API neu erstellt. Je nach Anzahl der vorhandenen Konfigurationen in Ihrem Konto kann dieser Vorgang einige Minuten dauern.

Um die neue Quick Setup-Konsole verwenden zu können, müssen Sie über Berechtigungen für die folgenden Aktionen verfügen:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm-quicksetup:*",
                "cloudformation:DescribeStackSetOperation",
                "cloudformation:ListStacks",
```

```
                "cloudformation:DescribeStacks",
                "cloudformation:DescribeStackResources",
                "cloudformation:ListStackSetOperations",
                "cloudformation:ListStackInstances",
                "cloudformation:DescribeStackSet",
                "cloudformation:ListStackSets",
                "cloudformation:DescribeStackInstance",
                "cloudformation:DescribeOrganizationsAccess",
                "cloudformation:ActivateOrganizationsAccess",
                "cloudformation:GetTemplate",
                "cloudformation:ListStackSetOperationResults",
                "cloudformation:DescribeStackEvents",
                "cloudformation:UntagResource",
                "ec2:DescribeInstances",
                "ssm:DescribeAutomationExecutions",
                "ssm:GetAutomationExecution",
                "ssm:ListAssociations",
                "ssm:DescribeAssociation",
                "ssm:GetDocument",
                "ssm:ListDocuments",
                "ssm:DescribeDocument",
                "ssm:ListResourceDataSync",
                "ssm:DescribePatchBaselines",
                "ssm:GetPatchBaseline",
                "ssm:DescribeMaintenanceWindows",
                "ssm:DescribeMaintenanceWindowTasks",
                "ssm:GetOpsSummary",
                "organizations:DeregisterDelegatedAdministrator",
                "organizations:DescribeAccount",
                "organizations:DescribeOrganization",
                "organizations:ListDelegatedAdministrators",
                "organizations:ListRoots",
                "organizations:ListParents",
                "organizations:ListOrganizationalUnitsForParent",
                "organizations:DescribeOrganizationalUnit",
                "organizations:ListAWSServiceAccessForOrganization",
                "s3:GetBucketLocation",
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "resource-groups:ListGroups",
                "iam:ListRoles",
                "iam:ListRolePolicies",
                "iam:GetRole",
                "iam:CreatePolicy",
```

```
                "organizations:RegisterDelegatedAdministrator",
                "organizations:EnableAWSServiceAccess",
                "cloudformation:TagResource"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:RollbackStack",
                "cloudformation:CreateStack",
                "cloudformation:UpdateStack",
                "cloudformation:DeleteStack"
            ],
            "Resource": [
                "arn:aws:cloudformation:*:*:stack/StackSet-AWS-QuickSetup-*",
                "arn:aws:cloudformation:*:*:stack/AWS-QuickSetup-*",
                "arn:aws:cloudformation:*:*:type/resource/*",
                "arn:aws:cloudformation:*:*:stack/StackSet-SSMQuickSetup"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateStackSet",
                "cloudformation:UpdateStackSet",
                "cloudformation:DeleteStackSet",
                "cloudformation:DeleteStackInstances",
                "cloudformation:CreateStackInstances",
                "cloudformation:StopStackSetOperation"
            ],
            "Resource": [
                "arn:aws:cloudformation:*:*:stackset/AWS-QuickSetup-*",
                "arn:aws:cloudformation:*:*:stackset/SSMQuickSetup",
                "arn:aws:cloudformation:*:*:type/resource/*",
                "arn:aws:cloudformation:*:*:stackset-target/AWS-QuickSetup-*:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:AttachRolePolicy",
```

```
                "iam:DetachRolePolicy",
                "iam:GetRolePolicy",
                "iam:PutRolePolicy"
            ],
            "Resource": [
                "arn:aws:iam::*:role/AWS-QuickSetup-*",
                "arn:aws:iam::*:role/service-role/AWS-QuickSetup-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/AWS-QuickSetup-*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ssm-quicksetup.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:DeleteAssociation",
                "ssm:CreateAssociation",
                "ssm:StartAssociationsOnce"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ssm:StartAutomationExecution",
            "Resource": [
                "arn:aws:ssm:*:*:document/AWS-EnableExplorer",
                "arn:aws:ssm:*:*:automation-execution/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetOpsSummary",
                "ssm:CreateResourceDataSync",
                "ssm:UpdateResourceDataSync"
            ],
            "Resource": "arn:aws:ssm:*:*:resource-data-sync/AWS-QuickSetup-*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "accountdiscovery.ssm.amazonaws.com",
                        "ssm.amazonaws.com",
                        "ssm-quicksetup.amazonaws.com",
                        "stacksets.cloudformation.amazonaws.com"
                    ]
                }
            },
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-
service-role/stacksets.cloudformation.amazonaws.com/
AWSServiceRoleForCloudFormationStackSetsOrgAdmin"
        }
    ]
}
```

Um Benutzern schreibgeschützte Berechtigungen zu gewähren, erlauben Sie nur `ssm-quicksetup:List*`- und `ssm-quicksetup:Get*`-Operationen für die Quick Setup-API.

Quick SetupErstellt während des Onboardings die folgenden AWS Identity and Access Management (IAM-) Rollen in Ihrem Namen:

- `AWS-QuickSetup-LocalExecutionRole` – Erteilt CloudFormation Berechtigungen zur Verwendung beliebiger Vorlagen, mit Ausnahme der Patch-Richtlinienvorlage, und zur Erstellung der erforderlichen Ressourcen.
- `AWS-QuickSetup-LocalAdministrationRole`— Erteilt Berechtigungen AWS CloudFormation zur Übernahme. `AWS-QuickSetup-LocalExecutionRole`

- `AWS-QuickSetup-PatchPolicy-LocalExecutionRole`— Erteilt Berechtigungen AWS CloudFormation zur Verwendung der Patch-Richtlinienvorlage und zur Erstellung der erforderlichen Ressourcen.
- `AWS-QuickSetup-PatchPolicy-LocalAdministrationRole`— Erteilt Berechtigungen AWS CloudFormation zum Übernehmen`AWS-QuickSetup-PatchPolicy-LocalExecutionRole`.

Wenn Sie ein Verwaltungskonto einrichten — das Konto, mit dem Sie eine Organisation erstellen AWS Organizations— werden in Ihrem Quick Setup Namen auch die folgenden Rollen erstellt:

- `AWS-QuickSetup-SSM-RoleForEnablingExplorer`— Erteilt Berechtigungen dem `AWS-EnableExplorer`-Automation-Runbook. Das `AWS-EnableExplorer` Runbook konfiguriertExplorer, ein Tool in Systems Manager, um Informationen für mehrere AWS-Konten und anzuzeigen. AWS-Regionen
- `AWSServiceRoleForAmazonSSM`— Eine dienstbezogene Rolle, die Zugriff auf AWS Ressourcen gewährt, die von Systems Manager verwaltet und verwendet werden.
- `AWSServiceRoleForAmazonSSM_AccountDiscovery`— Eine serviceverknüpfte Rolle, die Systems Manager berechtigt, beim Synchronisieren von Daten anzurufen AWS-Services , um AWS-Konto Informationen zu ermitteln. Weitere Informationen finden Sie unter Verwenden von Rollen zum Sammeln von AWS-Konto-Informationen für OpsCenter und Explorer.

Quick SetupErmöglicht beim Onboarding eines Verwaltungskontos den vertrauenswürdigen Zugriff zwischen AWS Organizations und CloudFormation die Bereitstellung von Quick Setup Konfigurationen im gesamten Unternehmen. Um vertrauenswürdigen Zugriff zu aktivieren, muss Ihr Verwaltungskonto über Administratorberechtigungen verfügen. Nach dem Onboarding benötigen Sie keine Administratorberechtigungen mehr. Weitere Informationen finden Sie unter Enable trusted access with Organizations (Aktivieren des vertrauenswürdigen Zugriffs mit Organizations).

Informationen zu AWS Organizations Kontotypen finden Sie unter AWS Organizations Terminologie und Konzepten im AWS Organizations Benutzerhandbuch.

> **Note**
>
> Quick Setupverwendet CloudFormation StackSets , um Ihre Konfigurationen AWS-Konten regionsübergreifend bereitzustellen. Wenn die Anzahl der Zielkonten multipliziert mit der Anzahl der Regionen 10 000 übersteigt, kann die Konfiguration nicht bereitgestellt werden. Wir empfehlen Ihnen, Ihren Anwendungsfall zu überprüfen und Konfigurationen zu erstellen,

die weniger Ziele verwenden, um dem Wachstum Ihres Unternehmens Rechnung zu tragen. Stack-Instances werden nicht für das Verwaltungskonto Ihrer Organisation bereitgestellt. Weitere Informationen finden Sie unter Considerations when creating a stack set with service-managed permissions (Überlegungen beim Erstellen eines Stack-Sets mit service-verwalteten Berechtigungen).

Manuelles Onboarding für die programmatische Arbeit mit der Quick Setup API

Wenn Sie die Konsole zum Arbeiten mit Quick Setup verwenden, übernimmt der Service die Onboarding-Schritte für Sie. Wenn Sie planen, die Quick Setup API AWS CLI zu verwenden SDKs oder mit ihr zu arbeiten, können Sie die Konsole trotzdem verwenden, um die Onboarding-Schritte für Sie abzuschließen, sodass Sie sie nicht manuell durchführen müssen. Einige Kunden müssen die Onboarding-Schritte für Quick Setup jedoch programmgesteuert durchführen, ohne mit der Konsole zu interagieren. Wenn diese Methode zu Ihrem Anwendungsfall passt, müssen Sie die folgenden Schritte ausführen. All diese Schritte müssen von Ihrem AWS Organizations Verwaltungskonto aus abgeschlossen werden.

Um das manuelle Onboarding abzuschließen für Quick Setup

1. Aktivieren Sie den vertrauenswürdigen Zugriff für CloudFormation mit Organizations. Dadurch erhält das Verwaltungskonto die Berechtigungen, die StackSets für die Erstellung und Verwaltung Ihrer Organisation erforderlich sind. Sie können die `ActivateOrganizationsAccess` API-Aktion verwenden CloudFormation, um diesen Schritt abzuschließen. Weitere Informationen finden Sie unter ActivateOrganizationsAccess in der AWS CloudFormation -API-Referenz.
2. Ermöglichen Sie die Integration von Systems Manager mit Organizations. Auf diese Weise kann Systems Manager eine serviceverknüpfte Rolle für alle Konten Ihrer Organisation erstellen. Auf diese Weise kann Systems Manager auch Vorgänge in Ihrem Namen in Ihrer Organisation und deren Konten ausführen. Sie können die `EnableAWSServiceAccess` API-Aktion verwenden AWS Organizations, um diesen Schritt abzuschließen. Der Dienstprinzipal für Systems Manager ist`ssm.amazonaws.com`. Weitere Informationen finden Sie unter Enable AWSService Access in der AWS Organizations API-Referenz.
3. Erstellen Sie die erforderliche IAM-Rolle für Explorer. Dadurch kann Quick Setup Dashboards für Ihre Konfigurationen erstellen, sodass Sie Bereitstellungs- und Zuordnungsstatus anzeigen können. Erstellen Sie eine IAM-Rolle und fügen Sie ihr die von `AWSSystemsManagerEnableExplorerExecutionPolicy` verwaltete Richtlinie hinzu.

Ändern Sie die Vertrauensrichtlinie für die Rolle so, dass sie den folgenden Anforderungen entspricht. Ersetzen Sie jede *account ID* durch Ihre Informationen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:*:ssm:*:111122223333:automation-
execution/*"
                }
            }
        }
    ]
}
```

4. Aktualisieren Sie die Quick Setup-Serviceeinstellung für Explorer. Sie können die `UpdateServiceSettings`-API-Aktion von Quick Setup verwenden, um diesen Schritt abzuschließen. Geben Sie den ARN für die IAM-Rolle an, die Sie im vorherigen Schritt für den `ExplorerEnablingRoleArn`-Anforderungsparameter erstellt haben. Weitere Informationen finden Sie unter UpdateServiceSettings in der Quick Setup-API-Referenz.
5. Erstellen Sie die erforderlichen IAM-Rollen für CloudFormation StackSets die Verwendung. Sie müssen eine Ausführungsrolle und eine Administratorrolle erstellen.

   a. Erstellen Sie die Ausführungsrolle. Der Ausführungsrolle sollte mindestens eine der `AWSQuickSetupDeploymentRolePolicy` oder `AWSQuickSetupPatchPolicyDeploymentRolePolicy` verwalteten Richtlinien zugeordnet sein. Wenn Sie nur Konfigurationen für Patch-Richtlinien erstellen, können Sie `AWSQuickSetupPatchPolicyDeploymentRolePolicy`

verwaltete Richtlinien verwenden. Alle anderen Konfigurationen verwenden die `AWSQuickSetupDeploymentRolePolicy`-Richtlinie. Ändern Sie die Vertrauensrichtlinie für die Rolle so, dass sie den folgenden Anforderungen entspricht. Ersetzen Sie jedes *`account ID`* Feld *`administration role name`* durch Ihre Informationen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/administration
 role name"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

b. Erstellen Sie die Administratorrolle. Die Berechtigungsrichtlinie muss wie folgt übereinstimmen. Ersetzen Sie jedes *`account ID`* und *`execution role name`* durch Ihre Informationen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "sts:AssumeRole"
            ],
            "Resource": "arn:*:iam::111122223333:role/execution role
 name",
            "Effect": "Allow"
        }
    ]
}
```

Ändern Sie die Vertrauensrichtlinie für die Rolle so, dass sie den folgenden Anforderungen entspricht. Ersetzen Sie jedes *account ID* durch Ihre Informationen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudformation.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn":
 "arn:aws:cloudformation:*:111122223333:stackset/AWS-QuickSetup-*"
                }
            }
        }
    ]
}
```

Verwendung eines delegierten Administrators für Quick Setup

Nachdem Sie ein delegiertes Administratorkonto für Quick Setup registriert haben, können Benutzer mit den entsprechenden Berechtigungen in diesem Konto Konfigurationsmanager erstellen, aktualisieren, anzeigen und löschen, die auf Organisationseinheiten in Ihrer AWS Organizations-Struktur abzielen. Über dieses delegierte Administratorkonto können Sie auch Konfigurationsmanager verwalten, die mit dem Verwaltungskonto Ihrer Organisation erstellt wurden.

Über das Verwaltungskonto in Organizations kann ein Konto innerhalb Ihrer Organisation als delegierter Administrator festgelegt werden. Wenn Sie ein Konto als delegierter Administrator registrieren für Quick Setup, wird dieses Konto automatisch auch ein delegierter Administrator

für AWS CloudFormation-StackSets und Systems Manager Explorer, da diese Services zur Bereitstellung und Überwachung von Quick Setup-Konfigurationen erforderlich sind.

> **Note**
>
> Derzeit wird der Konfigurationstyp der Patch-Richtlinie vom delegierten Administrator für Quick Setup nicht unterstützt. Patch-Richtlinienkonfigurationen für eine Organisation müssen im Verwaltungskonto einer Organisation erstellt und verwaltet werden. Weitere Informationen finden Sie unter the section called “Erstellen einer Patch-Richtlinie”.

In den folgenden Themen wird beschrieben, wie Sie einen delegierten Administrator für Quick Setup registrieren und die Registrierung aufheben.

Themen

- So registrieren Sie einen delegierten Administrator für Quick Setup
- So heben Sie die Registrierung eines delegierten Administrators Quick Setup auf

So registrieren Sie einen delegierten Administrator für Quick Setup

Gehen Sie wie folgt vor, um einen delegierten Administrator für Quick Setup zu registrieren.

Registrieren Sie einen delegierten Administrator Quick Setup wie folgt

1. Melden Sie sich bei Ihrem AWS Organizations-Verwaltungskonto an.
2. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
3. Wählen Sie im Navigationsbereich Quick Setup aus.
4. Wählen Sie Einstellungen aus.
5. Überprüfen Sie im Abschnott Delegierter Administrator für Quick Setup, ob Sie die erforderlichen dienstverknüpften Rollen- und Servicezugriffsoptionen konfiguriert haben. Wählen Sie bei Bedarf die Schaltflächen Create role (Rolle erstellen) und Enable access (Zugriff gewähren) aus, um diese Optionen zu konfigurieren.
6. Geben Sie für Konto-ID die AWS-Konto-ID ein. Dieses Konto muss ein Mitgliedskonto in AWS Organizations sein.
7. Wählen Sie Register delegated administrator (Delegierten Administrator registrieren).

So heben Sie die Registrierung eines delegierten Administrators Quick Setup auf

Gehen Sie wie folgt vor, um die Registrierung eines delegierten Administrators für Quick Setup aufzuheben.

Heben Sie die Registrierung eines delegierten Administrators Quick Setup auf wie folgt

1. Loggen Sie sich in Ihr AWS Organizations Verwaltungskonto ein.
2. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
3. Wählen Sie im Navigationsbereich Quick Setup aus.
4. Wählen Sie Einstellungen aus.
5. Wählen Sie im Abschnitt Delegierter Administrator für Quick Setup aus der Dropdown-Liste Aktionen die Option Abmelden aus.
6. Wählen Sie Bestätigen aus.

Lernen Sie Quick Setup-Terminologie und Details

Quick Setup, ein Tool in AWS Systems Manager, zeigt die Ergebnisse aller von Ihnen erstellten Konfigurationsmanager in allen AWS-Regionen in der Tabelle mit den Konfigurationsmanagern auf der Quick Setup-Startseite an. Auf dieser Seite können Sie über View details (Details anzeigen) die Details jeder Konfiguration anzeigen, Konfigurationen über das Dropdown-Menü Actions (Aktionen) löschen oder über Create (Erstellen) Konfigurationen erstellen. Die Tabelle Konfigurationsmanager enthält die folgenden Informationen:

- Name – Der Name des Konfigurationsmanagers, sofern er bei der Erstellung angegeben wurde.
- Configuration type (Konfigurationstyp) – Der Konfigurationstyp, der beim Erstellen der Konfiguration ausgewählt wurde.
- Version – Die Version des aktuell bereitgestellten Konfigurationstyps.
- Organizational units (Organisationseinheiten) – Zeigt die Organisationseinheiten, für die die Konfiguration bei Auswahl von Custom (benutzerdefinierten) Zielen bereitgestellt wird. Organisationseinheiten und benutzerdefinierte Ziele stehen nur im Verwaltungskonto Ihrer Organisation zur Verfügung. Das Verwaltungskonto ist das Konto, das Sie zum Erstellen einer Organisation in AWS Organizations verwenden.
- Bereitstellungstyp – Gibt an, ob die Bereitstellung für die gesamte Organisation (`Organizational`) oder nur für Ihr Konto (`Local`) gilt.

- Regions (Regionen) – Die Regionen, in denen die Konfiguration bereitgestellt wird, wenn Sie Custom (benutzerdefinierte) Ziele oder Ziele in Ihrem Current account (aktuellem Konto) auswählen.
- Deployment status (Bereitstellungsstatus) – Der Bereitstellungsstatus zeigt an, ob AWS CloudFormation die Ziel- oder Stack-Instance erfolgreich bereitgestellt hat. Die Ziel- und Stack-Instances enthalten die Konfigurationsoptionen, die Sie bei der Erstellung der Konfiguration ausgewählt haben.
- Association status (Zuordnungsstatus) – Der Zuordnungsstatus ist der Status aller Zuordnungen, die durch die von Ihnen erstellte Konfiguration generiert wurden. Die Zuordnungen für alle Ziele müssen erfolgreich ausgeführt werden, andernfalls lautet der Status Failed (Fehlgeschlagen).

  Quick Setup erstellt und führt eine State Manager-Zuordnung für jedes Konfigurationsziel aus. State Manager ist ein Tool in AWS Systems Manager.

Um die Konfigurationen anzuzeigen, die für die Region bereitgestellt wurden, die Sie gerade durchsuchen, wählen Sie den Tab Konfigurationen.

Konfigurationsdetails

Die Seite Configuration details (Konfigurationsdetails) zeigt Informationen über die Bereitstellung der Konfiguration und die entsprechenden Zuordnungen. Auf dieser Seite können Sie Konfigurationsoptionen bearbeiten, Ziele aktualisieren oder die Konfiguration löschen. Außerdem können Sie die Details der einzelnen Konfigurationsbereitstellungen anzeigen, um weitere Informationen über die Zuordnungen zu erhalten.

Je nach Art der Konfiguration wird eines oder mehrere der folgenden Statusdiagramme angezeigt:

Configuration deployment status (Status der Konfigurationsbereitstellungen)

Zeigt die Anzahl der Bereitstellungen, die erfolgreich waren, fehlgeschlagen sind, ausgeführt werden oder noch ausstehen. Die Bereitstellungen erfolgen in den angegebenen Zielkonten und Regionen, die von der Konfiguration betroffene Knoten enthalten.

Configuration association status (Status der Konfigurationszuordnung)

Zeigt die Anzahl der State Manager-Zuordnungen, die erfolgreich waren, fehlgeschlagen sind oder noch ausstehen. Quick Setup erstellt für die ausgewählten Konfigurationsoptionen eine Zuordnung in jeder Bereitstellung.

Einrichtung des Status

Zeigt die Anzahl der vom Konfigurationstyp durchgeführten Aktionen und ihren aktuellen Status an.

Ressourcen-Compliance

Zeigt die Anzahl der Ressourcen an, die mit der angegebenen Konfigurationsrichtlinie konform sind.

Die Seite Configuration details (Konfigurationsdetails) zeigt Informationen über die Bereitstellung Ihrer Konfiguration. Weitere Details zu den einzelnen Bereitstellungen können Sie anzeigen, indem Sie eine Bereitstellung auswählen und dann auf View details (Details anzeigen) klicken. Auf der Detailseite werden die Zuordnungen angezeigt, die für die Knoten in der jeweiligen Bereitstellung bereitgestellt wurden.

Bearbeiten und Löschen Ihrer Konfiguration

Die Konfigurationsoptionen einer Konfiguration können Sie auf der Seite Configuration details (Konfigurationsdetails) bearbeiten. Dazu wählen Sie Actions (Aktionen) und dann Edit configuration options (Konfigurationsoptionen bearbeiten) aus. Wenn Sie der Konfiguration neue Optionen hinzufügen, führt Quick Setup Ihre Bereitstellungen aus und erstellt neue Zuordnungen. Wenn Sie Optionen aus einer Konfiguration entfernen, führt Quick Setup die Bereitstellungen aus und entfernt alle entsprechenden Zuordnungen.

> **Note**
>
> Die Quick Setup-Konfigurationen für Ihr Konto können Sie jederzeit ändern. Um die Konfiguration einer Organisation zu bearbeiten, muss Configuration status (Konfigurationsstatus) entweder Success (Erfolg) oder Failed (Fehlgeschlagen) lauten.

Außerdem können Sie die Ziele in Ihren Konfigurationen aktualisieren. Dazu wählen Sie Actions (Aktionen) und Add OUs (Organisationseinheiten hinzufügen), Add Regions (Regionen hinzufügen), Remove OUs (Organisationseinheiten entfernen) oder Remove Regions (Regionen entfernen) aus. Wenn Ihr Konto nicht als Verwaltungskonto konfiguriert ist oder Sie die Konfiguration nur für das aktuelle Konto erstellt haben, können Sie die Ziel-Organisationseinheiten nicht aktualisieren. Beim Entfernen einer Region oder Organisationseinheit werden deren Zuordnungen ebenfalls entfernt.

Veröffentlicht regelmäßig neue Quick Setup-Versionen von Konfigurationen. Sie können die Option Konfiguration aktualisieren auswählen, um Ihre Konfiguration auf die neueste Version zu aktualisieren.

Sie können eine Konfiguration aus Quick Setup löschen, indem Sie die Konfiguration, danach Actions (Aktionen) und dann Delete configuration (Konfiguration löschen) auswählen. Alternativ können Sie die Konfiguration auf der Seite Configuration details (Konfigurationsdetails) unter dem Dropdown-Menü Actions (Aktionen) über die Option Delete configuration (Konfiguration löschen) löschen. Quick Setup fordert Sie dann auf, Remove all OUs and Regions (Alle Organisationseinheiten und Regionen zu entfernen). Dies kann einige Zeit in Anspruch nehmen. Beim Löschen einer Konfiguration werden alle entsprechenden Zuordnungen ebenfalls gelöscht. Bei diesem zweistufigen Löschvorgang werden alle bereitgestellten Ressourcen aus allen Konten und Regionen entfernt. Anschließend wird die Konfiguration gelöscht.

Compliance von Konfigurationen

Sie können anzeigen, ob Ihre Instances den von Ihren Konfigurationen erstellten Zuordnungen entsprechen, und zwar in Explorer oder Compliance, die beide Tools in AWS Systems Manager sind. Weitere Informationen zur Compliance finden Sie unter Erfahren Sie mehr über Compliance. Weitere Informationen zum Anzeigen der Compliance in Explorer finden Sie unter AWS Systems Manager Explorer.

Verwenden der Quick Setup-API zur Verwaltung von Konfigurationen und Bereitstellungen

Sie können die von bereitgestellte API verwendenQuick Setup, um Konfigurationen und Bereitstellungen mit dem AWS CLI oder Ihrem bevorzugten SDK zu erstellen und zu verwalten. Sie können sie auch verwenden CloudFormation , um eine Configuration Manager-Ressource zu erstellen, die Konfigurationen bereitstellt. Mithilfe der API erstellen Sie Konfigurationsmanager, die Konfigurationsdefinitionen bereitstellen. Konfigurationsdefinitionen enthalten alle erforderlichen Informationen, um einen bestimmten Konfigurationstyp bereitzustellen. Weitere Informationen zur Quick Setup-API finden Sie in der Quick-Setup-API-Referenz.

Die folgenden Beispiele zeigen, wie Konfigurationsmanager mithilfe von AWS CLI und CloudFormation erstellt werden.

AWS CLI

```
aws ssm-quicksetup create-configuration-manager \
```

```
--name configuration manager name \
--description Description of your configuration manager
--configuration-definitions JSON string containing configuration defintion
```

Im Folgenden finden Sie ein Beispiel für eine JSON-Zeichenfolge, die eine Konfigurationsdefinition für die Patch-Richtlinie enthält.

```
'{"Type":"AWSQuickSetupType-
PatchPolicy","LocalDeploymentAdministrationRoleArn":"arn:aws:iam::123456789012:role/
AWS-QuickSetup-StackSet-Local-
AdministrationRole","LocalDeploymentExecutionRoleName":"AWS-
QuickSetup-StackSet-Local-ExecutionRole","Parameters":
{"ConfigurationOptionsInstallNextInterval":"true","ConfigurationOptionsInstallValue":"cron(0
 2 ? * SAT#1
 *)","ConfigurationOptionsPatchOperation":"ScanAndInstall","ConfigurationOptionsScanNextInte
 1 * * ?
 *)","HasDeletedBaseline":"false","IsPolicyAttachAllowed":"true","OutputBucketRegion":"","Ou
east-1","PatchBaselineUseDefault":"custom","PatchPolicyName":"dev-patch-
policy","RateControlConcurrency":"5","RateControlErrorThreshold":"0%","RebootOption":"Reboot
\":{\"value\":\"arn:aws:ssm:us-east-1:123456789012:patchbaseline/
pb-0cb0c4966f86b059b\",\"label\":\"AWS-AlmaLinuxDefaultPatchBaseline\",\"description
\":\"Default Patch Baseline for Alma Linux Provided by AWS.\",\"disabled\":false},
\"AMAZON_LINUX_2\":{\"value\":\"arn:aws:ssm:us-east-1:123456789012:patchbaseline/
pb-0be8c61cde3be63f3\",\"label\":\"AWS-AmazonLinux2DefaultPatchBaseline\",
\"description\":\"Baseline containing all Security and Bugfix updates approved
 for Amazon Linux 2 instances\",\"disabled\":false},\"AMAZON_LINUX_2023\":{\"value
\":\"arn:aws:ssm:us-east-1:123456789012:patchbaseline/pb-05c9c9bf778d4c4d0\",
\"label\":\"AWS-AmazonLinux2023DefaultPatchBaseline\",\"description\":\"Default
 Patch Baseline for Amazon Linux 2023 Provided by AWS.\",\"disabled\":false},
\"DEBIAN\":{\"value\":\"arn:aws:ssm:us-east-1:123456789012:patchbaseline/
pb-09a5f8eb62bde80b1\",\"label\":\"AWS-DebianDefaultPatchBaseline\",\"description
\":\"Default Patch Baseline for Debian Provided by AWS.\",\"disabled\":false},
\"MACOS\":{\"value\":\"arn:aws:ssm:us-east-1:123456789012:patchbaseline/
pb-0ee4f94581368c0d4\",\"label\":\"AWS-MacOSDefaultPatchBaseline\",\"description
\":\"Default Patch Baseline for MacOS Provided by AWS.\",\"disabled\":false},
\"ORACLE_LINUX\":{\"value\":\"arn:aws:ssm:us-east-1:123456789012:patchbaseline/
pb-06bff38e95fe85c02\",\"label\":\"AWS-OracleLinuxDefaultPatchBaseline\",
\"description\":\"Default Patch Baseline for Oracle Linux Server Provided
 by AWS.\",\"disabled\":false},\"REDHAT_ENTERPRISE_LINUX\":{\"value\":
\"arn:aws:ssm:us-east-1:123456789012:patchbaseline/pb-0cbb3a633de00f07c\",
\"label\":\"AWS-RedHatDefaultPatchBaseline\",\"description\":\"Default Patch
 Baseline for Redhat Enterprise Linux Provided by AWS.\",\"disabled\":false},
\"ROCKY_LINUX\":{\"value\":\"arn:aws:ssm:us-east-1:123456789012:patchbaseline/
```

```
pb-03ec98bc512aa3ac0\",\"label\":\"AWS-RockyLinuxDefaultPatchBaseline\",
\"description\":\"Default Patch Baseline for Rocky Linux Provided by AWS.\",
\"disabled\":false},\"UBUNTU\":{\"value\":\"pb-06e3563bd35503f2b\",\"label\":
\"custom-UbuntuServer-Blog-Baseline\",\"description\":\"Default Patch Baseline
 for Ubuntu Provided by AWS.\",\"disabled\":false},\"WINDOWS\":{\"value\":
\"pb-016889927b2bb8542\",\"label\":\"custom-WindowsServer-Blog-Baseline\",
\"disabled\":false}}","TargetInstances":"","TargetOrganizationalUnits":"ou-9utf-
example","TargetRegions":"us-east-1,us-
east-2","TargetTagKey":"Patch","TargetTagValue":"true","TargetType":"Tags"}}' \
```

CloudFormation

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
SSMQuickSetupTestConfigurationManager:
Type: "AWS::SSMQuickSetup::ConfigurationManager"
Properties:
    Name: "MyQuickSetup"
    Description: "Test configuration manager"
    ConfigurationDefinitions:
    - Type: "AWSQuickSetupType-CFGRecording"
      Parameters:
        TargetAccounts:
            Ref: AWS::AccountId
        TargetRegions:
            Ref: AWS::Region
        LocalDeploymentAdministrationRoleArn: !Sub "arn:aws:iam::
${AWS::AccountId}:role/AWS-QuickSetup-StackSet-ContractTest-AdministrationRole"
        LocalDeploymentExecutionRoleName: "AWS-QuickSetup-StackSet-ContractTest-
ExecutionRole"
    Tags:
        foo1: "bar1"
```

Unterstützte Quick Setup-Konfigurationstypen

Unterstützte Konfigurationstypen

Quick Setup führt Sie durch die Konfiguration betrieblicher bewährter Systems Manager und AWS-Services stellt diese Konfigurationen automatisch bereit. Das Quick Setup-Dashboard zeigt eine Echtzeitansicht Ihres Konfigurationsbereitstellungsstatus an.

Sie können Quick Setup in einem einzelnen AWS-Konto oder über mehrere AWS-Konten und Regionen hinweg durch die Integration mit AWS Organizations nutzen. Durch Verwenden von Quick Setup über mehrere Konten hinweg können Sie sicherstellen, dass Ihre Organisation konsistente Konfigurationen aufrechterhält.

Quick Setup bietet Unterstützung für die folgenden Konfigurationstypen.

- Amazon-EC2-Host-Verwaltung mit Quick Setup einrichten
- Die Standardkonfiguration für die Host-Verwaltung für eine Organisation mit Quick Setup einrichten
- Erstellen Sie einen AWS Config-Konfigurationsrekorder mit Quick Setup
- Stellen Sie das AWS Config Conformance Pack bereit mit Quick Setup
- Das Patchen für Instances in einer Organisation mithilfe einer Patch-Richtlinie für Quick Setup konfigurieren
- Einrichtung der Organisation durch Change Manager
- DevOps-Guru einrichten mit Quick Setup
- Bereitstellen von Distributor-Paketen mit Quick Setup
- Automatisches Stoppen und Starten von EC2-Instances nach einem Zeitplan mit Quick Setup
- Einrichtung der Organisation durch OpsCenter
- Konfigurieren von AWS Resource Explorer mithilfe von Quick Setup

Amazon-EC2-Host-Verwaltung mit Quick Setup einrichten

Verwenden Sie Quick Setup, ein Tool von AWS Systems Manager, um die erforderlichen Sicherheitsrollen und häufig verwendeten Systems Manager-Tools in Ihren Amazon Elastic Compute Cloud (Amazon EC2)-Instances schnell zu konfigurieren. Sie können Quick Setup in einem einzelnen Konto oder über mehrere Konten und AWS-Regionen hinweg durch die Integration mit AWS Organizations nutzen. Diese Tools helfen Ihnen, den Zustand Ihrer Instances zu verwalten und zu überwachen und gleichzeitig das Minimum an erforderlichen Berechtigungen für den Einstieg bereitzustellen.

Wenn Sie mit den Services und Features von Systems Manager nicht vertraut sind, empfehlen wir, das AWS Systems Manager-Benutzerhandbuch zu lesen, bevor Sie eine Konfiguration mit Quick Setup erstellen. Weitere Informationen zur Systems Manager finden Sie unter Was ist AWS Systems Manager?.

Important

Quick Setup ist möglicherweise nicht das richtige Tool für das EC2-Management, wenn einer der folgenden Punkte auf Sie zutrifft:

- Sie versuchen zum ersten Mal, eine EC2-Instance zu erstellen, um AWS-Funktionen auszuprobieren.
- Sie haben noch nicht viel Erfahrung mit EC2-Instance-Management.

Stattdessen empfehlen wir Ihnen, die folgenden Inhalte zu untersuchen:

- Erste Schritte mit Amazon EC2
- Eine Instance mit dem neuen Launch Instance Wizard im Benutzerhandbuch zu Amazon EC2 starten
- Tutorial: Erste Schritte mit Amazon-EC2-Linux-Instances im Amazon-EC2-Benutzerhandbuch

Wenn Sie bereits mit dem EC2-Instance-Management vertraut sind und die Konfiguration und Verwaltung für mehrere EC2-Instances optimieren möchten, verwenden Sie Quick Setup. Unabhängig davon, ob Ihr Unternehmen über Dutzende, Tausende oder Millionen von EC2-Instances verfügt, verwenden Sie das folgende Quick Setup-Verfahren, um mehrere Optionen gleichzeitig für sie zu konfigurieren.

Note

Mit diesem Konfigurationstyp können Sie mehrere Optionen festlegen, entweder für eine komplette in AWS Organizations definierte Organisation, nur für einige Organisationskonten und Regionen oder für ein einzelnes Konto. Eine dieser Optionen besteht darin, alle zwei Wochen nach Updates für SSM Agent zu suchen und diese anzuwenden. Organisationsadministratoren können auch alle EC2-Instances in ihrer Organisation alle zwei Wochen mit Agenten-Updates aktualisieren, indem sie die Standardkonfiguration für die Host-Verwaltung verwenden. Weitere Informationen finden Sie unter Die Standardkonfiguration für die Host-Verwaltung für eine Organisation mit Quick Setup einrichten.

Konfiguration der Host-Management-Optionen für EC2-Instances

Führen Sie zum Einrichten der Hostverwaltung die folgenden Aufgaben in der AWS Systems Manager Quick Setup-Konsole aus.

So öffnen Sie die Host-Management-Konfigurationsseite

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Karte Host-Verwaltung die Option Erstellen aus.

> **Tip**
>
> Wenn Sie bereits eine oder mehrere Konfigurationen in Ihrem Konto haben, wählen Sie zunächst die Registerkarte Bibliothek oder die Schaltfläche Erstellen im Abschnitt Konfigurationen, um die Karten anzuzeigen.

So konfigurieren Sie die Hostverwaltungsoptionen von Systems Manager

- Um die Systems-Manager-Funktionalität zu konfigurieren, wählen Sie im Abschnitt Konfigurationsoptionen die Optionen in der Systems-Manager-Gruppe aus, die Sie für Ihre Konfiguration aktivieren möchten:

  Aktualisieren Sie den Systems Manager (SSM)-Agenten alle zwei Wochen

  Ermöglicht Systems Manager, alle zwei Wochen nach einer neuen Version des Agenten zu suchen. Wenn es eine neue Version gibt, aktualisiert Systems Manager den Agenten auf Ihrem verwalteten Knoten automatisch auf die neueste freigegebene Version. Quick Setup installiert den Agenten nicht auf Instances, auf denen er nicht bereits vorhanden ist. Informationen darüber, welche AMIs SSM Agent vorinstalliert haben, finden Sie unter Finden Sie AMIs mit dem vorinstallierten SSM Agent.

  Wir empfehlen Ihnen, diese Option zu wählen, um sicherzustellen, dass Ihre Knoten immer die aktuellste Version von SSM Agent ausführen. Weitere Informationen zu SSM Agent,

einschließlich Informationen zur manuellen Installation des Agenten, finden Sie unter Arbeiten mit SSM Agent.

Erfassen von Beständen aus Ihren Instances alle 30 Minuten

Quick Setup konfiguriert die Sammlung der folgenden Metadatentypen:

- AWS components – EC2-Treiber, Agenten, Versionen und mehr.
- Anwendungen – Anwendungsnamen, Publisher, Versionen und mehr.
- Knoten-Details – Systemname, Name des Betriebssystems (OS), OS-Version, letzter Boot-Vorgang, DNS, Domain, Arbeitsgruppe, OS-Architektur und mehr.
- Netzwerkkonfiguration – IP-Adresse, MAC-Adresse, DNS-Gateway, Subnetzmaske und mehr.
- Services – Name, Anzeigename, Status, abhängige Services, Servicetyp, Starttyp und mehr (nur Windows Server-Knoten).
- Windows-Rollen – Name, Anzeigename, Pfad, Featuretyp, Installationsstatus und mehr (nur Windows Server-Knoten).
- Windows-Updates – Hotfix-ID, installiert von, Installationsdatum und mehr (nur Windows Server-Knoten).

Weitere Informationen zu Inventory, ein Tool in AWS Systems Manager, finden Sie unter AWS Systems Manager-Bestand.

> **Note**
>
> Die Bestandserfassungs-Option kann bis zu 10 Minuten dauern, auch wenn Sie nur wenige Knoten ausgewählt haben.

Tägliches Scannen von Instances nach fehlenden Patches

Ermöglicht Patch Manager, einem Tool in Systems Manager, die Knoten täglich zu scannen und einen Bericht auf der Seite Compliance zu erstellen. Der Bericht zeigt, wie viele Knoten entsprechend der Standard-Patch-Baseline patchkompatibel sind. Der Bericht enthält eine Liste der einzelnen Knoten und deren Compliance-Status.

Informationen zu Patching-Vorgängen und Patch-Baselines finden Sie unter AWS Systems Manager Patch Manager.

Informationen zur Patch-Compliance finden Sie auf der Seite Systems-Manager-Compliance.

Informationen zum Patchen verwalteter Knoten in mehreren Konten und Regionen in einer Konfiguration finden Sie unter Patch-Richtlinienkonfigurationen in Quick Setup und Das Patchen für Instances in einer Organisation mithilfe einer Patch-Richtlinie für Quick Setup konfigurieren.

> **Important**
>
> Systems Manager unterstützt mehrere Methoden zum Scannen verwalteter Knoten auf Patch-Compliance. Wenn Sie mehr als eine dieser Methoden gleichzeitig implementieren, sind die angezeigten Patch-Compliance-Informationen immer das Ergebnis des letzten Scans. Ergebnisse früherer Scans werden überschrieben. Wenn die Scan-Methoden unterschiedliche Patch-Baselines mit unterschiedlichen Genehmigungsregeln verwenden, können sich die Informationen zur Patch-Compliance unerwartet ändern. Weitere Informationen finden Sie unter Identifizieren der Ausführung, die Patch-Compliance-Daten erstellt hat.

So konfigurieren Sie die Hostverwaltungsoptionen von Amazon CloudWatch

- Um die CloudWatch-Funktionalität zu konfigurieren, wählen Sie im Abschnitt Konfigurationsoptionen die Optionen in der Amazon-CloudWatch-Gruppe aus, die Sie für Ihre Konfiguration aktivieren möchten:

  CloudWatch-Agenten installieren und konfigurieren

  – Installiert die Basiskonfiguration des einheitlichen CloudWatch-Agent auf Ihren Amazon-EC2-Instances. Der Agent sammelt Metriken und Protokolldateien von Ihren Instances für Amazon CloudWatch. Diese Informationen werden zusammengefasst, damit Sie den Zustand Ihrer Instances schnell bestimmen können. Weitere Informationen zur Basiskonfiguration des CloudWatch-Agenten finden Sie unter CloudWatch agent predefined metric sets (Vordefinierte Metriksätze des CloudWatch-Agenten). Es können zusätzliche Kosten anfallen. Weitere Informationen hierzu finden Sie unter Amazon CloudWatch – Preise.

Aktualisieren Sie den CloudWatch-Agenten einmal alle 30 Tage

– Hiermit sucht Systems Manager automatisch alle 30 Tage nach einer neuen Version des CloudWatch-Agenten. Wenn es eine neue Version gibt, aktualisiert Systems Manager den Agent automatisch auf Ihrer Instance. Wir empfehlen Ihnen, diese Option zu wählen, um sicherzustellen, dass Ihre Instances immer die aktuellste Version des CloudWatch-Agenten ausführen.

So konfigurieren Sie die Hostverwaltungsoptionen für Amazon-EC2-Launch-Agent

- Um die Funktion von Amazon-EC2-Launch-Agent zu konfigurieren, wählen Sie im Abschnitt Konfigurationsoptionen die Optionen in der Gruppe von Amazon EC2 Launch Agent aus, die Sie für Ihre Konfiguration aktivieren möchten:

Den EC2-Launch-Agent einmal alle 30 Tage aktualisieren

Ermöglicht Systems Manager, alle 30 Tage zu prüfen, ob auf Ihrer Instance eine neue Version des Startagenten installiert ist. Wenn eine neue Version verfügbar ist, aktualisiert Systems Manager den Agenten auf Ihrer Instance. Wir empfehlen Ihnen, diese Option zu wählen, um sicherzustellen, dass Ihre Instances immer die aktuellste Version des jeweiligen Startagenten ausführen. Für Amazon-EC2-Windows-Instances unterstützt diese Option EC2Launch, EC2Launch v2 und EC2Config. Für Amazon-EC2-Linux-Instances unterstützt diese Option `cloud-init`. Für Amazon-EC2-Mac-Instances unterstützt diese Option `ec2-macos-init`. Quick Setup unterstützt nicht die Aktualisierung von Startagenten, die auf Betriebssystemen, die nicht vom Startagenten unterstützt werden, oder auf AL2023 installiert sind.

Weitere Informationen zu diesen Initialisierungsagenten finden Sie in den folgenden Themen:

- Konfigurieren einer Windows-Instance mithilfe von EC2Launch v2
- Konfigurieren einer Windows-Instance mithilfe von EC2Launch
- Konfigurieren einer Windows-Instance mithilfe des EC2Config-Service
- Cloud-Init-Dokumentation
- ec2-macos-init

Um die EC2-Instances auszuwählen, die durch die Host-Management-Konfiguration aktualisiert werden sollen

- Wählen Sie im Abschnitt Ziele die Methode aus, um die Konten und Regionen zu bestimmen, in denen die Konfiguration bereitgestellt werden soll:

  > **Note**
  >
  > Sie können nicht mehrere Quick Setup Host Management-Konfigurationen erstellen, die auf dieselbe AWS-Region abzielen.

  Entire organization

  Ihre Konfiguration wird in allen Organisationseinheiten (OUs) und AWS-Regionen in Ihrer Organisation bereitgestellt.

  > **Note**
  >
  > Die Option Entire organization (Gesamte Organisation) ist nur verfügbar, wenn Sie die Hostverwaltung über das Verwaltungskonto Ihrer Organisation konfigurieren.

  Custom

  1. Wählen Sie im Abschnitt Ziel-Organisationseinheiten die Organisationseinheiten aus, in denen Sie diese Host-Management-Konfiguration bereitstellen möchten.
  2. Wählen Sie im Abschnitt Zielregionen die Regionen aus, in denen Sie diese Host-Management-Konfiguration bereitstellen möchten.

  Current account

  Wählen Sie eine der Regionsoptionen und folgen Sie den Anweisungen für diese Option.

  Aktuelle Region

  Wählen Sie aus, wie Sie nur auf Instances in der aktuellen Region abzielen möchten:

- Alle Instances – Die Host-Verwaltungskonfiguration zielt automatisch auf alle EC2 in der aktuellen Region ab.
- Tag – Wählen Sie Hinzufügen und geben Sie den Schlüssel und den optionalen Wert ein, der den Instances hinzugefügt wird, auf die abgezielz werden soll.
- Ressourcengruppe – Wählen Sie unter Ressourcengruppe eine bestehende Ressourcengruppe aus, die die EC2-Instances enthält, auf die abgezielt werden soll.
- Manuell – Aktivieren Sie im Abschnitt Instances das Kontrollkästchen jeder EC2-Instance, auf die Sie abzielen möchten.

Regionen auswählen

Wählen Sie aus, wie Instances in der von Ihnen angegebenen Region ins Visier genommen werden sollen, indem Sie eine der folgenden Optionen wählen:

- Alle Instances – Auf alle Instances in den von Ihnen angegebenen Regionen wird abgezielt.
- Tag – Wählen Sie Hinzufügen und geben Sie den Schlüssel und den optionalen Wert ein, der den Instances hinzugefügt wurde, auf die abgezielt wird.

Wählen Sie im Abschnitt Zielregionen die Regionen aus, in denen Sie diese Host-Management-Konfiguration bereitstellen möchten.

Um eine Instance-Profiloption anzugeben

- Nur die gesamte Organisation und benutzerdefinierte Ziele.

  Wählen Sie im Abschnitt Instance-Profiloptionen aus, ob Sie die erforderlichen IAM-Richtlinien zu den vorhandenen Instance-Profilen hinzufügen möchten, die Ihren Instances zugeordnet sind, oder ob Sie Quick Setup die Erstellung der IAM-Richtlinien und Instance-Profile mit den für die von Ihnen gewählte Konfiguration erforderlichen Berechtigungen zulassen möchten.

Nachdem Sie alle Ihre Konfigurationsoptionen angegeben haben, wählen Sie Erstellen.

Die Standardkonfiguration für die Host-Verwaltung für eine Organisation mit Quick Setup einrichten

Mit Quick Setup, einem Tool in AWS Systems Manager, können Sie die Standardkonfiguration der Host-Verwaltung für alle Konten und Regionen aktivieren, die Ihrer Organisation in AWS Organizations hinzugefügt wurden. Dadurch wird sichergestellt, dass SSM Agent über alle EC2-

Instances (Amazon Elastic Compute Cloud) in der Organisation auf dem Laufenden gehalten wird und dass diese sich mit dem Systems Manager verbinden können.

Bevor Sie beginnen

Stellen Sie sicher, dass die folgenden Voraussetzungen erfüllt sind, bevor Sie diese Einstellung aktivieren.

- Die neueste Version von SSM Agent ist bereits auf allen EC2-Instances installiert, die in Ihrer Organisation verwaltet werden sollen.
- Ihre zu verwaltenden Instances verwenden Instance Metadata Service Version 2 (IMDSv2).
- Sie sind mit einer AWS Identity and Access Management (IAM)-Identität (Benutzer, Rolle oder Gruppe) mit Administratorrechten beim Verwaltungskonto für Ihre Organisation angemeldet, wie in AWS Organizations angegeben.

Verwenden der standardmäßigen EC2-Instance-Verwaltungsrolle

Die Standardkonfiguration für die Host-Verwaltung verwendet die `default-ec2-instance-management-role`-Diensteinstellung für Systems Manager. Dies ist eine Rolle mit Berechtigungen, die Sie allen Konten in Ihrer Organisation zur Verfügung stellen möchten, um die Kommunikation zwischen SSM Agent auf der Instance und dem Systems-Manager-Dienst in der Cloud zu ermöglichen.

Wenn Sie diese Rolle bereits mit dem CLI-Befehl update-service-setting festgelegt haben, verwendet die Standardkonfiguration für die Host-Verwaltung diese Rolle. Wenn Sie diese Rolle noch nicht festgelegt haben, erstellt Quick Setup sie und wendet sie für Sie an.

Überprüfen Sie mit dem get-service-setting-Befehl, ob diese Rolle bereits für Ihre Organisation festgelegt wurde.

Aktivieren Sie automatische 14-tägige Updates von SSM Agent.

Gehen Sie wie folgt vor, um die Standardkonfiguration für die Host-Verwaltung für Ihre gesamte AWS Organizations-Organisation zu aktivieren.

So aktivieren Sie automatische 14-tägige Updates von SSM Agent

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.

3. Wählen Sie auf der Karte Standardkonfiguration für die Host-Verwaltung die Option Erstellen aus.

> **Tip**
>
> Wenn Sie bereits eine oder mehrere Konfigurationen in Ihrem Konto haben, wählen Sie zunächst die Registerkarte Bibliothek oder die Schaltfläche Erstellen im Abschnitt Konfigurationen, um die Karten anzuzeigen.

4. Wählen Sie im Abschnitt Konfigurationsoptionen die Option Automatische Updates von SSM Agent alle zwei Wochen aktivieren aus.
5. Wählen Sie Create (Erstellen) aus.

Erstellen Sie einen AWS Config-Konfigurationsrekorder mit Quick Setup

Mit Quick Setup, einem Tool in AWS Systems Manager, können Sie schnell einen Konfigurations-Recorder erstellen, der von AWS Config unterstützt wird. Verwenden Sie den Konfigurations-Recorder zur Erkennung von Änderungen an Ihren Ressourcenkonfigurationen und zur Erfassung der Änderungen als Konfigurationselemente. Wenn Sie mit AWS Config nicht vertraut sind, empfehlen wir Ihnen, mehr über den Service zu erfahren, indem Sie den Inhalt des AWS Config Developer Guide lesen, bevor Sie eine Konfiguration mit Quick Setup erstellen. Weitere Informationen über AWS Config finden Sie unter Was ist AWS Config? im AWS Config-Entwicklerhandbuch.

Standardmäßig zeichnet der Configuration Recorder alle unterstützten Ressourcen in der AWS-Region auf, in der AWS Config ausgeführt wird. Sie können die Konfiguration so anpassen, dass nur die von Ihnen angegebenen Ressourcentypen aufgezeichnet werden. Weitere Informationen finden Sie unter Selecting which resources AWS Config records im AWS Config Developer Guide.

Beim Aufzeichnen von Konfigurationen mit AWS Config fallen Service-Nutzungsgebühren an. Preisinformationen finden Sie unter AWS Config Preise.

> **Note**
>
> Wenn Sie bereits einen Konfigurationsrekorder erstellt haben, stoppt Quick Setup die Aufzeichnung nicht und nimmt auch keine Änderungen an den Ressourcentypen vor, die Sie bereits aufzeichnen. Wenn Sie zusätzliche Ressourcentypen mit Quick Setup aufzeichnen möchten, hängt der Service sie an Ihre vorhandenen Rekordergruppen an. Durch das

Löschen des Konfigurationstyps Quick Setup Config recording (Config-Aufzeichnung) wird der Konfigurations-Recorder nicht gestoppt. Änderungen werden weiterhin aufgezeichnet, und die Servicenutzungsgebühren fallen an, bis Sie den Konfigurations-Recorder beenden. Weitere Informationen zur Verwaltung des Konfigurations-Recorders finden Sie unter Managing the Configuration Recorder (Verwalten des Konfigurations-Recorders) im AWS Config Developer Guide.

Führen Sie zur AWS Config-Aufzeichnung die folgenden Aufgaben in der AWS Systems Manager-Konsole aus.

Einrichten der AWS Config-Aufzeichnung mit Quick Setup

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Karte Konfigurationsaufnahme die Option Erstellen aus.

   > **Tip**
   >
   > Wenn Sie bereits eine oder mehrere Konfigurationen in Ihrem Konto haben, wählen Sie zunächst die Registerkarte Bibliothek oder die Schaltfläche Erstellen im Abschnitt Konfigurationen, um die Karten anzuzeigen.

4. Führen Sie im Abschnitt Cluster-Optionen Folgendes aus:

   a. Geben Sie unter Wählen Sie die aufzuzeichnenden AWS-Ressourcentypen an, ob alle unterstützten Ressourcen oder nur die von Ihnen ausgewählten Ressourcentypen aufgezeichnet werden sollen.

   b. Geben Sie unter Übermittlungseinstellungen an, ob ein neuer Amazon Simple Storage Service (Amazon S3)-Bucket erstellt werden soll, oder wählen Sie einen vorhandenen Bucket aus, an den Konfigurations-Snapshots gesendet werden sollen.

   c. Wählen Sie unter Benachrichtigungsoptionen die gewünschte Benachrichtigungsoption aus. AWS Config verwendet Amazon Simple Notification Service (Amazon SNS), um Sie über wichtige AWS Config-Ereignissen im Zusammenhang mit Ihren Ressourcen zu benachrichtigen. Wenn Sie die Option Use existing SNS topics (Vorhandene SNS-Themen verwenden) wählen, müssen Sie die AWS-Konto-ID und den Namen des vorhandenen

Amazon SNS-Themas in dem Konto, das Sie verwenden möchten, angeben. Wenn Sie mehrere AWS-Regionen anvisieren, müssen die Themennamen in jeder Region identisch sein.

5. Wählen Sie im Abschnitt Schedule aus, wie häufig Quick Setup Änderungen an Ressourcen, die von Ihrer Konfiguration abweichen, beheben soll. Die Standard-Option wird einmal ausgeführt. Wenn Sie nicht möchten, dass Quick Setup Änderungen an Ressourcen, die von Ihrer Konfiguration abweichen, wiederherstellt, wählen Sie unter Custom (Benutzerdefiniert) die Option Disable remediation (Wiederherstellung deaktivieren).
6. Wählen Sie im Abschnitt Ziele eine der folgenden Optionen aus, um die Konten und Regionen für die Aufzeichnung zu identifizieren.

> **Note**
>
> Wenn Sie in einem einzelnen Konto arbeiten, sind Optionen für die Arbeit mit Organizations und Organisationseinheiten (OUs) nicht verfügbar. Sie können auswählen, ob Sie diese Konfiguration auf alle AWS-Regionen in Ihrem Konto oder nur auf die von Ihnen ausgewählten Regionen anwenden möchten.

   - Gesamte Organisation – Alle Konten und Regionen in Ihrer Organisation.
   - Benutzerdefiniert – Nur die von Ihnen angegebenen Organisationseinheiten und Regionen.
     - Wählen Sie im Abschnitt Ziel-OUs die OUs aus, in denen Sie die Aufzeichnung zulassen möchten.
     - Wählen Sie im Abschnitt Zielregionen die Regionen aus, in denen Sie die Aufzeichnung zulassen möchten.
   - Aktuelles Konto – Nur die Regionen, die Sie in dem Konto angeben, bei dem Sie derzeit angemeldet sind, werden als Ziel ausgewählt. Wählen Sie eine der folgenden Optionen aus:
     - Aktuelle Region – Nur verwaltete Knoten in der Region, die in der Konsole ausgewählt wurde, werden als Ziel ausgewählt.
     - Regionen auswählen – Wählen Sie die einzelnen Regionen aus, auf die die Aufnahmekonfiguration angewendet werden soll.
7. Wählen Sie Erstellen aus.

Stellen Sie das AWS Config Conformance Pack bereit mit Quick Setup

Ein Konformitätspaket ist eine Sammlung von AWS Config-Regeln und Behebungsaktionen. Mit Quick Setup können Sie ein Konformitätspaket als einzelne Entität in einem Konto und einer AWS-Region oder organisationsweit in einem AWS Organizations bereitstellen. Auf diese Weise können Sie die Compliance Ihrer AWS-Ressourcen in großem Umfang verwalten, von der Richtliniendefinition bis hin zur Prüfung und aggregierten Berichten, unter Verwendung eines gemeinsamen Framework- und Paketmodells.

Um Konformitätspakete bereitzustellen, führen Sie in der AWS Systems Manager Quick Setup-Konsole die folgenden Aufgaben aus.

> **Note**
>
> Sie müssen die AWS Config-Aufzeichnung vor dem Bereitstellen dieser Konfiguration aktivieren. Weitere Informationen finden Sie unter Konformitätspakete im AWS Config Developer Guide.

Bereitstellen von Konformitätspaketen mit Quick Setup

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Karte Konformitätspakete die Option Erstellen.

   > **Tip**
   >
   > Wenn Sie bereits eine oder mehrere Konfigurationen in Ihrem Konto haben, wählen Sie zunächst die Registerkarte Bibliothek oder die Schaltfläche Erstellen im Abschnitt Konfigurationen, um die Karten anzuzeigen.

4. Wählen Sie im Abschnitt Konformitätspakete auswählen die Konformitätspakete aus, die Sie einsetzen möchten.
5. Wählen Sie im Abschnitt Schedule aus, wie häufig Quick Setup Änderungen an Ressourcen, die von Ihrer Konfiguration abweichen, beheben soll. Die Standard-Option wird einmal ausgeführt. Wenn Sie nicht möchten, dass Quick Setup Änderungen an Ressourcen, die von

Ihrer Konfiguration abweichen, wiederherstellt, wählen Sie unter Benutzerdefiniert die Option Deaktiviert.

6. Wählen Sie im Abschnitt Targets (Ziele) aus, ob Konformitätspakete für Ihre gesamte Organisation, für einige AWS-Regionen oder für das Konto, bei dem Sie gerade angemeldet sind, bereitgestellt werden sollen.

   Fahren Sie mit Schritt 8 fort, wenn Sie Ganze Organisation wählen.

   Fahren Sie mit Schritt 7 fort, wenn Sie Benutzerdefiniert wählen.

7. Aktivieren Sie im Abschnitt Target Regions (Zielregionen) die Kontrollkästchen der Regionen, für die Sie Konformitätspakete bereitstellen möchten.
8. Wählen Sie Erstellen aus.

Das Patchen für Instances in einer Organisation mithilfe einer Patch-Richtlinie für Quick Setup konfigurieren

Mit Quick Setup einem Tool in können Sie Patch-Richtlinien erstellen AWS Systems Manager, die auf folgenden Komponenten basieren: Patch Manager Eine Patch-Richtlinie definiert den Zeitplan und die Baseline, die beim automatischen Patchen Ihrer Amazon Elastic Compute Cloud (Amazon EC2) -Instances und anderer verwalteter Knoten verwendet werden sollen. Mit einer einzelnen Patch-Richtlinienkonfiguration können Sie Patches für alle Konten in mehreren AWS-Regionen in Ihrer Organisation, nur für die von Ihnen ausgewählten Konten und Regionen oder für ein einzelnes Konto-Region-Paar definieren. Weitere Informationen zu Patch-Richtlinien finden Sie unter Patch-Richtlinienkonfigurationen in Quick Setup.

Voraussetzung

Um eine Patch-Richtlinie für einen Knoten mit Quick Setup zu definieren, muss es sich bei dem Knoten um einen verwalteten Knoten handeln. Weitere Informationen zum Verwalten Ihrer Knoten finden Sie unter Einrichten der einheitlichen Systems-Manager-Konsole für eine Organisation.

> **Important**
>
> Methoden zum Scannen der Patch-Compliance – Systems Manager unterstützt mehrere Methoden zum Scannen verwalteter Knoten auf Patch-Compliance. Wenn Sie mehr als eine dieser Methoden gleichzeitig implementieren, sind die angezeigten Patch-Compliance-Informationen immer das Ergebnis des letzten Scans. Ergebnisse früherer Scans werden überschrieben. Wenn die Scan-Methoden unterschiedliche Patch-Baselines

mit unterschiedlichen Genehmigungsregeln verwenden, können sich die Informationen zur Patch-Compliance unerwartet ändern. Weitere Informationen finden Sie unter Identifizieren der Ausführung, die Patch-Compliance-Daten erstellt hat.

Konformitätsstatus der Zuordnungen und Patch-Richtlinien – Der Patching-Status für einen verwalteten Knoten, für den eine Quick Setup-Patch-Richtlinie gilt, entspricht dem Status der State Manager-Zuordnungsausführung für diesen Knoten. Wenn der Status der Zuordnungsausführung `Compliant` lautet, wird der Patching-Status für den verwalteten Knoten ebenfalls als `Compliant` markiert. Wenn der Status der Zuordnungsausführung `Non-Compliant` lautet, wird der Patching-Status für den verwalteten Knoten ebenfalls als `Non-Compliant` markiert.

Unterstützte Regionen für Patch-Richtlinienkonfigurationen

Patch-Richtlinien-Konfigurationen in Quick Setup werden derzeit in den folgenden Regionen unterstützt:

- USA Ost (Ohio): (us-east-2)
- USA Ost (Nord-Virginia): (us-east-1)
- USA West (Nordkalifornien) (us-west-1)
- USA West (Oregon): (us-west-2)
- Asien-Pazifik (Mumbai): (ap-south-1)
- Asien-Pazifik (Seoul): (ap-northeast-2)
- Asien-Pazifik (Singapur): (ap-southeast-1)
- Asien-Pazifik (Sydney): (ap-southeast-2)
- Asien-Pazifik (Tokyo) (ap-northeast-1)
- Kanada (Zentral): (ca-central-1)
- Europa (Frankfurt) (eu-central-1)
- Europa (Irland) (eu-west-1)
- Europa (London) (eu-west-2)
- Europa (Paris) (eu-west-3)
- Europa (Stockholm) (eu-north-1)
- Südamerika (São Paulo) (sa-east-1)

Berechtigungen für den S3-Bucket mit der Patch-Richtlinie

Wenn Sie eine Patch-Richtlinie erstellen, erstellt Quick Setup einen Amazon-S3-Bucket, der eine Datei mit dem Namen `baseline_overrides.json` enthält. In dieser Datei werden Informationen zu den Patch-Baselines gespeichert, die Sie für Ihre Patch-Richtlinie angegeben haben.

Der S3-Bucket-Name hat das folgende Format `aws-quicksetup-patchpolicy-`*`account-id`*`-`*`quick-setup-configuration-id`*.

Beispiel: `aws-quicksetup-patchpolicy-123456789012-abcde`

Wenn Sie eine Patch-Richtlinie für eine Organisation erstellen, wird der Bucket im Verwaltungskonto Ihrer Organisation erstellt.

Es gibt zwei Anwendungsfälle, in denen Sie anderen AWS Ressourcen die Erlaubnis erteilen müssen, mithilfe von AWS Identity and Access Management (IAM-) Richtlinien auf diesen S3-Bucket zuzugreifen:

- Fall 1: Verwenden Sie Ihr eigenes Instance-Profil oder Ihre eigene Servicerolle mit Ihren verwalteten Knoten, anstatt eines von Quick Setup bereitgestellten
- Fall 2: Verwenden Sie VPC-Endpunkte, um eine Verbindung zu Systems Manager herzustellen

Die Richtlinien für die Berechtigungen, die Sie in beiden Fällen benötigen, finden Sie im folgenden Abschnitt, Richtlinienberechtigungen für Quick Setup-S3-Buckets.

Fall 1: Verwenden Sie Ihr eigenes Instance-Profil oder Ihre eigene Servicerolle mit Ihren verwalteten Knoten, anstatt eines von Quick Setup bereitgestellten

Patch-Richtlinienkonfigurationen enthalten eine Option zum Hinzufügen erforderlicher IAM-Richtlinien zu bestehenden Instance-Profilen, die mit Ihren Instances verbunden sind.

Wenn Sie diese Option nicht wählen, aber möchten, dass Quick Setup Ihre verwalteten Knoten mit dieser Richtlinie patcht, müssen Sie sicherstellen, dass Folgendes implementiert ist:

- Die von IAM verwaltete Richtlinie `AmazonSSMManagedInstanceCore` muss an das IAM-Instance-Profil oder die IAM-Servicerolle angehängt werden, die verwendet wird, um Systems-Manager-Berechtigungen für Ihre verwalteten Knoten bereitzustellen.
- Sie müssen dem IAM-Instance-Profil oder der IAM-Servicerolle Berechtigungen für den Zugriff auf Ihren Patch-Richtlinien-Bucket als Inline-Richtlinie hinzufügen. Sie können Wildcard-Zugriff auf alle

`aws-quicksetup-patchpolicy`-Buckets oder nur auf den spezifischen Bucket gewähren, der für Ihre Organisation oder Ihr Konto erstellt wurde, wie in den früheren Codebeispielen gezeigt.

- Sie müssen Ihr IAM-Instance-Profil oder Ihre IAM-Servicerolle mit dem folgenden Schlüssel-Wert-Paar taggen.

  `Key: QSConfigId-`*`quick-setup-configuration-id`*`, Value: `*`quick-setup-configuration-id`*

  *`quick-setup-configuration-id`*stellt den Wert des Parameters dar, der auf den AWS CloudFormation Stack angewendet wird und der bei der Erstellung Ihrer Patch-Richtlinienkonfiguration verwendet wird. Gehen Sie wie nachfolgend beschrieben vor, um diese ID abzurufen:

  1. Öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
  2. Wählen Sie den Namen des Stacks aus, der zur Erstellung Ihrer Patch-Richtlinie verwendet wird. Der Name hat ein Format wie `StackSet-AWS-QuickSetup-PatchPolicy-LA-q4bkg-52cd2f06-d0f9-499e-9818-d887cEXAMPLE`.
  3. Wählen Sie die Registerkarte Parameters aus.
  4. Suchen Sie in der Parameterliste in der Spalte Schlüssel nach der QSConfiguration Schlüssel-ID. Suchen Sie in der Spalte Wert für die entsprechende Zeile nach der Konfigurations-ID, z. B. `abcde`

     In diesem Beispiel lautet der Schlüssel für das Tag, das auf Ihr Instance-Profil oder Ihre Servicerolle angewendet werden soll `QSConfigId-abcde`, und der Wert lautet `abcde`.

Informationen zum Hinzufügen von Tags zu einer IAM-Rolle finden Sie unter Taggen von IAM-Rollen und Verwalten von Tags in Instanzprofilen (AWS CLI oder AWS APIs) im IAM-Benutzerhandbuch.

Fall 2: Verwenden Sie VPC-Endpunkte, um eine Verbindung zu Systems Manager herzustellen

Wenn Sie VPC-Endpunkte für die Verbindung zu Systems Manager verwenden, muss Ihre VPC-Endpunktrichtlinie für S3 den Zugriff auf Ihren S3-Bucket für Quick Setup-Patch-Richtlinien erlauben.

Informationen zum Hinzufügen von Berechtigungen zu einer VPC-Endpunkt-Richtlinie für S3 finden Sie unter Steuerung des Zugriffs von VPC-Endpunkten mit Bucket-Richtlinien im Amazon S3-Benutzerhandbuch.

Richtlinienberechtigungen für Quick Setup-S3-Buckets

Sie können Wildcard-Zugriff auf alle `aws-quicksetup-patchpolicy`-Buckets oder nur auf den speziellen Bucket gewähren, der für Ihre Organisation oder Ihr Konto erstellt wurde. Verwenden Sie eines der beiden Formate, um die erforderlichen Berechtigungen für die beiden unten beschriebenen Fälle bereitzustellen.

All patch policy buckets

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AccessToAllPatchPolicyRelatedBuckets",
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::aws-quicksetup-patchpolicy-*"
    }
  ]
}
```

Specific patch policy bucket

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AccessToMyPatchPolicyRelatedBucket",
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::aws-quicksetup-patchpolicy-111122223333-quick-
setup-configuration-id"
    }
  ]
}
```

> **Note**
>
> Nachdem die Konfiguration der Patch-Richtlinie erstellt wurde, können Sie den vollständigen Namen Ihres Buckets in der S3-Konsole finden. Beispiel: `aws-quicksetup-patchpolicy-123456789012-abcde`

Zufällige Patch-Baseline bei Patch-Policy-Vorgängen IDs

Patching-Operationen für Patch-Richtlinien verwenden den `BaselineOverride`-Parameter im `AWS-RunPatchBaseline`-SSM-Befehlsdokument.

Wenn Sie `AWS-RunPatchBaseline` zum Patchen außerhalb einer Patch-Richtlinie verwenden, können Sie mit `BaselineOverride` eine Liste von Patch-Baselines angeben, die während des Vorgangs verwendet werden sollen und sich von den angegebenen Standardwerten unterscheiden. Sie erstellen diese Liste in einer Datei mit dem Namen `baseline_overrides.json` und fügen sie manuell zu einem Amazon-S3-Bucket hinzu, den Sie besitzen, wie in Verwenden des BaselineOverride -Parameters erklärt.

Für Patching-Operationen, die auf Patch-Richtlinien basieren, erstellt Systems Manager jedoch automatisch ein S3 Bucket und fügt diesem eine `baseline_overrides.json`-Datei hinzu. Jedes Mal, wenn Quick Setup einen Patching-Vorgang durchführt (unter Verwendung des Run Command-Tools), erzeugt das System eine zufällige ID für jede Patch-Baseline. Diese ID ist für jeden Patch-Vorgang der Richtlinie unterschiedlich, und die Patch-Baseline, die sie repräsentiert, ist in Ihrem Konto weder gespeichert noch für Sie zugänglich.

Daher wird die ID der in Ihrer Konfiguration ausgewählten Patch-Baseline in den Patching-Protokollen nicht angezeigt. Dies gilt sowohl für AWS verwaltete Patch-Baselines als auch für benutzerdefinierte Patch-Baselines, die Sie möglicherweise ausgewählt haben. Die im Protokoll angegebene Baseline-ID ist stattdessen diejenige, die für diesen speziellen Patching-Vorgang erzeugt wurde.

Wenn Sie außerdem versuchen, in Patch Manager Details zu einer Patch-Baseline anzuzeigen, die mit einer zufälligen ID erstellt wurde, meldet das System, dass die Patch-Baseline nicht existiert. Dieses Verhalten ist zu erwarten und kann ignoriert werden.

Erstellen einer Patch-Richtlinie

Führen Sie zum Erstellen einer Patch-Richtlinie die folgenden Aufgaben in der Systems-Manager-Konsole aus.

So erstellen Sie eine Patch-Richtlinie mit Quick Setup

1. Öffnen Sie die Konsole unter. AWS Systems Manager https://console.aws.amazon.com/systems-manager/

   Wenn Sie das Patchen für eine Organisation einrichten, stellen Sie sicher, dass Sie beim Verwaltungskonto der Organisation angemeldet sind. Sie können die Richtlinie nicht mit dem delegierten Administratorkonto oder einem Mitgliedskonto einrichten.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Karte Patch Manager (Patch-Manager) die Option Create (Erstellen) aus.

   > **Tip**
   >
   > Wenn Sie bereits eine oder mehrere Konfigurationen in Ihrem Konto haben, wählen Sie zunächst die Registerkarte Bibliothek oder die Schaltfläche Erstellen im Abschnitt Konfigurationen, um die Karten anzuzeigen.
4. Geben Sie für Configuration name (Konfigurationsname) einen Namen ein, um die Patch-Richtlinie zu identifizieren.
5. Wählen Sie im Abschnitt Scanning and installation (Scannen und Installation) unter Patch operation (Patching-Vorgang) aus, ob die Patch-Richtlinie die angegebenen Ziele Scan (Scannen) oder Patches auf den angegebenen Zielen Scan and install (Scannen und installieren) soll.
6. Wählen Sie unter Scanning schedule (Scan-Zeitplan) die Option Use recommended defaults (Empfohlene Standardwerte verwenden) oder Custom scan schedule (Benutzerdefinierter Scan-Zeitplan) aus. Der standardmäßige Scan-Zeitplan scannt Ihre Ziele täglich um 01:00 Uhr UTC.

   - Wenn Sie Custom scan schedule (Benutzerdefinierten Scan-Zeitplan) auswählen, wählen Sie die Scanning frequency (Scan-Frequenz) aus.
   - Wenn Sie Daily (Täglich) auswählen, geben Sie die Zeit in UTC ein, zu der Sie Ihre Ziele scannen möchten.
   - Wenn Sie Custom CRON Expression (Benutzerdefinierter CRON-Ausdruck) wählen, geben Sie den Zeitplan als CRON expression (CRON-Ausdruck) ein. Weitere Informationen zum Formatieren von CRON-Ausdrücken für Systems Manager finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

Wählen Sie außerdem Wait to scan targets until first CRON interval (Mit dem Scannen von Zielen bis zum ersten CRON-Intervall warten). Standardmäßig scannt Patch Manager Knoten sofort, sobald diese zu Zielen werden.

7. Wenn Sie Scan and install (Scannen und installieren) gewählt haben, wählen Sie den Installation schedule (Installationszeitplan) aus, der beim Installieren von Patches auf den angegebenen Zielen verwendet werden soll. Wenn Sie die Option Use recommended defaults (Empfohlene Standardeinstellungen verwenden) auswählen, installiert Patch Manager die wöchentlichen Patches am Sonntag um 02:00 Uhr UTC.

   - Wenn Sie Custom install schedule (Benutzerdefinierter Installationszeitplan) auswählen, wählen Sie die Installation Frequency (Installationsfrequenz).
   - Wenn Sie Daily (Täglich) auswählen, geben Sie die Zeit in UTC ein, zu der Sie Updates auf Ihren Zielen installieren möchten.
   - Wenn Sie Custom CRON expression (Benutzerdefinierter CRON-Ausdruck) auswählen, geben Sie den Zeitplan als CRON expression (CRON-Ausdruck) ein. Weitere Informationen zum Formatieren von CRON-Ausdrücken für Systems Manager finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.

     Deaktivieren Sie außerdem Wait to install updates until first CRON interval (Mit der Installation von Updates bis zum ersten CRON-Intervall warten), um Updates sofort auf Knoten zu installieren, sobald diese zu Zielen werden. Patch Manager wartet standardmäßig mit der Installation von Updates bis zum ersten CRON-Intervall.
   - Wählen Sie Reboot if needed (Bei Bedarf neu starten), um die Knoten nach der Patch-Installation neu zu starten. Ein Neustart nach der Installation wird empfohlen, kann jedoch zu Verfügbarkeitsproblemen führen.

8. Wählen Sie im Abschnitt Patch baseline (Patch-Baseline) die Patch-Baselines aus, die beim Scannen und Aktualisieren Ihrer Ziele verwendet werden sollen.

   Patch Manager verwendet standardmäßig die vordefinierten Patch-Baselines. Weitere Informationen finden Sie unter Vordefinierte Baselines.

   Wenn Sie Benutzerdefinierte Patch-Baseline wählen, ändern Sie die ausgewählte Patch-Baseline für Betriebssysteme, für die Sie keine vordefinierte AWS Patch-Baseline verwenden möchten.

> **Note**
>
> Wenn Sie VPC-Endpunkte für die Verbindung zu Systems Manager verwenden, stellen Sie sicher, dass Ihre VPC-Endpunktrichtlinie für S3 den Zugriff auf diesen S3-Bucket zulässt. Weitere Informationen finden Sie unter Berechtigungen für den S3-Bucket mit der Patch-Richtlinie.

> **Important**
>
> Wenn Sie eine Patch-Richtlinienkonfiguration in Quick Setup verwenden, werden Aktualisierungen, die Sie an benutzerdefinierten Patch-Baselines vornehmen, einmal pro Stunde mit Quick Setup synchronisiert.
> Wenn eine benutzerdefinierte Patch-Baseline gelöscht wird, auf die in einer Patch-Richtlinie verwiesen wurde, wird auf der Seite mit den Quick Setup-Konfigurationsdetails ein Banner für Ihre Patch-Richtlinie angezeigt. Das Banner informiert Sie darüber, dass die Patch-Richtlinie auf eine nicht mehr vorhandene Patch-Baseline verweist und nachfolgende Patching-Vorgänge fehlschlagen werden. Kehren Sie in diesem Fall zur Seite Quick Setup-Konfigurationen zurück, wählen Sie die Patch Manager-Konfiguration aus und wählen Sie Aktionen, Konfiguration bearbeiten. Der Name der gelöschten Patch-Baseline wird hervorgehoben, und Sie müssen eine neue Patch-Baseline für das betroffene Betriebssystem auswählen.

9. (Optional) Wählen Sie im Abschnitt Patching log storage (Patching-Protokollspeicherung) die Option Write output to S3 bucket (Ausgabe in S3-Bucket schreiben) aus, um Patch-Vorgangsprotokolle in einem Amazon-S3-Bucket zu speichern.

> **Note**
>
> Wenn Sie eine Patch-Richtlinie für eine Organisation einrichten, muss das Verwaltungskonto Ihrer Organisation mindestens über schreibgeschützte Berechtigungen für diesen Bucket verfügen. Alle in der Richtlinie enthaltenen Organisationseinheiten müssen über Schreibzugriff auf den Bucket verfügen. Informationen zum Gewähren von Bucket-Zugriff auf verschiedene Konten finden Sie unter Beispiel 2: Bucket-Besitzer, der kontoübergreifende Bucket-Berechtigungen gewährt im Benutzerhandbuch zu Amazon Simple Storage Service.

10. Wählen Sie S3 durchsuchen, um den Bucket auszuwählen, in dem Sie die Patch-Protokoll-Ausgabe speichern möchten. Das Verwaltungskonto muss über Lesezugriff auf diesen Bucket verfügen. Alle Nicht-Verwaltungskonten und Ziele, die im Abschnitt Targets (Ziele) konfiguriert sind, müssen für die Protokollierung über Schreibzugriff auf den bereitgestellten S3-Bucket verfügen.
11. Wählen Sie im Abschnitt Targets (Ziele) eine der folgenden Optionen aus, um die Konten und Regionen für diesen Patch-Richtlinienvorgang zu identifizieren.

    > **Note**
    >
    > Wenn Sie mit einem einzigen Konto arbeiten, sind Optionen für die Arbeit mit Organisationen und Organisationseinheiten (OUs) nicht verfügbar. Sie können auswählen, ob Sie diese Konfiguration auf alle AWS-Regionen in Ihrem Konto oder nur auf die von Ihnen ausgewählten Regionen anwenden möchten.
    > Wenn Sie zuvor eine Heimatregion für Ihr Konto angegeben haben und das neue Quick Setup-Konsolenerlebnis noch nicht genutzt haben, können Sie diese Region nicht aus der Ziel-Konfiguration ausschließen.

    - Gesamte Organisation – Alle Konten und Regionen in Ihrer Organisation.
    - Benutzerdefiniert — Nur die Regionen OUs und, die Sie angeben.
      - Wählen Sie im OUs Bereich Ziel den OUs Ort aus, an dem Sie die Patch-Richtlinie einrichten möchten.
      - Wählen Sie im Abschnitt Target Regions (Zielregionen) die Regionen aus, in denen Sie die Patch-Richtlinie anwenden möchten.
    - Current account (Aktuelles Konto) – Nur die Regionen, die Sie in dem Konto angeben, bei dem Sie derzeit angemeldet sind, werden als Ziel ausgewählt. Wählen Sie eine der folgenden Optionen:
      - Current Region (Aktuelle Region) – Nur verwaltete Knoten in der Region, die in der Konsole ausgewählt wurde, werden als Ziel ausgewählt.
      - Choose Regions (Regionen auswählen) – Wählen Sie die einzelnen Regionen aus, auf die die Patch-Richtlinie angewendet werden soll.
12. Wählen Sie unter Choose how you want to target instances (Wählen Sie, wie Sie Instances anvisieren möchten) eine der folgenden Möglichkeiten, um die Knoten zu identifizieren, die gepatcht werden sollen:

- Alle verwalteten Knoten — Alle verwalteten Knoten in den ausgewählten OUs und Regionen.
- Specify the resource group (Angabe der Ressourcengruppe) – Wählen Sie den Namen einer Ressourcengruppe aus der Liste, um die ihr zugeordneten Ressourcen anzuvisieren.

> **Note**
>
> Derzeit wird die Auswahl von Ressourcengruppen nur für Einzelkontokonfigurationen unterstützt. Um Ressourcen in mehreren Konten zu patchen, wählen Sie eine andere Zieloption.

- Specify a node tag (Angabe eines Knoten-Tags) – Nur Knoten, die mit dem von Ihnen angegebenen Schlüssel-Wert-Paar gekennzeichnet sind, werden in allen von Ihnen ausgewählten Konten und Regionen gepatcht.
- Manual (Manuell) – Wählen Sie verwaltete Knoten aus allen angegebenen Konten und Regionen manuell aus einer Liste aus.

> **Note**
>
> Diese Option unterstützt derzeit nur EC2 Amazon-Instances. Sie können maximal 25 Instances manuell in einer Patch-Richtlinien-Konfiguration hinzufügen.

13. Gehen Sie im Abschnitt Rate control (Ratensteuerung) wie folgt vor:

    - Geben Sie für Concurrency (Gleichzeitigkeit) eine Anzahl oder einen Prozentsatz von Knoten ein, auf denen die Patch-Richtlinie gleichzeitig ausgeführt werden soll.
    - Geben Sie für Error threshold (Fehlerschwellenwert) die Anzahl oder den Prozentsatz der Knoten ein, bei denen ein Fehler auftreten kann, bevor die Patch-Richtlinie fehlschlägt.

14. (Optional) Aktivieren Sie das Kontrollkästchen Erforderliche IAM-Richtlinien zu bestehenden Instance-Profilen hinzufügen, die mit Ihren Instances verbunden sind.

    Diese Auswahl wendet die durch diese Quick Setup Konfiguration erstellten IAM-Richtlinien auf Knoten an, denen bereits ein Instance-Profil (EC2 Instances) oder eine Service-Rolle (hybrid-aktivierte Knoten) zugewiesen ist. Wir empfehlen diese Auswahl, wenn Ihre verwalteten Knoten bereits über ein Instance-Profil oder eine Servicerolle verfügen, die jedoch nicht alle Berechtigungen enthalten, die für die Arbeit mit Systems Manager erforderlich sind.

Ihre Auswahl hier wird auf verwaltete Knoten angewendet, die später in den Konten und Regionen erstellt werden, für die diese Patch-Richtlinienkonfiguration gilt.

> **Important**
>
> Wenn Sie dieses Kontrollkästchen nicht aktivieren, aber möchten, dass Quick Setup Ihre verwalteten Knoten mit dieser Richtlinie patcht, müssen Sie Folgendes tun:
> Fügen Sie Ihrem IAM-Instance-Profil oder Ihrer IAM-Servicerolle Berechtigungen für den Zugriff auf den S3-Bucket hinzu, der für Ihre Patch-Richtlinie erstellt wurde
> Taggen Sie Ihr IAM-Instance-Profil oder Ihre IAM-Servicerolle mit einem bestimmten Schlüssel-Wert-Paar.
> Weitere Informationen finden Sie unter Fall 1: Verwenden Sie Ihr eigenes Instance-Profil oder Ihre eigene Servicerolle mit Ihren verwalteten Knoten, anstatt eines von Quick Setup bereitgestellten.

15. Wählen Sie Erstellen aus.

    Um den Patch-Status zu überprüfen, nachdem die Patch-Richtlinie erstellt wurde, können Sie über die Quick Setup-Seite auf die Konfiguration zugreifen.

DevOps-Guru einrichten mit Quick Setup

Sie können DevOps Guru-Optionen schnell konfigurieren, indem Sie Quick Setup verwenden. Amazon DevOps Guru ist ein Machine Learning (ML)-gestützter Service, der es leicht macht, die betriebliche Leistung und Verfügbarkeit einer Anwendung zu verbessern. DevOps Guru erkennt Verhaltensweisen, die sich von normalen Betriebsmustern unterscheiden, sodass Sie Betriebsprobleme erkennen können, bevor sie sich auf Ihre Kunden auswirken. DevOps Guru erfasst automatisch Betriebsdaten aus Ihren AWS-Anwendungen und bietet ein einziges Dashboard, um Probleme in Ihren Betriebsdaten zu visualisieren. Sie können mit DevOps Guru beginnen, um die Anwendungsverfügbarkeit und Zuverlässigkeit ohne manuelle Einrichtung oder Machine Learning-Erfahrung zu verbessern.

Konfigurieren von DevOps Guru mit Quick Setup ist in den folgenden AWS-Regionen verfügbar:

- USA Ost (Nord-Virginia)
- USA Ost (Ohio)
- USA West (Oregon)

- Europe (Frankfurt)
- Europa (Irland)
- Europa (Stockholm)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)

Information zu den Preisen finden auf der Seite Amazon DevOps Guru-Preise.

Führen Sie zur Einrichtung von DevOps Guru die folgenden Aufgaben in der AWS Systems Manager Quick Setup-Konsole aus.

Einrichten von DevOps Guru mit Quick Setup

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Karte DevOps Guru die Option Erstellen aus.

   > **Tip**
   >
   > Wenn Sie bereits eine oder mehrere Konfigurationen in Ihrem Konto haben, wählen Sie zunächst die Registerkarte Bibliothek oder die Schaltfläche Erstellen im Abschnitt Konfigurationen, um die Karten anzuzeigen.

4. Wählen Sie im Abschnitt Configuration options (Konfigurationsoptionen) die AWS-Ressourcentypen, die Sie analysieren möchten, und Ihre Benachrichtigungseinstellungen.

   Wenn Sie die Option Analyze all AWS resources in all the accounts in my organization (Alle -Ressourcen in allen Konten in meiner Organisation analysieren) nicht auswählen, können Sie später in der DevOps-Guru-Konsole AWS-Ressourcen zur Analyse auswählen. DevOps Guru analysiert verschiedene AWS-Ressourcentypen (wie Amazon Simple Storage Service (Amazon S3-)Buckets und Amazon Elastic Compute Cloud (Amazon EC2)-Instances, die in zwei Preisgruppen unterteilt sind. Sie zahlen für die analysierten AWS-Ressourcenstunden, für jede aktive Ressource. Eine Ressource ist nur aktiv, wenn sie Metriken, Ereignisse oder Protokolleinträge innerhalb einer Stunde erzeugt. Der Preis, den Sie für einen bestimmten AWS-Ressourcentyp zahlen, ist von der Preisgruppe abhängig.

Wenn Sie die Option Enable SNS notifications (SNS-Benachrichtigungen aktivieren) auswählen, wird ein Amazon Simple Notification Service (Amazon SNS)-Thema in jedem AWS-Konto in den Organisationseinheiten (OUs) erstellt, auf die Sie mit Ihrer Konfiguration abzielen. DevOps Guru verwendet dieses Thema, um Sie über wichtige DevOps Guru-Ereignisse zu informieren, wie z. B. die Erstellung einer neuen Erkenntnis. Wenn Sie diese Option nicht aktivieren, können Sie später in der DevOps Guru-Konsole ein Thema hinzufügen.

Wenn Sie die Option Enable AWS Systems Manager OpsItems wählen, werden operationale Arbeitselemente (OpsItems) für zugehörige Amazon EventBridge-Ereignisse und Amazon CloudWatch-Alarme erstellt.

5. Wählen Sie im Abschnitt Schedule aus, wie häufig Quick Setup Änderungen an Ressourcen, die von Ihrer Konfiguration abweichen, beheben soll. Die Standard-Option wird einmal ausgeführt. Wenn Sie nicht möchten, dass Quick Setup Änderungen an Ressourcen, die von Ihrer Konfiguration abweichen, wiederherstellt, wählen Sie unter Custom (Benutzerdefiniert) die Option Disabled (deaktiviert).
6. Wählen Sie im Abschnitt Targets (Ziele), ob DevOps Guru die Analyse von Ressourcen in einigen Organisationseinheiten (OUs) oder in dem Konto, bei dem Sie gerade angemeldet sind, gestatten soll.

   Fahren Sie mit Schritt 8 fort, wenn Sie Benutzerdefiniert wählen.

   Fahren Sie mit Schritt 9 fort, wenn Sie Custom account (Benutzerdefiniertes Konto) wählen.
7. Aktivieren Sie im Abschnitt Target OUs (Ziel-OUs) und Target Regions (Zielregionen) die Kontrollkästchen der OUs und Regionen, in denen Sie DevOps Guru verwenden möchten.
8. Wählen Sie die Regionen aus, in denen Sie DevOps Guru im aktuellen Konto verwenden möchten.
9. Wählen Sie Erstellen aus.

Bereitstellen von Distributor-Paketen mit Quick Setup

Distributor ist ein Tool in AWS Systems Manager. Ein Distributor-Paket ist eine Sammlung installierbarer Software oder Komponenten, das als einzelne Einheit bereitgestellt werden kann. Mit Quick Setup können Sie eine Distributor-Paket in einem AWS-Konto und einer AWS-Region oder über eine Organisation in AWS Organizations hinweg bereitstellen. Derzeit können nur der EC2Launch v2 Agent, das Amazon Elastic File System (Amazon EFS) Utilities-Paket und der

Amazon CloudWatch Agent mit Quick Setup bereitgestellt werden. Mehr über Distributor erfahren Sie unter AWS Systems Manager Distributor.

Um Distributor-Pakete bereitzustellen, führen Sie in der AWS Systems Manager Quick Setup-Konsole die folgenden Aufgaben aus.

Bereitstellen von Distributor-Paketen mit Quick Setup

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Karte Distributor die Option Erstellen aus.

   > **Tip**
   >
   > Wenn Sie bereits eine oder mehrere Konfigurationen in Ihrem Konto haben, wählen Sie zunächst die Registerkarte Bibliothek oder die Schaltfläche Erstellen im Abschnitt Konfigurationen, um die Karten anzuzeigen.

4. Wählen Sie im Abschnitt Configuration options (Konfigurationsoptionen) das Paket aus, die Sie bereitstellen wollen.
5. Wählen Sie im Abschnitt Targets (Ziele) aus, ob Sie das Paket für Ihre gesamte Organsiation, einige Ihrer Organisationseinheiten (OUs) oder das Konto, bei dem Sie angemeldet sind, bereitstellen möchten.

   Fahren Sie mit Schritt 8 fort, wenn Sie Entire organization (Ganze Organisation) wählen.

   Fahren Sie mit Schritt 7 fort, wenn Sie Custom (Benutzerdefiniert) wählen.
6. Aktivieren Sie im Abschnitt Target OUs die Kontrollkästchen der OUs (Organisationseinheiten) und Regionen, für die Sie das Paket bereitstellen möchten.
7. Wählen Sie Erstellen aus.

Automatisches Stoppen und Starten von EC2-Instances nach einem Zeitplan mit Quick Setup

Mit Quick Setup, einem Tool in AWS Systems Manager, können Sie Resource Scheduler konfigurieren, um das Starten und Beenden von Amazon Elastic Compute Cloud (Amazon EC2)-Instances zu automatisieren.

Diese Quick Setup-Konfiguration unterstützt Sie bei der Senkung der Betriebskosten, indem Instances gemäß dem von Ihnen festgelegten Zeitplan gestartet und beendet werden. Dieses Tool hilft Ihnen, unnötige Kosten für die Ausführung von Instances zu vermeiden, wenn diese nicht benötigt werden.

So kann es beispielsweise sein, dass Sie Ihre Instances ständig ausführen lassen, obwohl sie nur 10 Stunden pro Tag und 5 Tage pro Woche verwendet werden. Stattdessen können Sie Ihre Instances so planen, dass sie jeden Tag nach den Geschäftszeiten beendet werden. Das Ergebnis wäre eine Einsparung von 70 Prozent für diese Instances, da die Ausführung von 168 Stunden auf 50 Stunden reduziert wird. Bei der Nutzung des Quick Setup-Service fallen keine Kosten an. Kosten können jedoch durch die von Ihnen eingerichteten Ressourcen und die Nutzungsbeschränkungen entstehen, wobei keine Gebühren für die Dienste anfallen, die zur Einrichtung Ihrer Konfiguration verwendet werden.

Mit Resource Scheduler können Sie festlegen, dass Instances über mehrere AWS-Regionen und AWS-Konten nach einem von Ihnen definierten Zeitplan automatisch beendet und gestartet werden. Die Quick Setup-Konfiguration verwendet den von Ihnen angegebenen Tag-Schlüssel und Wert für Amazon-EC2-Instances. Nur die Instances mit einem Tag, das mit dem Wert übereinstimmt, den Sie in Ihrer Konfiguration angeben, werden vom Resource Scheduler beendet oder gestartet. Beachten Sie, dass Sie, wenn die an die Instance angehängten Amazon-EBS-Volumes verschlüsselt sind, der IAM-Rolle die erforderlichen Berechtigungen für den AWS KMS-Schlüssel hinzufügen müssen, damit Resource Scheduler die Instance starten kann.

Maximale Anzahl von Instances pro Konfiguration

Eine individuelle Konfiguration unterstützt die Zeitplanung von bis zu 5 000 Instances pro Region. Wenn in Ihrem Fall mehr als 5 000 Instances in einer bestimmten Region geplant werden müssen, müssen Sie mehrere Konfigurationen erstellen. Kennzeichnen Sie Ihre Instances entsprechend, damit jede Konfiguration bis zu 5 000 Instances verwalten kann. Beim Erstellen mehrerer Quick Setup-Konfigurationen für Resource Scheduler müssen Sie verschiedene Tag-Schlüsselwerte angeben. Beispielsweise kann eine Konfiguration den Tag-Schlüssel `Environment` mit dem Wert `Production` verwenden, während eine andere `Environment` und `Development` verwendet.

Verhaltensweisen bei der Zeitplanung

In den folgenden Punkten werden bestimmte Verhaltensweisen von Zeitplankonfigurationen beschrieben:

- Resource Scheduler startet die gekennzeichneten Instances nur, wenn sich diese im `Stopped`-Status befinden. Ebenso werden Instances nur dann beendet, wenn sie sich im `running`-Status

befinden. Resource Scheduler arbeitet nach einem ereignisgesteuerten Modell und startet oder beendet Instances nur zu den von Ihnen festgelegten Zeiten. Sie erstellen beispielsweise einen Zeitplan, der Instances um 9:00 Uhr startet. Resource Scheduler startet alle Instances, die dem von Ihnen angegebenen Tag zugeordnet sind und sich im `Stopped`-Status befinden, um 09:00 Uhr. Wenn die Instances zu einem späteren Zeitpunkt manuell angehalten werden, startet Resource Scheduler diese nicht erneut, um den `Running`-Status beizubehalten. Wenn eine Instance manuell gestartet wird, nachdem sie gemäß Ihrem Zeitplan angehalten wurde, wird Resource Scheduler die Instance nicht erneut anhalten.

- Wenn Sie einen Zeitplan mit einer Startzeit erstellen, die an einem 24-stündigen Tag nach der Anhaltezeit liegt, geht Resource Scheduler davon aus, dass Ihre Instances über Nacht ausgeführt werden. Sie erstellen beispielsweise einen Zeitplan, der Instances um 21:00 Uhr startet und Instances um 07:00 Uhr beendet. Resource Scheduler startet alle Instances, die dem von Ihnen angegebenen Tag zugeordnet sind und sich im `Stopped`-Status befinden, um 21:00 Uhr und beendet sie um 07:00 Uhr am nächsten Tag. Bei Nachtplänen gilt die Startzeit für die Tage, die Sie für Ihren Zeitplan auswählen. Die Anhaltezeit gilt jedoch für den folgenden Tag in Ihrem Zeitplan.
- Wenn Sie eine Zeitplankonfiguration erstellen, kann der aktuelle Status Ihrer Instances an die Anforderungen des Zeitplans angepasst werden.

  Angenommen, heute ist ein Mittwoch und Sie geben einen Zeitplan an, nach dem Ihre verwalteten Instances nur dienstags und donnerstags um 9 Uhr beginnen und um 17 Uhr enden. Da Sie sich aktuell außerhalb der vorgeschriebenen Ausführungszeiten für die Instances befinden, werden sie nach der Erstellung der Konfiguration gestoppt. Die Instances werden erst zur nächsten vorgeschriebenen Stunde, am Donnerstag um 9 Uhr, wieder ausgeführt.

  Wenn sich Ihre Instances derzeit in einem `Stopped`-Status befinden und Sie einen Zeitplan angeben, nach dem sie zum aktuellen Zeitpunkt ausgeführt werden sollen, startet Resource Scheduler sie, nachdem die Konfiguration erstellt wurde.

Wenn Sie Ihre Konfiguration löschen, werden Instances nicht mehr gemäß dem zuvor definierten Zeitplan beendet und gestartet. In seltenen Fällen werden Instances aufgrund von API-Operationsfehlern möglicherweise nicht erfolgreich beendet oder gestartet.

Führen Sie die folgenden Aufgaben in der AWS Systems Manager Quick Setup-Konsole aus, um die Zeitplanung für Amazon-EC2-Instances einzurichten.

So richten Sie die Instance-Zeitplanung mit Quick Setup ein

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Karte Resource Scheduler die Option Erstellen aus.

   > **Tip**
   >
   > Wenn Sie bereits eine oder mehrere Konfigurationen in Ihrem Konto haben, wählen Sie zunächst die Registerkarte Bibliothek oder die Schaltfläche Erstellen im Abschnitt Konfigurationen, um die Karten anzuzeigen.

4. Geben Sie im Abschnitt Instance tag (Instance-Tag) den Tag-Schlüssel und den Wert für die Instances an, die Sie Ihrem Zeitplan zuordnen möchten.
5. Geben Sie im Abschnitt Schedule options (Zeitplanoptionen) die Zeitzone, die Tage und die Uhrzeiten an, zu denen Sie Ihre Instances starten und beenden möchten.
6. Wählen Sie im Abschnitt Targets (Ziele) aus, ob Sie die Zeitplanung für eine Custom (Benutzerdefinierte) Gruppe von Organisationseinheiten (OUs) oder das Current account (Aktuelle Konto), bei dem Sie angemeldet sind, einrichten möchten:

   - Custom (Benutzerdefiniert) – Wählen Sie im Abschnitt Target OUs (Ziel-Organisationseinheiten) die Organisationseinheiten aus, in denen Sie die Zeitplanung einrichten möchten. Wählen Sie als Nächstes im Abschnitt Target Regions (Zielregionen) die Regionen aus, in denen Sie die Zeitplanung einrichten möchten.
   - Current account (aktuelles Konto)— Wählen Sie Current Region (aktuelle Region) oder Choose Regions (Regionen wählen). Wenn Sie Choose Regions (Regionen auswählen) ausgewählt haben, wählen Sie die Target Regions (Zielregionen) aus, in denen Sie die Zeitplanung einrichten möchten.

7. Überprüfen Sie die Informationen zum Zeitplan im Abschnitt Summary (Zusammenfassung).
8. Wählen Sie Erstellen aus.

Konfigurieren von AWS Resource Explorer mithilfe von Quick Setup

Mit Quick Setup, einem Tool in AWS Systems Manager, können Sie AWS Resource Explorer schnell konfigurieren, um Ressourcen in Ihrem AWS-Konto oder in der gesamten AWS-Organisation zu

suchen und zu finden. Sie können mithilfe von Metadaten wie Namen, Tags und IDs nach Ihren Ressourcen suchen. AWS Resource Explorer bietet mithilfe von Indizes schnelle Antworten auf Ihre Suchanfragen. Resource Explorer erstellt und verwaltet Indizes mithilfe einer Vielzahl von Datenquellen, um Informationen über Ressourcen in Ihrem AWS-Konto zu sammeln.

Quick Setup für Resource Explorer automatisiert den Indexkonfigurationsprozess. Weitere Informationen zu AWS Resource Explorer finden Sie unter Was ist AWS Resource Explorer? im Benutzerhandbuch für AWS Resource Explorer.

Während des Quick Setup führt Resource Explorer die folgenden Schritte aus:

- Erstellt in jeder AWS-Region Ihres AWS-Konto einen Index.
- Aktualisiert den Index in der Region, die Sie als Aggregatorindex für das Konto angeben.
- Erstellt eine Standardansicht in der Aggregator-Index-Region. Diese Ansicht hat keine Filter und gibt daher alle im Index gefundenen Ressourcen zurück.

Mindestberechtigungen

Um die Schritte im folgenden Verfahren auszuführen, müssen Sie über die folgenden Berechtigungen verfügen:

- Aktion: `resource-explorer-2:*` – Ressource: keine spezifische Ressource (`*`)
- Aktion: `iam:CreateServiceLinkedRole` – Ressource: keine spezifische Ressource (`*`)

So konfigurieren Sie Resource Explorer

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Karte Resource Explorer die Option Erstellen aus.
4. Wählen Sie im Abschnitt Aggregator-Index-Region aus, welche Region der Aggregatorindex enthalten soll. Sie sollten die Region auswählen, die für den geografischen Standort Ihrer Benutzer geeignet ist.
5. (Optional) Aktivieren Sie das Kontrollkästchen Vorhandene Aggregatorindizes in anderen als den oben ausgewählten Regionen ersetzen.
6. Wählen Sie im Abschnitt Ziele die Zielorganisation oder bestimmte Organisationseinheiten (OUs) aus, die die Ressourcen enthalten, die Sie ermitteln möchten.

7. Wählen Sie im Abschnitt Regionen aus, welche Regionen in die Konfiguration aufgenommen werden sollen.
8. Überprüfen Sie die Konfigurationszusammenfassung, und wählen Sie dann Erstellen.

Auf der Resource-Explorer-Seite können Sie den Konfigurationsstatus überwachen.

Fehlerbehebung von Quick Setup-Ergebnissen

Im Folgenden finden Sie Informationen zur Behandlung von Problemen mit Quick Setup, einem Tool in AWS Systems Manager. Dieses Thema umfasst je nach Art des Problems spezifische Aufgaben zur Lösung von Quick Setup-Problemen.

Problem: Fehlgeschlagene Bereitstellung

Eine Bereitstellung schlägt fehl, wenn der Stack-Set von CloudFormation während der Erstellung fehlgeschlagen ist. Gehen Sie wie folgt vor, um einen Bereitstellungsfehler zu untersuchen.

1. Navigieren Sie zur AWS CloudFormation-Konsole.
2. Wählen Sie den Stack aus, der von der Quick Setup-Konfiguration erstellt wurde. Der Stack name (Stack-Name) beinhaltet `QuickSetup`, gefolgt von der Art der ausgewählten Konfiguration, wie etwa `SSMHostMgmt`.

   > **Note**
   >
   > Manchmal löscht CloudFormation fehlgeschlagene Stack-Bereitstellungen. Wenn der Stack in der Tabelle Stacks nicht verfügbar ist, wählen Sie Deleted (Gelöscht) in der Filterliste aus.

3. Zeigen Sie den Status und den Statusgrund an. Weitere Informationen zum Stack-Status finden Sie unter Stack-Statuscodes im Benutzerhandbuch von AWS CloudFormation.
4. Um nachzuvollziehen, welcher Schritt genau fehlgeschlagen ist, sehen Sie sich auf der Registerkarte Events (Ereignisse) den Status der einzelnen Ereignisse an.
5. Lesen Sie den Abschnitt Fehlerbehebung im Benutzerhandbuch von AWS CloudFormation.
6. Wenn sich der Bereitstellungsfehler mit den CloudFormation-Schritten zur Fehlerbehebung nicht beheben lässt, löschen Sie die Konfiguration und erstellen Sie sie neu.

Problem: Fehlgeschlagene Zuordnung

Die Tabelle Configuration details (Konfigurationsdetails) auf der Seite Configuration details Ihrer Konfiguration zeigt als Configuration status (Konfigurationsstatus) Failed (Fehlgeschlagen) an, wenn eine der Zuordnungen bei der Einrichtung fehlgeschlagen ist. Gehen Sie zur Fehlerbehebung einer fehlgeschlagenen Zuordnung wie folgt vor.

1. Wählen Sie in der Tabelle Configuration details (Konfigurationsdetails) die fehlgeschlagene Konfiguration und dann View Details (Details anzeigen) aus.
2. Kopieren Sie den Association name (Zuordnungsnamen).
3. Navigieren Sie zu State Manager und fügen Sie den Zuordnungsnamen in das Suchfeld ein.
4. Wählen Sie die Zuordnung und dann die Registerkarte Execution history (Ausführungsverlauf) aus.
5. Wählen Sie unter Execution ID (Ausführungs-ID) die Zuordnungsausführung aus, die fehlgeschlagen ist.
6. Auf der Seite Association execution targets (Zuordnungs-Ausführungsziele) werden alle Knoten aufgelistet, auf denen die Zuordnung ausgeführt wurde. Wählen Sie die Schaltfläche Output (Ausgabe) für eine fehlgeschlagene Ausführung aus.
7. Wählen Sie auf der Seite Output (Ausgabe) Step – Output (Schritt – Ausgabe) aus, um die Fehlermeldung für diesen Schritt in der Befehlsausführung anzuzeigen. Jeder Schritt kann eine andere Fehlermeldung anzeigen. Überprüfen Sie die Fehlermeldungen für alle Schritte, um das Problem zu beheben.

Wenn sich das Problem durch die Anzeige der Schrittausgabe nicht beheben lässt, können Sie versuchen, die Zuordnung neu zu erstellen. Um die Zuordnung neu zu erstellen, löschen Sie zunächst die fehlgeschlagene Zuordnung in State Manager. Bearbeiten Sie anschließend die Konfiguration, wählen Sie die von Ihnen gelöschte Option aus und klicken Sie auf Update (Aktualisieren).

> **Note**
>
> Um fehlgeschlagene Zuordnungen für die Konfiguration einer Organisation zu untersuchen, müssen Sie sich bei dem Konto mit der fehlgeschlagenen Zuordnung anmelden und das zuvor beschriebene Verfahren für fehlgeschlagene Zuordnungen anwenden. Die Association ID (Zuordnungs-ID) ist kein Hyperlink zum Zielkonto beim Anzeigen von Ergebnissen vom Verwaltungskonto.

Problem: Drift-Status

Auf der Detailseite einer Konfiguration können Sie den Abweichungsstatus der einzelnen Bereitstellungen anzeigen. Eine Konfigurationsabweichung tritt auf, wenn Benutzer Änderungen an einem Service oder einer Funktion vornehmen, die mit der Auswahl in Quick Setup im Konflikt stehen. Wenn sich eine Zuordnung nach der Erstkonfiguration geändert hat, zeigt die Tabelle ein Warnsymbol, das die Anzahl der abweichenden Elemente angibt. Sie können die Ursache der Abweichung feststellen, indem Sie den Mauszeiger über das Symbol bewegen.

Wenn eine Zuordnung in State Manager gelöscht wird, zeigen die zugehörigen Bereitstellungen eine Abweichungswarnung an. Bearbeiten Sie zur Behebung dieses Problems die Konfiguration und wählen Sie die Option aus, die beim Löschen der Zuordnung entfernt wurde. Wählen Sie Update (Aktualisieren) aus und warten Sie, bis die Bereitstellung abgeschlossen ist.

AWS Systems Manager Anwendungstools

Application Tools ist eine Suite von Funktionen, mit denen Sie Ihre in AWS ausgeführten Anwendungen verwalten können.

Themen

- AWS AppConfig
- AWS Systems Manager Application Manager
- AWS Systems Manager Parameter Store

AWS AppConfig

Informationen zu AWS AppConfig wurden in separate Benutzerhandbücher verschoben. Weitere Informationen finden Sie hier:

- AWS AppConfig-Benutzerhandbuch
- AWS AppConfig-API-Referenz

AWS Systems Manager Application Manager

Application Manager, ein Tool in AWS Systems Manager, hilft DevOps Technikern dabei, Probleme mit ihren AWS Ressourcen im Kontext ihrer Anwendungen und Cluster zu untersuchen und zu

beheben. Application Managerfasst Betriebsinformationen aus mehreren Tools AWS-Services und Systems Manager Manager-Tools in einem einzigen AWS-Managementkonsole zusammen.

Bei Application Manager einer Anwendung handelt es sich um eine logische Gruppe von AWS Ressourcen, die Sie als Einheit betreiben möchten. Diese logische Gruppe kann verschiedene Versionen einer Anwendung, Besitzgrenzen für Operatoren oder Entwicklerumgebungen darstellen, um nur einige zu nennen. Application Manager Unterstützung für Container-Cluster umfasst sowohl Amazon Elastic Kubernetes Service (Amazon EKS) als auch Amazon Elastic Container Service (Amazon ECS) Cluster.

Beim ersten Öffnen von Application Manager wird die Seite Was Application Manager für Sie tun kann angezeigt. Wenn Sie Erste Schritte wählen, Application Manager werden automatisch Metadaten zu Ihren Ressourcen importiert, die in anderen Tools AWS-Services oder Systems Manager Manager-Tools erstellt wurden. Application Managerzeigt diese Ressourcen dann in einer Liste an, die nach vordefinierten Kategorien gruppiert ist.

Für Anwendungen umfasst die Liste Folgendes:

- AWS CloudFormation stapelt
- Benutzerdefinierte Anwendungen
- AWS Launch Wizard Anwendungen
- AppRegistry Anwendungen
- AWS SAP Enterprise Workload-Anwendungen
- Amazon ECS-Cluster
- Amazon EKS-Cluster

Application ManagerZeigt nach der Einrichtung AWS-Services und Konfiguration der Systems Manager Manager-Tools die folgenden Arten von Informationen zu Ihren Ressourcen an:

- Informationen über den aktuellen Status, den Status und den Zustand von Amazon EC2 Auto Scaling der Amazon Elastic Compute Cloud (Amazon EC2) -Instances in Ihrer Anwendung
- Von Amazon bereitgestellte Alarme CloudWatch
- Konformitätsinformationen, bereitgestellt von AWS Config und State Manager (eine Komponente von Systems Manager)
- Kubernetes-Clusterinformationen, die von Amazon EKS bereitgestellt werden
- Von Amazon Logs AWS CloudTrail bereitgestellte CloudWatch Protokolldaten

- OpsItemsvon Systems Manager bereitgestellt OpsCenter
- Ressourcendetails, die von denjenigen bereitgestellt werden AWS-Services , die sie hosten.
- Container-Cluster-Informationen, die von Amazon ECS bereitgestellt werden.

Um Probleme mit Komponenten oder Ressourcen zu beheben, stellt Application Manager Ihnen auch Runbooks bereit, die Sie Ihren Anwendungen zuordnen können. Um mit Application Manager zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Application Manager aus.

Was sind die Vorteile der Nutzung von Application Manager?

Application Managerreduziert die Zeit, die DevOps Techniker benötigen, um Probleme mit AWS Ressourcen zu erkennen und zu untersuchen. Application ManagerZeigt zu diesem Zweck viele Arten von Betriebsinformationen im Kontext einer Anwendung in einer Konsole an. Application Managerreduziert außerdem den Zeitaufwand für die Problembehebung, indem Runbooks bereitgestellt werden, die allgemeine Problembehebungsaufgaben für Ressourcen ausführen. AWS

Über welche Features verfügt Application Manager?

Application Manager umfasst die folgenden Features:

- Importieren Sie Ihre Ressourcen automatisch AWS

  Bei der Ersteinrichtung können Sie festlegen, AWS-Konto dass Ressourcen, die auf CloudFormation Stacks, Launch Wizard Wizard-Bereitstellungen AWS -Ressourcengruppen, AppRegistry Anwendungen und Amazon ECS- und Amazon EKS-Clustern basieren, Application Manager automatisch importiert und angezeigt werden. Das System zeigt diese Ressourcen in vordefinierten Anwendungs- oder Clusterkategorien an. Wenn danach neue Ressourcen dieser Art zu Ihrem hinzugefügt werden AWS-Konto, werden die neuen Ressourcen Application Manager automatisch in den vordefinierten Anwendungs- und Clusterkategorien angezeigt.

- CloudFormation Stapel und Vorlagen erstellen oder bearbeiten

  Application Managerhilft Ihnen bei der Bereitstellung und Verwaltung von Ressourcen für Ihre Anwendungen durch die Integration mit CloudFormation. Sie können CloudFormation Vorlagen und Stacks in Application Manager erstellen, bearbeiten und löschen. Application Managerenthält auch eine Vorlagenbibliothek, in der Sie Vorlagen klonen, erstellen und speichern können. Application Managerund CloudFormation zeigt dieselben Informationen über den aktuellen Status eines Stacks

an. Vorlagen und Vorlagenaktualisierungen werden in Systems Manager gespeichert, bis Sie den Stack bereitstellen. Zu diesem Zeitpunkt werden die Änderungen auch angezeigt CloudFormation.

- Informationen zu Ihren Instances im Kontext einer Anwendung anzeigen

  Application Managerintegriert sich in Amazon Elastic Compute Cloud (Amazon EC2), um Informationen über Ihre Instances im Kontext einer Anwendung anzuzeigen. Application Managerzeigt den Instance-Status, den Status und den Zustand von Amazon EC2 Auto Scaling für eine ausgewählte Anwendung in einem grafischen Format an. Die Registerkarte Instances enthält auch eine Tabelle mit den folgenden Informationen für jede Instance in Ihrer Anwendung.

  - Instance-Status (Ausstehend, Angehalten, Wird ausgeführt, Beendet)
  - Ping-Status für SSM Agent
  - Status und Name des letzten Systems-Manager-Automation-Runbooks, das auf der Instance verarbeitet wurde
  - Eine Anzahl von Amazon CloudWatch Logs-Alarmen pro Bundesstaat.
    - `ALARM` – Die Metrik oder der Ausdruck liegt außerhalb des festgelegten Schwellenwerts.
    - `OK` – Die Metrik oder der Ausdruck liegt innerhalb des festgelegten Schwellenwerts.
    - `INSUFFICIENT_DATA` – Der Alarm wurde soeben gestartet; die Metrik ist nicht verfügbar oder es sind nicht genügend Daten verfügbar, damit die Metrik den Alarmstatus bestimmen kann.
  - Zustand der Auto-Scaling-Gruppe für die übergeordneten und einzelnen Auto-Scaling-Gruppen
- Anzeigen von Betriebsmetriken und Alarmen für eine Anwendung oder ein Cluster

  Application Managerlässt sich in Amazon integrieren CloudWatch, um Betriebsmetriken und Alarme in Echtzeit für eine Anwendung oder einen Cluster bereitzustellen. Sie können einen Drilldown in Ihre Anwendungsstruktur durchführen, um Alarme auf jeder Komponentenebene anzuzeigen oder Alarme für einen einzelnen Cluster anzuzeigen.
- Anzeigen von Protokolldaten für eine Anwendung

  Application Managerlässt sich in Amazon CloudWatch Logs integrieren, um Protokolldaten im Kontext Ihrer Anwendung bereitzustellen, ohne Systems Manager verlassen zu müssen.
- Anzeigen und Verwalten von OpsItems für eine Anwendung oder ein Cluster

  Application Managerintegriert AWS Systems Manager OpsCenter, um eine Liste der operativen Arbeitselemente (OpsItems) für Ihre Anwendungen und Cluster bereitzustellen. Die Liste spiegelt automatisch generierte und manuell erstellte OpsItems. Sie können Details über die Ressource anzeigen, die eine OpsItem und den OpsItem Status, Quelle und Schweregrad erstellen.

- Anzeigen von Ressourcen-Compliance-Daten für eine Anwendung oder Cluster

  Application Managerlässt sich integrieren AWS Config, um Konformitäts- und Verlaufsinformationen zu Ihren AWS Ressourcen gemäß den von Ihnen festgelegten Regeln bereitzustellen. Application Managerlässt sich auch integrierenAWS Systems Manager State Manager, um Compliance-Informationen über den Status bereitzustellen, den Sie für Ihre Amazon Elastic Compute Cloud (Amazon EC2) -Instances beibehalten möchten.

- Informationen zu Amazon ECS und Amazon EKS Cluster-Infrastruktur anzeigen

  Application Manager integriert in Amazon ECS und Amazon EKS, um Informationen über den Zustand Ihrer Cluster-Infrastrukturen und eine Komponentenlaufzeitansicht der Computing-, Netzwerk- und Speicherressourcen in einem Cluster bereitzustellen.

  Sie können jedoch keine Betriebsinformationen zu Ihren Amazon EKS-Pods oder -Containern in Application Manager verwalten. Sie können nur Betriebsinformationen zu der Infrastruktur verwalten und anzeigen, die Ihre Amazon EKS-Ressourcen hostet.

- Anzeigen von Ressourcen-Preisdetails für eine Anwendung

  Application Managerist über das Kosten-Widget in AWS Cost Explorer AWS Fakturierung und Kostenmanagement, eine Funktion von, integriert. Nachdem Sie den Cost Explorer in der Fakturierungs- und Kostenmanagement-Konsole aktiviert haben, zeigt das Cost-Widget in Application Manager Preisdaten für eine bestimmte Anwendung oder Anwendungskomponente ohne Container an. Sie können Filter im Widget verwenden, um Preisdaten nach verschiedenen Zeiträumen, Details und Preisarten in einem Balken- oder Liniendiagramm anzuzeigen.

- Anzeigen detaillierter Ressourceninformationen in einer Konsole

  Wählen Sie einen Ressourcennamen aus, der unter Application Manager gelistet ist und zeigen Sie kontextbezogene Informationen und Betriebsinformationen zu dieser Ressource an, ohne Systems Manager verlassen zu müssen.

- Erhalten Sie automatische Ressourcenaktualisierungen für Anwendungen

  Wenn Sie Änderungen an einer Ressource in einer Dienstkonsole vornehmen und diese Ressource Teil einer Anwendung in Application Manager ist, werden diese Änderungen automatisch von Systems Manager angezeigt. Wenn Sie beispielsweise einen Stack in der CloudFormation Konsole aktualisieren und dieser Stack Teil einer Application Manager Anwendung ist, werden die Stack-Aktualisierungen automatisch in übernommenApplication Manager.

- Entdecken Sie Launch Wizard-Anwendungen automatisch

Application Manager ist in AWS Launch Wizard integriert. Wenn Sie den Launch Wizard zum Bereitstellen von Ressourcen für eine Anwendung verwendet haben, kann Application Manager diese automatisch importieren und in einem Abschnitt des Launch Wizard anzeigen.

- Überwachen Sie Anwendungsressourcen mithilfe Application Manager von CloudWatch Application Insights

  Application Managerintegriert sich in Amazon CloudWatch Application Insights. Application Insights identifiziert Schlüsselmetriken, Protokolle und Alarme und richtet diese für Ihre Anwendungsressourcen und Ihren Technologie-Stack ein. Application Insights überwacht kontinuierlich Metriken und Protokolle, um Anomalien und Fehler zu erkennen und zu korrelieren. Wenn das System Fehler oder Anomalien erkennt, generiert Application Insights CloudWatch Ereignisse, anhand derer Sie Benachrichtigungen einrichten oder Maßnahmen ergreifen können. Sie können Application Insights auf den Tabs Übersicht und Überwachungin Application Manager aktivieren und ansehen. Weitere Informationen zu Application Insights finden Sie unter Was ist Amazon CloudWatch Application Insights im CloudWatch Amazon-Benutzerhandbuch.
- Beheben von Problemen mit Runbooks

  Application Managerenthält vordefinierte Systems Manager Manager-Runbooks zur Behebung häufiger Probleme mit AWS Ressourcen. Sie können ein Runbook für alle anwendbaren Ressourcen in einer Anwendung ausführen, ohne Application Manager verlassen zu müssen.

Entstehen Kosten für die Verwendung von Application Manager?

Application Manager ist ohne Aufpreis erhältlich.

Was sind die Ressourcenkontingente für Application Manager?

Sie können Kontingente für alle Systems-Manager-Tools unter Service Quotas für Systems Manager im Allgemeine Amazon Web Services-Referenz anzeigen. Wenn nicht anders angegeben, gilt jedes Kontingent spezifisch für eine Region.

Themen

- Einrichten von zugehörigen Services
- Konfigurieren von Berechtigungen für Systems Manager Application Manager
- Hinzufügen von Anwendungen und Container-Clustern zu Application Manager
- Arbeiten mit -Anwendungen

Einrichten von zugehörigen Services

Application Manager, ein Tool in AWS Systems Manager, zeigt Ressourcen und Informationen aus anderen Tools AWS-Services und Systems Manager Manager-Tools an. Um die Menge der Arbeitsvorgangsinformationen zu maximieren, die in Application Manager angezeigt werden, empfehlen wir, diese anderen Services oder Tools einzurichten und zu konfigurieren bevor Sie Application Manager verwenden.

Themen

- Einrichten von Aufgaben zum Importieren von Ressourcen
- Einrichten von Aufgaben zum Anzeigen von Vorgangsinformationen zu Ressourcen

Einrichten von Aufgaben zum Importieren von Ressourcen

Die folgenden Einrichtungsaufgaben helfen Ihnen beim Anzeigen von AWS Ressourcen inApplication Manager. Nachdem jede dieser Aufgaben abgeschlossen ist, kann Systems Manager Ressourcen automatisch in Application Manager importieren. Nachdem Ihre Ressourcen importiert wurden, können Sie Anwendungen in Application Manager erstellen und Ihre importierten Ressourcen in sie verschieben. So können Sie Betriebsinformationen im Kontext einer Anwendung anzeigen.

(Optional) Organisieren Sie Ihre AWS Ressourcen mithilfe von Tags

Sie können Ihren AWS Ressourcen Metadaten in Form von Tags zuweisen. Jedes Tag ist eine Bezeichnung, die aus einem benutzerdefinierten Schlüssel und Wert besteht. Mit Tags können Sie Ressourcen verwalten, identifizieren, organisieren, suchen und filtern. Sie können Tags erstellen, um Ressourcen nach Zweck, Eigentümer, Umgebung oder anderen Kriterien zu kategorisieren.

(Optional) Organisiert Ihre AWS Ressourcen mithilfe von AWS -Ressourcengruppen

Sie können Ressourcengruppen verwenden, um Ihre AWS Ressourcen zu organisieren. Ressourcengruppen vereinfachen die gleichzeitige Verwaltung, Überwachung und Automatisierung von Aufgaben für viele Ressourcen.

Application Manager importiert automatisch alle Ressourcengruppen und listet sie in der Kategorie Benutzerdefinierte Anwendungen.

(Optional) Richten Sie Ihre AWS Ressourcen ein und stellen Sie sie bereit, indem Sie AWS CloudFormation

CloudFormation ermöglicht es Ihnen, AWS Infrastrukturbereitstellungen vorhersehbar und wiederholt zu erstellen und bereitzustellen. Es hilft Ihnen bei der Verwendung AWS-Services von

Amazon EC2, Amazon Elastic Block Store (Amazon EBS), Amazon Simple Notification Service (Amazon SNS), ELB und AWS Auto Scaling. Mit können Sie zuverlässige CloudFormation, skalierbare und kostengünstige Anwendungen in der Cloud erstellen, ohne sich Gedanken über die Erstellung und Konfiguration der zugrunde liegenden AWS Infrastruktur machen zu müssen.

Application Managerimportiert automatisch alle Ihre CloudFormation Ressourcen und listet sie in der Kategorie CloudFormation Stacks auf. Sie können CloudFormation Stapel und Vorlagen in erstellen. Application Manager Stapel- und Vorlagenänderungen werden automatisch zwischen Application Manager und CloudFormation synchronisiert. Sie können auch Anwendungen in Application Manager erstellen und Stacks in diese verschieben. Auf diese Weise können Sie Betriebsinformationen für Ressourcen in Ihren Stacks im Kontext einer Anwendung anzeigen. Preisinformationen finden Sie unter CloudFormation – Preise.

(Optional) Richten Sie Ihre Anwendungen ein und stellen Sie sie bereit, indem Sie AWS Launch Wizard

Der Launch Wizard führt Sie durch den Prozess der Dimensionierung, Konfiguration und Bereitstellung von AWS Ressourcen für Drittanbieteranwendungen, ohne dass Sie einzelne AWS Ressourcen manuell identifizieren und bereitstellen müssen.

Application Manager importiert automatisch all Ihre Launch Wizard-Ressourcen und listet sie in der Kategorie Launch Wizard. Weitere Informationen zu AWS Launch Wizard finden Sie unter Erste Schritte mit AWS Launch Wizard für SQL Server. Launch Wizard ist ohne Aufpreis erhältlich. Sie zahlen nur für die AWS Ressourcen, die Sie für den Betrieb Ihrer Lösung bereitstellen.

(Optional) Richten Sie Ihre containerisierten Anwendungen mithilfe von Amazon ECS und Amazon EKS ein und stellen diese bereit.

Amazon Elastic Container Service (Amazon ECS) ist ein hoch skalierbarer, schneller Container-Management-Service, der das Ausführen, Beenden und Verwalten von Containern in einem Cluster vereinfacht. Ihre Container sind in einer Aufgabendefinition definiert, die Sie zum Ausführen einzelner Aufgaben oder Aufgaben innerhalb eines Dienstes verwenden.

Amazon EKS ist ein verwalteter Service, der Ihnen hilft, Kubernetes auf AWS auszuführen, ohne Ihre eigene Kubernetes-Steuerebene oder -Knoten zu installieren, zu betreiben und zu warten. Kubernetes ist ein Open-Source-System zur Automatisierung der Bereitstellung, Skalierung und Verwaltung von Anwendungen in Containern.

Application Manager importiert automatisch alle Ihre Amazon ECS- und Amazon EKS-Infrastrukturressourcen und listet sie im Container-Cluster-Tab auf. Sie können jedoch keine

Betriebsinformationen zu Ihren Amazon EKS-Pods oder -Containern in Application Manager verwalten. Sie können nur Betriebsinformationen zu der Infrastruktur verwalten und anzeigen, die Ihre Amazon EKS-Ressourcen hostet. Weitere Preisinformationen finden Sie unter Amazon ECS Preis und Amazon EKS Preis.

Einrichten von Aufgaben zum Anzeigen von Vorgangsinformationen zu Ressourcen

Die folgenden Setup-Aufgaben helfen Ihnen beim Anzeigen von Betriebsinformationen über Ihre AWS -Ressourcen in Application Manager.

(Empfohlen) Verifizieren Sie Runbook-Berechtigungen

Sie können Probleme mit AWS Ressourcen mithilfe Application Manager von Systems Manager Automation-Runbooks beheben. Um dieses Tool zu verwenden, müssen Sie Berechtigungen konfigurieren oder überprüfen. Preisinformationen finden Sie unter AWS Systems Manager – Preise.

(Optional) Aktivieren Sie Cost Explorer

AWS Cost Explorer ist eine Funktion AWS Cost Management , mit der Sie Ihre Kostendaten für weitere Analysen visualisieren können. Wenn Sie den Cost Explorer aktivieren, können Sie in der Application Manager-Konsole Kosteninformationen, den Kostenverlauf und die Kostenoptimierung für die Ressourcen Ihrer Anwendung einsehen.

(Optional) CloudWatch Amazon-Protokolle und -Alarme einrichten und konfigurieren

CloudWatch ist ein Überwachungs- und Verwaltungsservice, der Daten und umsetzbare Erkenntnisse für Hybrid AWS- und Multi-Cloud-Anwendungen und Infrastrukturressourcen bereitstellt. Mit CloudWatch können Sie all Ihre Leistungs- und Betriebsdaten in Form von Protokollen und Metriken von einer einzigen Plattform aus sammeln und darauf zugreifen. Um CloudWatch Protokolle und Alarme für Ihre Ressourcen anzeigen zu könnenApplication Manager, müssen Sie sie einrichten und konfigurieren CloudWatch. Preisinformationen finden Sie unter CloudWatch – Preise.

> **Note**
>
> CloudWatch Die Unterstützung von Protokollen gilt nur für Anwendungen, nicht für Cluster.

(Optional) AWS Config einrichten und konfigurieren

AWS Config bietet einen detaillierten Überblick über die Ressourcen, die mit Ihren verknüpft sind AWS-Konto, einschließlich ihrer Konfiguration, ihrer Beziehung zueinander und der Art und Weise, wie sich die Konfigurationen und ihre Beziehungen im Laufe der Zeit verändert haben. Sie können AWS Config es verwenden, um die Konfigurationseinstellungen Ihrer AWS Ressourcen auszuwerten. Dazu erstellen Sie AWS Config Regeln, die Ihre idealen Konfigurationseinstellungen darstellen. Es verfolgt AWS Config kontinuierlich die Konfigurationsänderungen, die in Ihren Ressourcen vorgenommen werden, und überprüft, ob diese Änderungen gegen eine der Bedingungen in Ihren Regeln verstoßen. Wenn eine Ressource gegen eine Regel verstößt, werden die AWS Config Ressource und die Regel als nicht konform gekennzeichnet. Application Managerzeigt Konformitätsinformationen zu Regeln an AWS Config . Um diese Daten anzeigen zu könnenApplication Manager, müssen Sie sie einrichten und konfigurieren AWS Config. Preisinformationen finden Sie unter AWS Config – Preise.

(Optional) Erstellen Sie State Manager Zuordnungen

Sie können Systems Manager State Manager verwenden, um eine Konfiguration zu erstellen, die Ihren verwalteten Knoten zugewiesen wird. Die Konfiguration, auch Zuordnung genannt, definiert den Zustand, den Sie auf Ihren Knoten beibehalten möchten. Um Zuordnungs-Compliance-Daten in Application Manager anzuzeigen, müssen Sie eine oder mehrere State Manager-Zuordnungen konfigurieren. State Manager wird ohne Aufpreis angeboten.

(Optional) Einrichten und Konfigurieren von OpsCenter

Sie können operative Arbeitselemente (OpsItems) über Ihre Ressourcen in Application Manager durch Verwendung von OpsCenter anzeigen. Sie können Amazon CloudWatch und Amazon so konfigurieren EventBridge , dass sie OpsCenter basierend OpsItems auf Alarmen und Ereignissen automatisch an senden. Sie können die Schlüssel auch manuell OpsItems eingeben. Preisinformationen finden Sie unter AWS Systems Manager – Preise.

Konfigurieren von Berechtigungen für Systems Manager Application Manager

Sie können alle Funktionen eines Tools in verwenden Application Manager AWS Systems Manager, wenn Ihre AWS Identity and Access Management (IAM-) Entität (z. B. ein Benutzer, eine Gruppe oder eine Rolle) Zugriff auf die in diesem Thema aufgeführten API-Operationen hat. Die API-Operationen sind in zwei Tabellen unterteilt, um Ihnen zu helfen, die verschiedenen Funktionen zu verstehen, die sie ausführen.

In der folgenden Tabelle sind die API-Vorgänge aufgeführt, die Systems Manager aufruft, wenn Sie eine Ressource in Application Manager wählen, da Sie die Ressourcendetails anzeigen möchten. Wenn beispielsweise eine Amazon EC2 Auto Scaling Scaling-Gruppe Application Manager aufgeführt ist und Sie diese Gruppe auswählen, um ihre Details anzuzeigen, ruft Systems Manager die `autoscaling:DescribeAutoScalingGroups` API-Operationen auf. Wenn Sie keine Auto Scaling Gruppen in Ihrem Konto haben, wird dieser API-Vorgang nicht von Application Manager ausgeführt.

Ausschließlich Ressourcendetails

```
acm:DescribeCertificate
acm:ListTagsForCertificate
autoscaling:DescribeAutoScalingGroups
cloudfront:GetDistribution
cloudfront:ListTagsForResource
cloudtrail:DescribeTrails
cloudtrail:ListTags
cloudtrail:LookupEvents
codebuild:BatchGetProjects
codepipeline:GetPipeline
codepipeline:ListTagsForResource
dynamodb:DescribeTable
dynamodb:ListTagsOfResource
ec2:DescribeAddresses
ec2:DescribeCustomerGateways
ec2:DescribeHosts
ec2:DescribeInternetGateways
ec2:DescribeNetworkAcls
ec2:DescribeNetworkInterfaces
ec2:DescribeRouteTables
ec2:DescribeSecurityGroups
ec2:DescribeSubnets
ec2:DescribeVolumes
ec2:DescribeVpcs
ec2:DescribeVpnConnections
ec2:DescribeVpnGateways
elasticbeanstalk:DescribeApplications
elasticbeanstalk:ListTagsForResource
elasticloadbalancing:DescribeInstanceHealth
elasticloadbalancing:DescribeListeners
elasticloadbalancing:DescribeLoadBalancers
```

Ausschließlich Ressourcendetails

```
elasticloadbalancing:DescribeTags
iam:GetGroup
iam:GetPolicy
iam:GetRole
iam:GetUser
lambda:GetFunction
rds:DescribeDBClusters
rds:DescribeDBInstances
rds:DescribeDBSecurityGroups
rds:DescribeDBSnapshots
rds:DescribeDBSubnetGroups
rds:DescribeEventSubscriptions
rds:ListTagsForResource
redshift:DescribeClusterParameters
redshift:DescribeClusterSecurityGroups
redshift:DescribeClusterSnapshots
redshift:DescribeClusterSubnetGroups
redshift:DescribeClusters
s3:GetBucketTagging
```

In der folgenden Tabelle sind die API-Vorgänge aufgeführt, die Systems Manager verwendet, um Änderungen an Anwendungen und Ressourcen vorzunehmen, die unter Application Manager gelistet sind oder um Vorgangsinformationen für eine ausgewählte Anwendung oder Ressource anzuzeigen.

Aktionen und Details der Anwendung

```
applicationinsights:CreateApplication
applicationinsights:DescribeApplication
applicationinsights:ListProblems
ce:GetCostAndUsage
ce:GetTags
ce:ListCostAllocationTags
ce:UpdateCostAllocationTagsStatus
cloudformation:CreateStack
cloudformation:DeleteStack
cloudformation:DescribeStackDriftDetectionStatus
cloudformation:DescribeStackEvents
cloudformation:DescribeStacks
```

Aktionen und Details der Anwendung

```
cloudformation:DetectStackDrift
cloudformation:GetTemplate
cloudformation:GetTemplateSummary
cloudformation:ListStacks
cloudformation:UpdateStack
cloudwatch:DescribeAlarms
cloudwatch:DescribeInsightRules
cloudwatch:DisableAlarmActions
cloudwatch:EnableAlarmActions
cloudwatch:GetMetricData
cloudwatch:ListTagsForResource
cloudwatch:PutMetricAlarm
config:DescribeComplianceByConfigRule
config:DescribeComplianceByResource
config:DescribeConfigRules
config:DescribeRemediationConfigurations
config:GetComplianceDetailsByConfigRule
config:GetComplianceDetailsByResource
config:GetResourceConfigHistory
config:ListDiscoveredResources
config:PutRemediationConfigurations
config:SelectResourceConfig
config:StartConfigRulesEvaluation
config:StartRemediationExecution
ec2:DescribeInstances
ecs:DescribeCapacityProviders
ecs:DescribeClusters
ecs:DescribeContainerInstances
ecs:ListClusters
ecs:ListContainerInstances
ecs:TagResource
eks:DescribeCluster
eks:DescribeFargateProfile
eks:DescribeNodegroup
eks:ListClusters
eks:ListFargateProfiles
eks:ListNodegroups
eks:TagResource
iam:CreateServiceLinkedRole
iam:ListRoles
logs:DescribeLogGroups
resource-groups:CreateGroup
```

Aktionen und Details der Anwendung

```
resource-groups:DeleteGroup
resource-groups:GetGroup
resource-groups:GetGroupQuery
resource-groups:GetTags
resource-groups:ListGroupResources
resource-groups:ListGroups
resource-groups:Tag
resource-groups:Untag
resource-groups:UpdateGroup
s3:ListAllMyBuckets
s3:ListBucket
s3:ListBucketVersions
servicecatalog:GetApplication
servicecatalog:ListApplications
sns:CreateTopic
sns:ListSubscriptionsByTopic
sns:ListTopics
sns:Subscribe
ssm:AddTagsToResource
ssm:CreateDocument
ssm:CreateOpsMetadata
ssm:DeleteDocument
ssm:DeleteOpsMetadata
ssm:DescribeAssociation
ssm:DescribeAutomationExecutions
ssm:DescribeDocument
ssm:DescribeDocumentPermission
ssm:GetDocument
ssm:GetInventory
ssm:GetOpsMetadata
ssm:GetOpsSummary
ssm:GetServiceSetting
ssm:ListAssociations
ssm:ListComplianceItems
ssm:ListDocuments
ssm:ListDocumentVersions
ssm:ListOpsMetadata
ssm:ListResourceComplianceSummaries
ssm:ListTagsForResource
ssm:ModifyDocumentPermission
ssm:RemoveTagsFromResource
ssm:StartAssociationsOnce
```

Aktionen und Details der Anwendung

```
ssm:StartAutomationExecution
ssm:UpdateDocument
ssm:UpdateDocumentDefaultVersion
ssm:UpdateOpsItem
ssm:UpdateOpsMetadata
ssm:UpdateServiceSetting
tag:GetTagKeys
tag:GetTagValues
tag:TagResources
tag:UntagResources
```

Beispielrichtlinie für alle Application Manager-Berechtigungen

Um Application Manager-Berechtigungen für eine IAM-Entität (z. B. einen Benutzer, eine Gruppe oder eine Rolle) zu konfigurieren, erstellen Sie anhand des folgenden Beispiels eine IAM-Richtlinie. Dieses Richtlinienbeispiel enthält alle API-Vorgänge, die von Application Manager verwendet werden.

JSON

```
                {
"Version":"2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "acm:DescribeCertificate",
            "acm:ListTagsForCertificate",
            "applicationinsights:CreateApplication",
            "applicationinsights:DescribeApplication",
            "applicationinsights:ListProblems",
            "autoscaling:DescribeAutoScalingGroups",
            "ce:GetCostAndUsage",
            "ce:GetTags",
            "ce:ListCostAllocationTags",
            "ce:UpdateCostAllocationTagsStatus",
            "cloudformation:CreateStack",
            "cloudformation:DeleteStack",
            "cloudformation:DescribeStackDriftDetectionStatus",
```

```
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStacks",
                "cloudformation:DetectStackDrift",
                "cloudformation:GetTemplate",
                "cloudformation:GetTemplateSummary",
                "cloudformation:ListStacks",
                "cloudformation:ListStackResources",
                "cloudformation:UpdateStack",
                "cloudfront:GetDistribution",
                "cloudfront:ListTagsForResource",
                "cloudtrail:DescribeTrails",
                "cloudtrail:ListTags",
                "cloudtrail:LookupEvents",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DescribeInsightRules",
                "cloudwatch:DisableAlarmActions",
                "cloudwatch:EnableAlarmActions",
                "cloudwatch:GetMetricData",
                "cloudwatch:ListTagsForResource",
                "cloudwatch:PutMetricAlarm",
                "codebuild:BatchGetProjects",
                "codepipeline:GetPipeline",
                "codepipeline:ListTagsForResource",
                "config:DescribeComplianceByConfigRule",
                "config:DescribeComplianceByResource",
                "config:DescribeConfigRules",
                "config:DescribeRemediationConfigurations",
                "config:GetComplianceDetailsByConfigRule",
                "config:GetComplianceDetailsByResource",
                "config:GetResourceConfigHistory",
                "config:ListDiscoveredResources",
                "config:PutRemediationConfigurations",
                "config:SelectResourceConfig",
                "config:StartConfigRulesEvaluation",
                "config:StartRemediationExecution",
                "dynamodb:DescribeTable",
                "dynamodb:ListTagsOfResource",
                "ec2:DescribeAddresses",
                "ec2:DescribeCustomerGateways",
                "ec2:DescribeHosts",
                "ec2:DescribeInstances",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeNetworkAcls",
                "ec2:DescribeNetworkInterfaces",
```

```
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVolumes",
                "ec2:DescribeVpcs",
                "ec2:DescribeVpnConnections",
                "ec2:DescribeVpnGateways",
                "ecs:DescribeCapacityProviders",
                "ecs:DescribeClusters",
                "ecs:DescribeContainerInstances",
                "ecs:ListClusters",
                "ecs:ListContainerInstances",
                "ecs:TagResource",
                "eks:DescribeCluster",
                "eks:DescribeFargateProfile",
                "eks:DescribeNodegroup",
                "eks:ListClusters",
                "eks:ListFargateProfiles",
                "eks:ListNodegroups",
                "eks:TagResource",
                "elasticbeanstalk:DescribeApplications",
                "elasticbeanstalk:ListTagsForResource",
                "elasticloadbalancing:DescribeInstanceHealth",
                "elasticloadbalancing:DescribeListeners",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeTags",
                "iam:CreateServiceLinkedRole",
                "iam:GetGroup",
                "iam:GetPolicy",
                "iam:GetRole",
                "iam:GetUser",
                "iam:ListRoles",
                "lambda:GetFunction",
                "logs:DescribeLogGroups",
                "rds:DescribeDBClusters",
                "rds:DescribeDBInstances",
                "rds:DescribeDBSecurityGroups",
                "rds:DescribeDBSnapshots",
                "rds:DescribeDBSubnetGroups",
                "rds:DescribeEventSubscriptions",
                "rds:ListTagsForResource",
                "redshift:DescribeClusterParameters",
                "redshift:DescribeClusters",
                "redshift:DescribeClusterSecurityGroups",
```

```
                "redshift:DescribeClusterSnapshots",
                "redshift:DescribeClusterSubnetGroups",
                "resource-groups:CreateGroup",
                "resource-groups:DeleteGroup",
                "resource-groups:GetGroup",
                "resource-groups:GetGroupQuery",
                "resource-groups:GetTags",
                "resource-groups:ListGroupResources",
                "resource-groups:ListGroups",
                "resource-groups:Tag",
                "resource-groups:Untag",
                "resource-groups:UpdateGroup",
                "s3:GetBucketTagging",
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:ListBucketVersions",
                "servicecatalog:GetApplication",
                "servicecatalog:ListApplications",
                "sns:CreateTopic",
                "sns:ListSubscriptionsByTopic",
                "sns:ListTopics",
                "sns:Subscribe",
                "ssm:AddTagsToResource",
                "ssm:CreateDocument",
                "ssm:CreateOpsMetadata",
                "ssm:DeleteDocument",
                "ssm:DeleteOpsMetadata",
                "ssm:DescribeAssociation",
                "ssm:DescribeAutomationExecutions",
                "ssm:DescribeDocument",
                "ssm:DescribeDocumentPermission",
                "ssm:GetDocument",
                "ssm:GetInventory",
                "ssm:GetOpsMetadata",
                "ssm:GetOpsSummary",
                "ssm:GetServiceSetting",
                "ssm:ListAssociations",
                "ssm:ListComplianceItems",
                "ssm:ListDocuments",
                "ssm:ListDocumentVersions",
                "ssm:ListOpsMetadata",
                "ssm:ListResourceComplianceSummaries",
                "ssm:ListTagsForResource",
                "ssm:ModifyDocumentPermission",
```

```
                "ssm:RemoveTagsFromResource",
                "ssm:StartAssociationsOnce",
                "ssm:StartAutomationExecution",
                "ssm:UpdateDocument",
                "ssm:UpdateDocumentDefaultVersion",
                "ssm:UpdateOpsMetadata",
                "ssm:UpdateOpsItem",
                "ssm:UpdateServiceSetting",
                "tag:GetResources",
                "tag:GetTagKeys",
                "tag:GetTagValues",
                "tag:TagResources",
                "tag:UntagResources"
            ],
            "Resource": "*"
        }
    ]
}
```

Note

Sie können die Fähigkeit eines Benutzers einschränken, Änderungen an Anwendungen und Ressourcen in Application Manager vorzunehmen, indem Sie die folgenden API-Operationen aus der IAM-Berechtigungsrichtlinie entfernen, die ihrem Benutzer, ihrer Gruppe oder ihrer Rolle zugeordnet ist. Durch entfernen dieser Aktionen, steht nur der read-only-Modus in Application Manager zur Verfügung. Im Folgenden sind alle Funktionen aufgeführt APIs , mit denen Benutzer Änderungen an der Anwendung oder anderen zugehörigen Ressourcen vornehmen können.

```
applicationinsights:CreateApplication
ce:UpdateCostAllocationTagsStatus
cloudformation:CreateStack
cloudformation:DeleteStack
cloudformation:UpdateStack
cloudwatch:DisableAlarmActions
cloudwatch:EnableAlarmActions
cloudwatch:PutMetricAlarm
config:PutRemediationConfigurations
config:StartConfigRulesEvaluation
config:StartRemediationExecution
```

```
ecs:TagResource
eks:TagResource
iam:CreateServiceLinkedRole
resource-groups:CreateGroup
resource-groups:DeleteGroup
resource-groups:Tag
resource-groups:Untag
resource-groups:UpdateGroup
sns:CreateTopic
sns:Subscribe
ssm:AddTagsToResource
ssm:CreateDocument
ssm:CreateOpsMetadata
ssm:DeleteDocument
ssm:DeleteOpsMetadata
ssm:ModifyDocumentPermission
ssm:RemoveTagsFromResource
ssm:StartAssociationsOnce
ssm:StartAutomationExecution
ssm:UpdateDocument
ssm:UpdateDocumentDefaultVersion
ssm:UpdateOpsMetadata
ssm:UpdateOpsItem
ssm:UpdateServiceSetting
tag:TagResources
tag:UntagResources
```

Informationen zum Erstellen und Bearbeiten von IAM-Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch. Informationen zum Zuweisen dieser Richtlinie zu einer IAM-Entität (z. B. einem Benutzer, einer Gruppe oder einer Rolle) finden Sie unter Hinzufügen und Entfernen von IAM-Identitätsberechtigungen.

Hinzufügen von Anwendungen und Container-Clustern zu Application Manager

Application Manager ist eine Komponente von AWS Systems Manager. In Application Manager ist eine Anwendung eine logische Gruppierung von AWS-Ressourcen, die Sie als Einheit betreiben möchten. Diese logische Gruppe kann verschiedene Versionen einer Anwendung, Besitzgrenzen für Operatoren oder Entwicklerumgebungen darstellen, um nur einige zu nennen.

Beim ersten Öffnen von Application Manager wird die Seite Was Application Manager für Sie tun kann angezeigt. Wenn Sie Erste Schritte auswählen, importiert Application Manager automatisch

Metadaten zu Ihren Ressourcen, die in anderen AWS-Services oder Systems-Manager-Tools erstellt wurden. Application Manager zeigt diese Ressourcen dann in einer Liste an, die nach vordefinierten Kategorien gruppiert ist.

Für Anwendungen umfasst die Liste Folgendes:

- AWS CloudFormation-Stacks
- Benutzerdefinierte Anwendungen
- AWS Launch Wizard-Anwendungen
- AppRegistry-Anwendungen
- AWS SAP Enterprise Workload-Anwendungen
- Amazon ECS-Cluster
- Amazon EKS-Cluster

Nach Abschluss des Imports können Sie Vorgangsinformationen für eine Anwendung oder eine bestimmte Ressource in diesen vordefinierten Kategorien anzeigen. Wenn Sie mehr Kontext zu einer Ressourcensammlung bereitstellen möchten, können Sie eine Anwendung manuell in Application Manager erstellen. Anschließend können Sie Ressourcen oder Ressourcengruppen zu dieser Anwendung hinzufügen. Nachdem Sie eine Anwendung in Application Manager erstellt haben, können Sie Betriebsinformationen zu Ihrer Ressource im Kontext einer Anwendung anzeigen.

Erstellen einer Anwendung in Application Manager

Gehen Sie wie folgt vor, um eine Anwendung in Application Manager zu erstellen und um Ressourcen zu dieser Anwendung hinzuzufügen.

So erstellen Sie eine Anwendung in Application Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie Anwendung erstellen aus.
4. Wählen Sie eine Option aus der Drop-down-Liste aus und füllen Sie die Felder auf der daraufhin geöffneten Seite aus.

Arbeiten mit -Anwendungen

Application Managerist ein Bestandteil von AWS Systems Manager. Dieser Abschnitt enthält Themen, die Ihnen bei der Verwendung vonApplication Manager-Anwendungen helfen und helfen, Betriebsinformationen über Ihre AWS -Ressourcen einzusehen.

Inhalt

- Anwendungsübersicht in Application Manager
- Verwaltung Ihrer EC2 Anwendungsinstanzen
- Ressourcen, die mit Ihrer Anwendung verknüpft sind
- Verwaltung der Konformität Ihrer Anwendungen
- Verwenden von CloudWatch Application Insights zur Überwachung einer Anwendung
- Anzeigen von OpsItems für eine Anwendung
- Verwalten Ihrer Anwendungsprotokolle
- Automation-Runbooks verwenden, um Anwendungsprobleme zu beheben
- Taggen Sie Ressourcen in Application Manager
- Arbeiten mit CloudFormation-Vorlagen und Stacks in Application Manager
- Arbeiten mit Clustern in Application Manager

Anwendungsübersicht in Application Manager

In Application Manager, eine Komponente von AWS Systems Manager, zeigt der Reiter Übersicht eine Zusammenfassung der Amazon CloudWatch Alarme, betriebliche Arbeitselemente (OpsItems), CloudWatch Application Insights und Runbook-Verlauf. Klicken Sie auf Alle anzeigen (View all) für jede Karte, um die entsprechende Registerkarte zu öffnen, auf der Sie alle Anwendungseinblicke, Alarme, OpsItems, oder den Runbook-Verlauf einsehen können.

Informationen zu Application Insights

CloudWatch Application Insights identifiziert Schlüsselmetriken, Protokolle und Alarme und richtet diese für Ihre Anwendungsressourcen und Ihren Technologie-Stack ein. Application Insights überwacht kontinuierlich Metriken und Protokolle, um Anomalien und Fehler zu erkennen und zu korrelieren. Wenn das System Fehler oder Anomalien erkennt, generiert Application Insights CloudWatch Events, mit denen Sie Benachrichtigungen einrichten oder Aktionen ausführen

können. Wenn Sie die Schaltfläche Konfiguration bearbeiten auf der Registerkarte Überwachung wählen, öffnet das System die CloudWatch Application Insights-Konsole. Weitere Informationen zu Application Insights finden Sie unter Was ist Amazon CloudWatch Application Insights im Amazon CloudWatch-Benutzerhandbuch.

Über Kosten-Explorer

Application Manager ist über das Cost-Widget und die Registerkarte Cost mit AWS Cost Explorer, einem Feature von AWS Cost Management, integriert. Nachdem Sie den Cost Explorer in der Kostenmanagement-Konsole aktiviert haben, zeigt das Cost-Widget und die Registerkarte Cost in Application Manager die Kostendaten für eine bestimmte Nicht-Container-Anwendung oder Anwendungskomponente an. Sie können Filter im Widget oder der Registerkarte verwenden, um Preisdaten nach verschiedenen Zeiträumen, Details und Preisarten in einem Balken- oder Liniendiagramm anzuzeigen.

Sie können dieses Feature aktivieren, indem Sie die Schaltfläche Go to AWS Cost Management console wählen. Standardmäßig werden die Daten auf die letzten drei Monate gefiltert. Wenn Sie für eine Nicht-Container-Anwendung die Schaltfläche View all (Alle anzeigen) wählen, öffnet Application Manager die Registerkarte Resources (Ressourcen). Für Container-Anwendungen öffnet die Schaltfläche View all (Alle anzeigen) die AWS Cost Explorer-Konsole.

Aktionen, die Sie auf dieser Seite ausführen können

Auf der Registerkarte Overview (Übersicht) auf dieser Seite können Sie Informationen zu den folgenden Widgets aktivieren und abrufen. Wenn ein Widget aktiviert ist, wählen Sie dessen View all (Alle anzeigen) aus, um relevante Anwendungsdetails für diesen Bereich anzuzeigen.

- Wählen Sie im Abschnitt Insights and Alarms (Erkenntnisse und Alarme) die Zahl für einen Schweregrad aus, um die Registerkarte Monitoring (Überwachung) zu öffnen, auf der Sie weitere Details zu Alarmen des ausgewählten Schweregrads anzeigen können.
- Wählen Sie im Abschnitt Cost (Kosten) die Option View all (Alle anzeigen) aus, um die Registerkarte Resources (Ressourcen) zu öffnen, auf der Sie Kostendaten für eine bestimmte Anwendung oder Anwendungskomponente anzeigen können.
- Wählen Sie im Abschnitt Compliance die Option View all (Alle anzeigen) aus, um die Registerkarte Compliance zu öffnen, auf der Sie Compliance-Informationen von AWS Config und State Manager-Zuordnungen anzeigen können.

> **Note**
>
> Um Patch-Compliance-Details anzuzeigen, wählen Sie direkt die Registerkarte Compliance aus. Anschließend können Sie Patch-Compliance-Details für die verwalteten Knoten anzeigen, die von der ausgewählten Anwendung verwendet werden.

- Wählen Sie in der Sektion Runbooks ein Runbook aus, um es auf der Seite Dokumente des Systems Managers, auf der Sie weitere Details zum Dokument anzeigen können, zu öffnen.
- Wählen Sie in der Sektion OpsItems einen Schweregrad aus, um die Registerkarte OpsItems zu öffnen, auf der Sie alle OpsItems des gewählten Schweregrads einsehen können.
- Wählen Sie eine Alle anzeigen-Schaltfläche, um die entsprechende Registerkarte zu öffnen. Sie können alle Alarm-, OpsItems oder Runbook-Verlaufeinträge für die Anwendung anzeigen.

So öffnen Sie die Registerkarte Overview (Übersicht)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.

Verwaltung Ihrer EC2 Anwendungsinstanzen

Application Managerintegriert sich in Amazon Elastic Compute Cloud (Amazon EC2), um Informationen über Ihre Instances im Kontext einer Anwendung anzuzeigen. Application Managerzeigt den Instance-Status, den Status und den Zustand von Amazon EC2 Auto Scaling für eine ausgewählte Anwendung in einem grafischen Format an. Die Registerkarte Instances enthält auch eine Tabelle mit den folgenden Informationen für jede Instance in Ihrer Anwendung:

- Instance-Status (Ausstehend, Angehalten, Wird ausgeführt, Beendet)
- Ping-Status für SSM Agent

- Status und Name des letzten Systems-Manager-Automation-Runbooks, das auf der Instance verarbeitet wurde
- Eine Anzahl von Amazon CloudWatch Logs-Alarmen pro Bundesstaat.
  - `ALARM` – Die Metrik oder der Ausdruck liegt außerhalb des festgelegten Schwellenwerts.
  - `OK` – Die Metrik oder der Ausdruck liegt innerhalb des festgelegten Schwellenwerts.
  - `INSUFFICIENT_DATA` – Der Alarm wurde soeben gestartet; die Metrik ist nicht verfügbar oder es sind nicht genügend Daten verfügbar, damit die Metrik den Alarmstatus bestimmen kann.
- Zustand der Auto-Scaling-Gruppe für die übergeordneten und einzelnen Auto-Scaling-Gruppen

Wenn Sie eine Instance in der Tabelle All instances (Alle Instances) auswählen, zeigt Application Manager Informationen zu dieser Instance auf vier Registerkarten an:

- Details — Alle Instance-Details von Amazon EC2, einschließlich Amazon Machine Image (AMI), DNS-Informationen, IP-Adressinformationen und mehr.
- Health — Der aktuelle Status, wie er durch EC2 System- und Instanzstatusprüfungen bereitgestellt wird.
- Execution history (Ausführungshistorie) – Ausführungsprotokolle für Systems-Manager-Automation-Runbooks und API-Aufrufe, die von der Instance verarbeitet werden.
- CloudWatch Alarme — Name, Status und mehr für alle von der Instance ausgelösten CloudWatch Alarme.

Aktionen, die Sie auf dieser Seite ausführen können

Hier sind folgende Aktionen möglich:

- Instances starten, anhalten und beenden.
- Wenden Sie ein Chef-Rezept an.
- Fügen Sie Instances einer Auto-Scaling-Gruppe hinzu oder trennen Sie Instances von einer Auto-Scaling-Gruppe.
- Aktivieren Sie automatische Updates für SSM Agent.

So öffnen Sie die Registerkarte Instances

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Custom applications (Benutzerdefinierte Anwendungen).
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Registerkarte Instances aus.

So zeigen Sie die Details Ihrer Anwendungs-Instances an

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Custom applications (Benutzerdefinierte Anwendungen).
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Registerkarte Instances aus.
6. Wählen Sie die Schaltfläche neben der Instance aus, deren Details Sie anzeigen möchten.
7. Überprüfen Sie die Instance-Details unten auf der Seite.

So aktualisieren Sie SSM Agent automatisch

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Custom applications (Benutzerdefinierte Anwendungen).
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Registerkarte Instances aus.

6. Wählen Sie in der Dropdownliste Agentenaktionen die Option SSM Agent-Update konfigurieren aus.
7. Wählen Sie Alle Instances aus, um automatische SSM Agent-Updates für alle verwalteten Instances zu konfigurieren. Wählen Sie alternativ Instance, um automatische SSM Agent-Updates für eine einzelne Instance in Ihrer Anwendung zu konfigurieren.
8. Wählen Sie den Schalter Automatische Updates aktivieren aus.
9. Wählen Sie in der Dropdownliste Zeitplan angeben den Zeitplan aus, den Sie für SSM Agent-Updates verwenden möchten.
10. Wählen Sie Konfigurieren.

Ressourcen, die mit Ihrer Anwendung verknüpft sind

In Application Manager, eine Komponente von AWS Systems Manager, zeigt die Registerkarte Ressourcen die AWS-Ressourcen in Ihrer Anwendung an. Wenn Sie eine Komponente der obersten Ebene auswählen, werden auf dieser Seite alle Ressourcen für diese Komponente und alle Unterkomponenten angezeigt. Wenn Sie eine Unterkomponente auswählen, werden auf dieser Seite nur die Ressourcen angezeigt, die dieser Unterkomponente zugewiesen sind.

Aktionen, die Sie auf dieser Seite ausführen können

Hier sind folgende Aktionen möglich:

- Wählen Sie einen Ressourcennamen aus, um Informationen darüber anzuzeigen, einschließlich Details, die von der Konsole bereitgestellt wurden, auf der die Ressource erstellt wurde, Tags, Amazon CloudWatch Alarme, AWS Config-Details und AWS CloudTrail-Protokollinformationen.
- Wählen Sie die Optionsschaltfläche neben einem Ressourcennamen. Wählen Sie dann die Schaltfläche Zeitplan der Ressource, um die AWS Config-Konsole zu öffnen, in der Sie Compliance-Informationen zu einer ausgewählten Ressource anzeigen können.
- Wenn Sie AWS Cost Explorer aktiviert haben, zeigt der Abschnitt Cost Explorer Preisdaten für eine bestimmte Nicht-Container-Anwendung oder -Anwendungs-Komponente an. Sie können diese Funktion aktivieren, indem Sie die Schaltfläche Go to AWS Cost Management console wählen. Verwenden Sie die Filter in diesem Abschnitt, um Preisinformationen zu Ihrer Anwendung anzuzeigen.

So öffnen Sie die Registerkarte Resources (Ressourcen)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Registerkarte Resources (Ressourcen) aus.

Verwaltung der Konformität Ihrer Anwendungen

In Application Manager, eine Komponente von AWS Systems Manager, zeigt die Seite Configurations (Konfigurationen) Informationen zur Compliance von Regeln für AWS Config-Ressourcen und -Konfigurationen an. Auf dieser Seite werden auch AWS Systems Manager State Manager Compliance-Informationen angezeigt. Sie können eine Ressource, eine Regel oder eine Zuordnung auswählen, um die entsprechende Konsole für weitere Informationen zu öffnen. Auf dieser Seite werden die Compliance-Informationen der letzten 90 Tage angezeigt.

Aktionen, die Sie auf dieser Seite ausführen können

Hier sind folgende Aktionen möglich:

- Wählen Sie einen Ressourcennamen, um die AWS Config-Konsole zu öffnen, in der Sie Compliance-Informationen zu einer ausgewählten Ressource anzeigen können.
- Wählen Sie die Optionsschaltfläche neben einem Ressourcennamen. Wählen Sie dann die Schaltfläche Zeitplan der Ressource, um die AWS Config-Konsole zu öffnen, in der Sie Compliance-Informationen zu einer ausgewählten Ressource anzeigen können.
- In der Sektion Compliance-Regeln können Sie zudem Folgendes durchführen:
  - Wählen Sie einen Namen aus, um die AWS Config-Konsole zu wählen, auf der Sie Informationen über diese Regel anzeigen können.
  - Wählen Sie Hinzufügen von Regeln, um die AWS Config-Konsole zu öffnen, in der Sie eine Regel erstellen können.

  - Wählen Sie die Optionsschaltfläche neben einem Regelnamen, wählen Sie Aktionen und wählen Sie dann Verwalten der Behebung, um die Behebungsaktion für eine Regel zu ändern.
  - Wählen Sie die Optionsschaltfläche neben einem Regelnamen, wählen Sie Aktionen und wählen Sie dann Erneut bewerten, um AWS Config eine Compliance-Überprüfung für die gewählte Regel auszuführen.
- In der Sektion Association compliance können Sie zudem Folgendes durchführen:
  - Wählen Sie einen Zuordnungsnamen aus, um die Seite Associations zu öffnen, wo Sie Informationen über diese Assoziation einsehen können.
  - Wählen Sie Erstellen einer Zuordnung, um Systems Manager zu öffnenState Manager, in welchem Sie eine Zuordnung erstellen können.
  - Wählen Sie die Optionsschaltfläche neben einem Assoziationsnamen und wählen Sie Zuordnung anwenden, um alle in der Zuordnung angegebenen Aktionen sofort zu starten.

So öffnen Sie die Compliance-Registerkarte

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Compliance-Registerkarte.

Verwenden von CloudWatch Application Insights zur Überwachung einer Anwendung

InApplication Manager, einer Komponente von AWS Systems Manager, werden auf der Registerkarte Überwachung Amazon CloudWatch Application Insights und Alarmdetails für Ressourcen in einer Anwendung angezeigt.

Informationen zu Application Insights

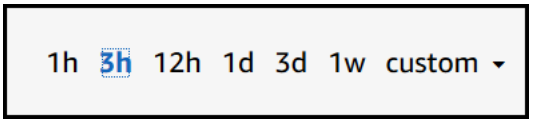
CloudWatch Application Insights identifiziert und richtet wichtige Kennzahlen, Protokolle und Alarme für Ihre Anwendungsressourcen und Ihren Technologie-Stack ein. Application Insights überwacht kontinuierlich Metriken und Protokolle, um Anomalien und Fehler zu erkennen und zu korrelieren.

Wenn das System Fehler oder Anomalien erkennt, generiert Application Insights CloudWatch Ereignisse, anhand derer Sie Benachrichtigungen einrichten oder Maßnahmen ergreifen können. Wenn Sie auf der Registerkarte Überwachung auf die Schaltfläche Konfiguration bearbeiten klicken, öffnet das System die CloudWatch Application Insights-Konsole. Weitere Informationen zu Application Insights finden Sie unter Was ist Amazon CloudWatch Application Insights im CloudWatch Amazon-Benutzerhandbuch.

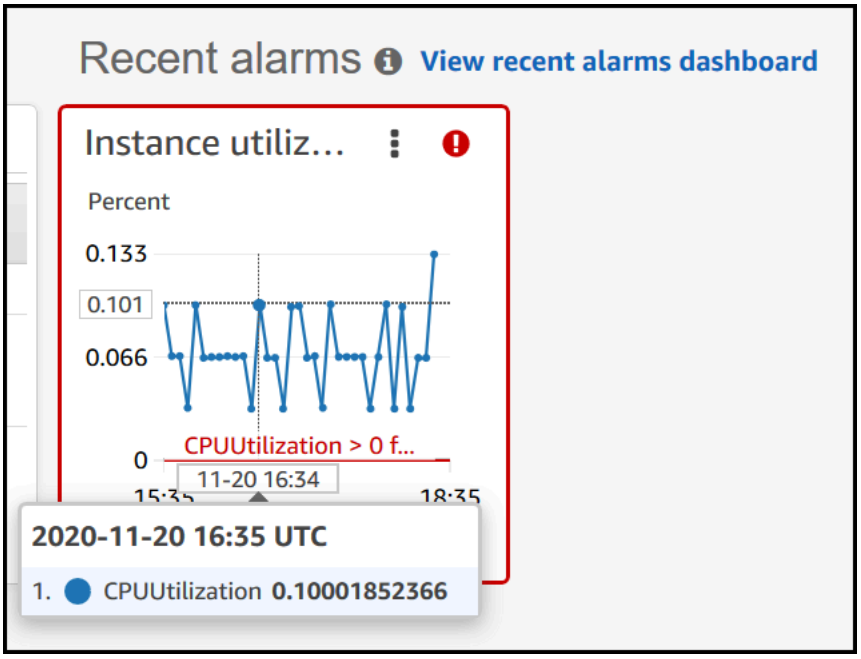
Aktionen, die Sie auf dieser Seite ausführen können

Hier sind folgende Aktionen möglich:

- Wählen Sie im Bereich Alarme nach Service einen AWS Servicenamen aus, CloudWatch um den ausgewählten Service und Alarm zu öffnen.
- Passen Sie den Zeitraum für Daten an, die in Widgets in der Sektion Aktuelle Alarme angezeigt werden, indem Sie einen der vordefinierten Zeitperiodenwerte auswählen. Sie können benutzerdefiniert wählen, um Ihren eigenen Zeitraum zu definieren.

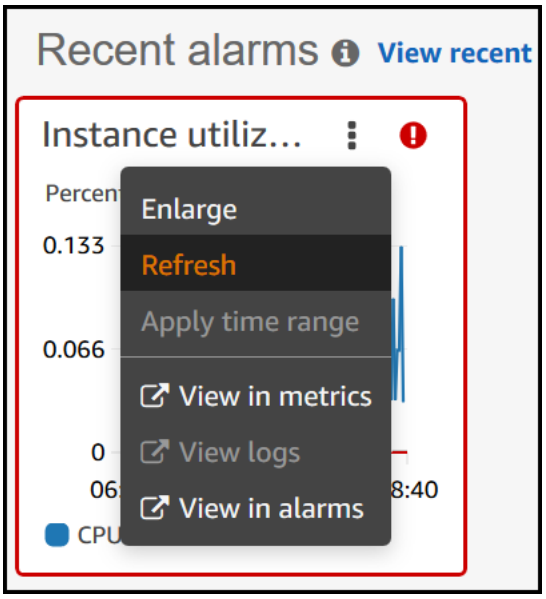


- Bewegen Sie den Cursor über ein Widget in der Sektion Aktuelle Alarme, um ein Datenpop-up für eine bestimmte Zeit anzuzeigen.



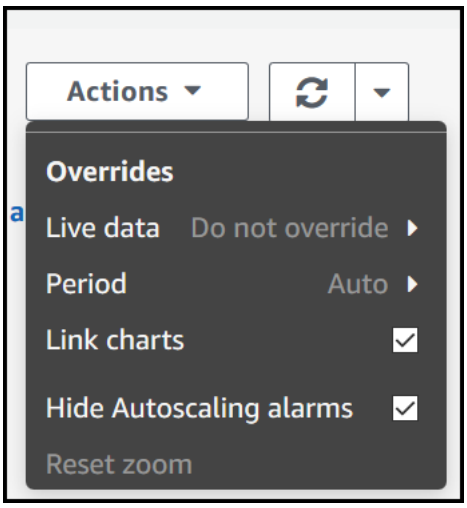
- Wählen Sie das Optionsmenü in einem Widget, um Anzeigeoptionen anzuzeigen. Klicken Sie auf Vergrößern, um ein Widget zu erweitern. Klicken Sie auf Aktualisieren, um die Daten in einem

Widget zu aktualisieren. Klicken und ziehen Sie den Cursor in einer Widget-Datenanzeige, um einen bestimmten Bereich auszuwählen. Sie können dann Zeitrahmen auswählen wählen.



- Wählen Sie das Menü Aktionen, um Optionen zum Überschreiben von Alarmdaten anzuzeigen. Folgende Optionen sind verfügbar:
  - Wählen Sie, ob Ihr Widget Live-Daten anzeigt. Live-Daten sind Daten, die innerhalb der letzten Minute veröffentlicht und noch nicht vollständig aggregiert wurden. Wenn Live-Daten deaktiviert sind, werden nur Datenpunkte mit einem Aggregationszeitraum von mindestens einer Minute in der Vergangenheit angezeigt. Bei Verwendung von 5-Minuten-Zeiträumen wird der Datenpunkt für 12:35 von 12:35 zu 12:40 aggregiert und um 12:41 angezeigt.

    Wenn Live-Daten aktiviert sind, wird der neueste Datenpunkt angezeigt, sobald Daten im entsprechenden Aggregationsintervall veröffentlicht werden. Bei jeder Aktualisierung der Anzeige, ändert sich der aktuellste Datenpunkt möglicherweise, wenn neue Daten innerhalb dieses Aggregationszeitraums veröffentlicht werden.
  - Geben Sie einen Zeitraum für Live-Daten an.
  - Verknüpfen Sie die Diagramme in der Sektion Aktuelle Alarme, sodass, wenn Sie ein Diagramm vergrößern oder verkleinern, das andere Diagramm gleichzeitig vergrößert oder verkleinert wird. Sie können die Verknüpfung mit Diagrammen aufheben, um den Zoom auf ein Diagramm zu beschränken.
  - Auto Scaling-Alarme ausblenden.



So öffnen Sie die Registerkarte Monitoring (Überwachung)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Registerkarte Überwachung.

Anzeigen von OpsItems für eine Anwendung

In Application Manager, eine Komponente vonAWS Systems Manager, zeigt die Registerkarte OpsItems operationelle Arbeitselemente (OpsItems) für Ressourcen in der ausgewählten Anwendung an. Sie können Systems Manager OpsCenter so konfigurieren, dass automatisch OpsItems aus Amazon CloudWatch Alarmen und Amazon EventBridge-Ereignisse erstellt werden. Sie können auch manuell OpsItems erstellen.

Aktionen, die Sie auf dieser Registerkarte ausführen können

Hier sind folgende Aktionen möglich:

- Filtern Sie die Liste von OpsItems durch Verwendung des Suchfelds. Sie können nach OpsItem-Name, ID, Quell-ID oder Schweregrad filtern. Sie können die Liste auch basierend auf dem Status filtern. OpsItems unterstützt die folgenden Status: „Offen“, „In Bearbeitung“, „Öffnen und In Bearbeitung“, „Abgeschlossen“ oder „Alle“.
- Ändern des Status einer OpsItem, indem Sie die Optionsschaltfläche daneben auswählen und dann eine Option im Menü Status einstellen wählen.
- Öffnen Sie Systems Manager OpsCenter, um eine OpsItem zu erstellen, indem Sie Create OpsItem wählen.

Um die Registerkarte OpsItems zu öffnen

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Registerkarte OpsItems aus.

Verwalten Ihrer Anwendungsprotokolle

In Application Manager, eine Komponente vonAWS Systems Manager, zeigt die Registerkarte Protokolle eine Liste der Protokollgruppen aus Amazon CloudWatch Logs an.

Aktionen, die Sie auf dieser Registerkarte ausführen können

Hier sind folgende Aktionen möglich:

- Wählen Sie einen Protokollgruppennamen aus, um ihn in CloudWatch Logs zu öffnen. Sie können dann einen Protokolldatenstrom auswählen, um Protokolle für eine Ressource im Kontext einer Anwendung anzuzeigen.
- Klicken Sie auf Erstellen von Protokollgruppen, um eine Protokollgruppe in CloudWatch Logs zu erstellen.

So öffnen Sie die Registerkarte Logs (Protokolle)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Registerkarte Protokolle aus.

Automation-Runbooks verwenden, um Anwendungsprobleme zu beheben

Sie können Probleme mit AWS-Ressourcen von Application Manager (einem Tool in AWS Systems Manager) mithilfe von Automation Runbooks beheben. Ein Automatisierungs-Runbook definiert die Aktionen, die Systems Manager auf Ihren verwalteten Instances und anderen AWS-Ressourcen ausführt, wenn eine Automatisierung ausgeführt wird. Automation ist ein Tool in AWS Systems Manager. Ein Runbook enthält einen oder mehrere Schritte, die in sequenzieller Reihenfolge ausgeführt werden. Jeder Schritt basiert auf einer einzigen Aktion. Die Ausgabe von einem Schritt kann als Eingabe in einem späteren Schritt verwendet werden.

Wenn Sie Start runbook (Starten von Runbook) von einer Application Manager-Anwendung oder -Cluster wählen, zeigt das System eine gefilterte Liste von verfügbaren Runbooks basierend auf dem Typ der Ressourcen in Ihrer Anwendung oder Ihrem Cluster an. Wenn Sie das Runbook auswählen, das Sie starten möchten, öffnet Systems Manager die Seite Ausführen des Automatisierungsdokuments.

Application Manager umfasst die folgenden Verbesserungen für die Arbeit mit Runbooks.

- Wenn Sie den Namen einer Ressource in Application Manager wählen und dann Runbook ausführen wählen, zeigt das System eine gefilterte Liste von Runbooks für diesen Ressourcentyp an.
- Sie können eine Automatisierung für alle Ressourcen desselben Typs initiieren, indem Sie ein Runbook in der Liste auswählen und dann Für Ressourcen desselben Typs ausführen wählen.

Bevor Sie beginnen

Bevor Sie ein Runbook von Application Managerstarten, gehen Sie wie folgt vor:

- Stellen Sie sicher, dass Sie über die richtigen Berechtigungen zum Starten von Runbooks verfügen. Weitere Informationen finden Sie unter Einrichten der Automatisierung.
- Lesen Sie die Dokumentation zur Automatisierungsprozedur zum Starten von Runbooks. Weitere Informationen finden Sie unter Führen Sie einen automatisierten Vorgang aus, der von Systems Manager Automation unterstützt wird.

So starten Sie ein Runbook aus Application Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie Runbook starten. Application Manager öffnet das Popup-Fenster Automatisierungs-Widget. Informationen zu den Optionen im Automatisierungs-Widget finden Sie unter Führen Sie einen automatisierten Vorgang aus, der von Systems Manager Automation unterstützt wird.

Taggen Sie Ressourcen in Application Manager

Sie können Tags schnell in Anwendungen und AWS-Ressourcen in Application Manager löschen oder hinzufügen. Gehen Sie folgendermaßen vor, um ein Tag zu einer Anwendung hinzuzufügen oder aus einer Anwendung und alle AWS-Ressourcen in dieser Anwendung.

Um ein Tag in einer Anwendung und allen Ressourcen in der Anwendung hinzuzufügen oder zu löschen

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.

3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. In der Sektion Anwendungsinformation wählen Sie die Zahl unter Anwendungstags. Wenn der Anwendung keine Tags zugewiesen sind, ist die Zahl Null.
6. Um einen Tag hinzuzufügen, wählen Sie Add new tag (Neuen Tag hinzufügen). Geben Sie einen Schlüssel und einen optionalen Wert ein. Zum Entfernen eines Tags wählen Sie Remove (Entfernen).
7. Wählen Sie Speichern.

Gehen Sie wie folgt vor, um ein Tag einer bestimmten Ressource in Application Manager hinzuzufügen oder zu entfernen.

So fügen Sie ein Tag zu einer Ressource hinzu oder löschen es aus

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie eine Kategorie in der Sektion Anwendungen aus. Wenn Sie eine Anwendung öffnen möchten, die Sie manuell in Application Manager erstellt haben, wählen Sie Benutzerdefinierte Anwendungen aus.
4. Wählen Sie die Anwendung in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.
5. Wählen Sie die Registerkarte Resources (Ressourcen) aus.
6. Wählen Sie einen Ressourcennamen.
7. Klicken Sie im Abschnitt Tags auf Edit (Bearbeiten).
8. Um einen Tag hinzuzufügen, wählen Sie Add new tag (Neuen Tag hinzufügen). Geben Sie einen Schlüssel und einen optionalen Wert ein. Zum Entfernen eines Tags wählen Sie Remove (Entfernen).
9. Wählen Sie Speichern.

Arbeiten mit CloudFormation-Vorlagen und Stacks in Application Manager

Application Manager, ein Tool in AWS Systems Manager, unterstützt Sie bei der Bereitstellung und Verwaltung von Ressourcen für Ihre Anwendungen durch die Integration in AWS CloudFormation. Sie können CloudFormation-Vorlagen und -Stacks in Application Manager erstellen, bearbeiten und löschen. Bei einem Stack handelt es sich um eine Sammlung von AWS-Ressourcen, die Sie als einzelne Einheit verwalten können. Dies bedeutet, dass Sie eine Sammlung von AWS-Ressourcen mithilfe von CloudFormation Stacks erstellen, aktualisieren oder löschen können. Eine Vorlage ist eine formatierte Textdatei in JSON oder YAML, die die Ressourcen angibt, die Sie in Ihren Stacks bereitstellen möchten.

Application Manager enthält auch eine Vorlagenbibliothek, in der Sie Vorlagen klonen, erstellen und speichern können. Application Manager und CloudFormation zeigen dieselben Informationen über den aktuellen Status eines Stacks an. Vorlagen und Vorlagenaktualisierungen werden in Systems Manager gespeichert, bis Sie den Stapel bereitstellen. Zu diesem Zeitpunkt werden die Änderungen auch in CloudFormation angezeigt.

Nachdem Sie einen Stapel in Application Manager erstellt haben, zeigt die Seite CloudFormation-Stacks hilfreiche Informationen dazu an. Dazu gehört die Vorlage, die zum Erstellen verwendet wurde, eine Anzahl von OpsItems für Ressourcen in Ihrem Stack, der Stack-Status undDrift-Status.

Über Kosten-Explorer

Application Manager ist über das Cost-Widget in AWS Cost Explorer, ein Feature von AWS Cost Management, integriert. Nachdem Sie den Cost Explorer in der Kostenmanagement-Konsole aktiviert haben, zeigt das Cost-Widget in Application Manager Preisdaten für eine bestimmte Anwendung oder Anwendungskomponente ohne Container an. Sie können Filter im Widget verwenden, um Preisdaten nach verschiedenen Zeiträumen, Details und Preisarten in einem Balken- oder Liniendiagramm anzuzeigen.

Sie können dieses Feature aktivieren, indem Sie die Schaltfläche Go to AWS Cost Management console wählen. Standardmäßig werden die Daten auf die letzten drei Monate gefiltert. Wenn Sie für eine Nicht-Container-Anwendung die Schaltfläche View all (Alle anzeigen) wählen, öffnet Application Manager die Registerkarte Resources (Ressourcen). Für Container-Anwendungen öffnet die Schaltfläche View all (Alle anzeigen) die AWS Cost Explorer-Konsole.

> **Note**
>
> Cost Explorer verwendet Tags, um Ihre Anwendungskosten zu verfolgen. Wenn Ihre CloudFormation-Stack-basierte Anwendung nicht mit dem `AppManagerCFNStackKey-`

Tag-Schlüssel konfiguriert ist, kann Cost Explorer keine genauen Kostendaten in Application Manager anzeigen. Wenn der `AppManagerCFNStackKey`-Tag-Schlüssel nicht erkannt wird, werden Sie in der Konsole aufgefordert, das Tag zu Ihrem CloudFormation-Stack hinzuzufügen, um die Kostenverfolgung zu aktivieren. Durch das Hinzufügen wird der Tag-Schlüssel dem Amazon-Ressourcennamen (ARN) Ihres Stacks zugeordnet und das Cost-Widget kann genaue Kostendaten anzeigen.

Important

Das Hinzufügen des `AppManagerCFNStackKey`-Tags löst ein Stack-Update aus. Alle manuellen Konfigurationen, die nach der ursprünglichen Bereitstellung des Stacks vorgenommen wurden, werden nach dem Hinzufügen des Benutzer-Tags nicht mehr berücksichtigt. Weitere Informationen über das Aktualisierungsverhalten von Ressourcen finden Sie unter Aktualisierungsverhalten von Stack-Ressourcen im AWS CloudFormation-Benutzerhandbuch

Bevor Sie beginnen

Verwenden Sie die folgenden Links, um mehr über CloudFormation Konzepte zu erfahren, bevor Sie CloudFormation-Vorlagen und -Stacks mithilfe von Application Manager erstellen bearbeiten oder löschen.

- Was ist AWS CloudFormation?
- Bewährte Methoden für CloudFormation
- Lernen der Grundlagen von Vorlagen
- Arbeiten mit AWS CloudFormation-Stacks
- Arbeiten mit AWS CloudFormation-Vorlagen
- Mustervorlagen

Themen

- Verwendung von Application Manager zur Verwaltung von CloudFormation-Vorlagen
- Verwenden von Application Manager zur Verwaltung von CloudFormation-Stacks

Verwendung von Application Manager zur Verwaltung von CloudFormation-Vorlagen

Application Manager, ein Tool in AWS Systems Manager, enthält eine Vorlagenbibliothek und andere Tools, mit denen Sie AWS CloudFormation-Vorlagen verwalten können. Dieser Abschnitt enthält folgende Informationen.

Themen

- Arbeiten mit der Vorlagenbibliothek
- Erstellung von Vorlagen
- Bearbeiten einer Vorlage

Arbeiten mit der Vorlagenbibliothek

Die Application Manager-Vorlagenbibliothek bietet Tools, mit denen Sie Vorlagen anzeigen, erstellen, bearbeiten, löschen und klonen können. Sie können Stacks auch direkt aus der Vorlagenbibliothek bereitstellen. Die Vorlagen werden als Systems Manager (SSM) -Dokumente vom Typ `CloudFormation` gespeichert. Wenn Sie Vorlagen als SSM-Dokumente speichern, können Sie Versionskontrollen verwenden, um mit verschiedenen Versionen einer Vorlage zu arbeiten. Sie können auch Berechtigungen festlegen und Vorlagen teilen. Nachdem Sie einen Stack erfolgreich bereitgestellt haben, sind der Stack und die Vorlage in Application Manager und CloudFormation verfügbar.

Bevor Sie beginnen

Es wird empfohlen, die folgenden Themen zu lesen, um mehr über SSM-Dokumente zu erfahren, bevor Sie mit dem Arbeiten mit CloudFormation-Vorlagen in Application Manager beginnen.

- AWS Systems Manager-Documents
- Freigeben von SSM-Dokumenten
- Bewährte Methoden für freigegebene SSM-Dokumente

So zeigen Sie die Vorlagenbibliothek in Application Manager an

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie CloudFormation-Vorlagenbibliothek aus.

Erstellung von Vorlagen

Im folgenden Verfahren wird beschrieben, wie Sie eine CloudFormation-Vorlage in Application Manager erstellen. Wenn Sie eine Vorlage erstellen, geben Sie die Stackdetails der Vorlage entweder in JSON oder YAML ein. Wenn Sie noch keine Erfahrungen mit JSON oder YAML haben, können Sie AWS Infrastructure Composer, ein Tool zum visuellen Erstellen und Ändern von Vorlagen, verwenden. Weitere Informationen finden Sie im Benutzerhandbuch für AWS CloudFormation unter Visuelles Erstellen von Vorlagen mit Infrastructure Composer. Informationen zur Struktur und Syntax einer Vorlage finden Sie im Benutzerhandbuch für AWS CloudFormation unter Abschnitte einer CloudFormation-Vorlage.

Sie können eine Vorlage auch aus mehreren Vorlagenausschnitten erstellen. Vorlagenausschnitte sind Beispiele, die zeigen, wie Vorlagen für eine bestimmte Ressource geschrieben werden. Sie können z. B. Ausschnitte für Amazon Elastic Compute Cloud (Amazon EC2) -Instances, Amazon Simple Storage Service (Amazon S3) -Domänen, CloudFormation-Mappings und mehr einsehen. Ausschnitte werden nach Ressourcen gruppiert. Sie finden CloudFormation-Vorlagenausschnitte für allgemeine Zwecke in der Sektion Allgemeine Vorlagenausschnitte im AWS CloudFormation-Benutzerhandbuchaus.

Erstellen einer CloudFormation Vorlage in Application Manager (Konsole)

Führen Sie die folgenden Schritte aus, um eine CloudFormation-Vorlage in Application Manager mithilfe von AWS-Managementkonsole auszuführen.

Erstellen einer CloudFormation Vorlage in Application Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Klicken Sie auf CloudFormation-Vorlagenbibliothek und wählen Sie Vorlage erstellen oder eine vorhandene Vorlage aus und gehen Sie dann auf Aktionen, Klonen.
4. Geben Sie für Name einen Namen für die Vorlage ein, mit dem Sie die erstellten Ressourcen oder den Zweck des Stacks identifizieren können.
5. (Optional) Geben Sie für Versionsname einen Namen oder eine Nummer ein, um die Vorlagenversion zu identifizieren.
6. In der Sektion Code-Editor wählen Sie entweder YAML oder JSONund geben den Vorlagencode ein oder kopieren ihn und fügen ihn ein.

7. (Optional) Wenden Sie im Abschnitt Tags ein oder mehrere Tag-Schlüssel-Name/Wert-Paare auf die Vorlage an.
8. (Optional) Geben Sie in der Sektion Berechtigungen eine AWS-Konto-ID ein und wählen Hinzufügen eines Kontos. Diese Aktion stellt die Leseberechtigung für die Vorlage bereit. Der Kontoinhaber kann die Vorlage bereitstellen und klonen, kann sie jedoch nicht bearbeiten oder löschen.
9. Wählen Sie Erstellen aus. Die Vorlage wird im Systems Manager (SSM) Document service gespeichert.

Erstellen einer CloudFormation Vorlage in Application Manager (Konsole)

Nachdem Sie den Inhalt Ihrer CloudFormation Vorlage in JSON oder YAML erstellt haben, können Sie den AWS Command Line Interface (AWS CLI) oder AWS -Tools für PowerShell verwenden, um die Vorlage als SSM-Dokument zu speichern. Ersetzen Sie jeden *`Beispiel Platzhalter für Ressourcen`* mit Ihren eigenen Informationen.

Bevor Sie beginnen

Installieren und konfigurieren Sie die AWS CLI oder AWS -Tools für PowerShell, falls noch nicht erfolgt. Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.

Linux & macOS

```
aws ssm create-document \
    --content file://path/to/template_in_json_or_yaml \
    --name "a_name_for_the_template" \
    --document-type "CloudFormation" \
    --document-format "JSON_or_YAML" \
    --tags "Key=tag-key,Value=tag-value"
```

Windows

```
aws ssm create-document ^
    --content file://C:\path\to\template_in_json_or_yaml ^
    --name "a_name_for_the_template" ^
    --document-type "CloudFormation" ^
    --document-format "JSON_or_YAML" ^
    --tags "Key=tag-key,Value=tag-value"
```

PowerShell

```
$json = Get-Content -Path "C:\path\to\template_in_json_or_yaml | Out-String
New-SSMDocument `
    -Content $json `
    -Name "a_name_for_the_template" `
    -DocumentType "CloudFormation" `
    -DocumentFormat "JSON_or_YAML" `
    -Tags "Key=tag-key,Value=tag-value"
```

Bei erfolgreicher Ausführung gibt der Befehl eine Antwort zurück, die in etwa wie folgt aussieht:

```
{
    "DocumentDescription": {
        "Hash": "c1d9640f15fbdba6deb41af6471d6ace0acc22f213bdd1449f03980358c2d4fb",
        "HashType": "Sha256",
        "Name": "MyTestCFTemplate",
        "Owner": "428427166869",
        "CreatedDate": "2021-06-04T09:44:18.931000-07:00",
        "Status": "Creating",
        "DocumentVersion": "1",
        "Description": "My test template",
        "PlatformTypes": [],
        "DocumentType": "CloudFormation",
        "SchemaVersion": "1.0",
        "LatestVersion": "1",
        "DefaultVersion": "1",
        "DocumentFormat": "YAML",
        "Tags": [
            {
                "Key": "Templates",
                "Value": "Test"
            }
        ]
    }
```

Bearbeiten einer Vorlage

Führen Sie die folgenden Schritte aus, um eine CloudFormation-Vorlage in Application Manager zu bearbeiten. Vorlagenänderungen sind in CloudFormation verfügbar, nachdem Sie ein Stack bereitgestellt haben, das die aktualisierte Vorlage verwendet.

Bearbeiten einer CloudFormation Vorlage in Application Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie CloudFormation-Vorlagenbibliothek aus.
4. Wählen Sie eine Vorlage aus und wählen Sie dann Actions (Akitionen), Edit (Bearbeiten). Sie können den Namen einer Vorlage nicht ändern, aber Sie können alle anderen Details ändern.
5. Wählen Sie Speichern. Die Vorlage wird im Systems Manager-Dokumentdienst gespeichert.

Verwenden von Application Manager zur Verwaltung von CloudFormation-Stacks

Application Manager, ein Tool in AWS Systems Manager, unterstützt Sie bei der Bereitstellung und Verwaltung von Ressourcen für Ihre Anwendungen durch die Integration in AWS CloudFormation. Sie können CloudFormation-Vorlagen und -Stacks in Application Manager erstellen, bearbeiten und löschen. Bei einem Stack handelt es sich um eine Sammlung von AWS-Ressourcen, die Sie als einzelne Einheit verwalten können. Dies bedeutet, dass Sie eine Sammlung von AWS-Ressourcen mithilfe von CloudFormation Stacks erstellen, aktualisieren oder löschen können. Eine Vorlage ist eine formatierte Textdatei in JSON oder YAML, die die Ressourcen angibt, die Sie in Ihren Stacks bereitstellen möchten. Dieser Abschnitt enthält folgende Informationen.

Themen

- Erstellen eines Stacks
- Aktualisieren eines Stacks

Erstellen eines Stacks

In den folgenden Verfahren wird beschrieben, wie Sie ein CloudFormation-Stack mithilfe von Application Manager erstellen. Ein Stack basiert auf einer Vorlage. Wenn Sie einen Stack erstellen, können Sie entweder eine vorhandene Vorlage auswählen oder eine neue erstellen. Nachdem Sie den Stack erstellt haben, versucht das System sofort, die im Stack identifizierten Ressourcen zu erstellen. Nachdem das System die Ressourcen erfolgreich bereitgestellt hat, können die Vorlage und der Stack in Application Manager und CloudFormation eingesehen und bearbeitet werden.

> **Note**
>
> Es fallen keine Gebühren an beim Erstellen eines Stacks in Application Manager an, aber Sie zahlen für AWS-Ressourcen, die im Stack erstellt wurden.

Erstellen eines CloudFormation -Stacks mithilfe von Application Manager (Konsole)

Gehen Sie wie folgt vor, um einen neuen Stack mithilfe von Application Manager in der AWS-Managementkonsole.

Erstellen eines CloudFormation-Stacks

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie Anwendung erstellen, CloudFormation-Stack aus.
4. In der Sektion Vorbereiten einer -Vorlage wählen Sie eine Option aus. Wenn Sie Vorhandene Vorlage verwenden wählen, können Sie zudem die Registerkarten in der Sektion Auswahl einer Vorlage verwenden, um die gewünschte Vorlage zu suchen. (Wenn Sie eine der anderen Optionen auswählen, schließen Sie den Assistenten ab, um eine Vorlage vorzubereiten.)
5. Klicken Sie auf die Schaltfläche neben einem Vorlagennamen und wählen Sie Weiter aus.
6. Überprüfen Sie auf der Seite Vorlagendetails angeben die Details der Vorlage, um sicherzustellen, dass der Prozess die gewünschten Ressourcen erstellt.

   - (Optional) Wenden Sie im Abschnitt Tags ein oder mehrere Tag-Schlüssel-Name/Wert-Paare auf die Vorlage an.
   - Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung.
   - Wählen Sie Weiter aus.
7. Geben Sie auf der Seite Stack-Details bearbeiten für Stack-Name einen Namen ein, der Ihnen hilft, die vom Stack erstellten Ressourcen oder seinen Zweck zu identifizieren.

   - Die Sektion Parameter enthält alle optionalen und erforderlichen Parameter, die in der Vorlage angegeben sind. Geben Sie in jedes Feld einen oder mehrere Parameter ein.

- (Optional) Wenden Sie im Bereich Tags ein oder mehrere Tag-Schlüsselname/-wertpaare auf den Stack an.
- (Optional) Klicken Sie im Bereich Berechtigungen eine AWS Identity and Access Management(IAM) -Rolle oder einen IAM Amazon-Ressourcennamen (ARN) ein. Das System verwendet die angegebene Dienstrolle, um alle in Ihrem Stack angegebenen Ressourcen zu erstellen. Wenn Sie keine IAM-Rolle angeben, verwendet CloudFormation eine temporäre Sitzung, die das System anhand Ihrer Benutzeranmeldeinformationen erstellt. Weitere Informationen über diese IAM-Rolle finden Sie unter CloudFormation-Servicerolle im AWS CloudFormation-Benutzerhandbuch.
- Wählen Sie Weiter aus.

8. Überprüfen Sie auf der Seite Überprüfung und Bereitstellung alle Details des Stacks. Wählen Sie eine Bearbeiten-Schaltfläche auf dieser Seite, um Änderungen vorzunehmen.
9. Wählen Sie Stack bereitstellen.

Application Manager zeigt die Seite CloudFormation-Stacks und den Status der Stackerstellung und -bereitstellung. Wenn CloudFormation den Stack nicht erstellen und bereitstellen kann, lesen Sie die folgenden Themen im AWS CloudFormation-Benutzerhandbuch.

- Stack-Statuscodes
- CloudFormation-Fehlerbehebung

Nachdem Ihre Stack-Ressourcen bereitgestellt und ausgeführt wurden, können Benutzer Ressourcen direkt bearbeiten, indem sie den zugrunde liegenden Service verwenden, der die Ressource erstellt hat. Beispielsweise kann ein Benutzer mit der Amazon Elastic Compute Cloud (Amazon EC2) Konsole eine Server-Instance aktualisieren, die als Teil eines CloudFormation-Stacks erstellt wurde. Einige Änderungen können versehentlich oder absichtlich vorgenommen werden, um auf zeitkritische Betriebsereignisse zu reagieren. Unabhängig davon können Änderungen, die außerhalb von CloudFormation vorgenommen werden, Stack-Aktualisierungen oder Löschvorgänge erschweren. Mit der Abweichungserkennung oder dem Abweichungsstatus können Sie Stack-Ressourcen identifizieren, an denen Konfigurationsänderungen außerhalb des CloudFormation-Managements vorgenommen wurden. Informationen zum Abweichungsstatus finden Sie unter Erkennen von nicht verwalteten Konfigurationsänderungen an Stacks und Ressourcen.

Erstellen eines CloudFormation -Stacks mithilfe von Application Manager (Befehlszeile)

Verwenden Sie die folgenden AWS Command Line Interface(AWS CLI) Verfahren, um einen Stack mithilfe einer CloudFormation Vorlage bereitzustellen, die als SSM-Dokument in Systems Manager gespeichert ist. Ersetzen Sie jeden *Beispiel Platzhalter für Ressourcen* mit Ihren eigenen Informationen. Weitere Informationen zu anderen AWS CLI-Verfahren zum Erstellen von Stacks finden Sie unter Erstellen eines Stacks im AWS CloudFormation-Benutzerhandbuch.

Bevor Sie beginnen

Installieren und konfigurieren Sie die AWS CLI oder AWS -Tools für PowerShell, falls noch nicht erfolgt. Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version der AWS CLI und Installieren des AWS -Tools für PowerShell.

Linux & macOS

```
aws cloudformation create-stack \
    --stack-name a_name_for_the_stack \
    --template-url "ssm-doc://arn:aws:ssm:Region:account_ID:document/template_name"
 \
```

Windows

```
aws cloudformation create-stack ^
     --stack-name a_name_for_the_stack ^
     --template-url "ssm-doc://arn:aws:ssm:Region:account_ID:document/template_name"
 ^
```

PowerShell

```
New-CFNStack `
    -StackName "a_name_for_the_stack" `
    -TemplateURL "ssm-doc://arn:aws:ssm:Region:account_ID:document/template_name" `
```

Aktualisieren eines Stacks

Sie können Updates auf einem CloudFormation-Stack bereitstellen, indem Sie den Stack direkt in Application Manager bearbeiten. Mit einer direkten Aktualisierung legen Sie Aktualisierungen für eine Vorlage oder Eingabeparameter fest. Nachdem Sie die Änderungen gespeichert und bereitgestellt

haben, aktualisiert CloudFormation die AWS-Ressourcen entsprechend den von Ihnen angegebenen Änderungen.

Vor der Aktualisierung können Sie eine Vorschau der Änderungen anzeigen, die CloudFormation an Ihrem Stack mithilfe von Änderungssets vornehmen wird. Weitere Informationen finden Sie unter Aktualisieren von Stacks mithilfe von Änderungssets im AWS CloudFormation-Benutzerhandbuch.

So aktualisieren Sie einen CloudFormation-Stack in Application Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie das Optionsfeld neben dem Namen einer Anwendung und dann Aktionen, Stack aktualisieren aus.
4. Wählen Sie auf der Seite Vorlagenquelle angeben eine der folgenden Optionen aus und wählen Sie dann Next (Weiter).
   - Wählen Sie Aktuell im Stack bereitgestellten Vorlagencode verwenden, um eine Vorlage anzuzeigen. Wählen Sie in der Liste Versions (Versionen) eine Vorlagenversion aus und wählen Sie dann Next (Weiter) aus.
   - Wählen Sie Wechseln zu einer anderen Vorlage, um eine neue Vorlage für den Stack auszuwählen oder zu erstellen.
5. Wenn Sie die Änderungen an der Vorlage vorgenommen haben, wählen Sie Weiter aus.
6. Auf der Seite Stackdetails bearbeiten können Sie Parameter, Tags und Berechtigungen bearbeiten. Sie können den Namen eines Stacks nicht ändern. Nehmen Sie die gewünschten Änderungen vor und wählen Sie dann Weiter.
7. Überprüfen Sie auf der Seite Überprüfung und Bereitstellung alle Details des Stacks und wählen Sie dann Stack bereitstellen.

Arbeiten mit Clustern in Application Manager

Dieser Abschnitt enthält Themen, die Ihnen bei der Arbeit mit Container-Clustern von Amazon Elastic Container Service (Amazon ECS) und Amazon Elastic Kubernetes Service (Amazon EKS)Application Manager, einer Komponente von, helfen sollen. AWS Systems Manager

Inhalt

- Arbeiten mit Amazon ECS in Application Manager

- Arbeiten mit Amazon EKS in Application Manager
- Arbeiten mit Runbooks für Cluster

Arbeiten mit Amazon ECS in Application Manager

Mit Application Manager, einem Tool in AWS Systems Manager, können Sie Ihre Amazon Elastic Container Service (Amazon ECS)-Cluster-Infrastruktur anzeigen und verwalten. Application Manager wendet ein Tag auf Ihren Amazon-ECS-Cluster an und verwendet dabei den Amazon-Ressourcennamen (ARN) des Clusters als Tag-Wert. Application Manager bietet eine Komponentenlaufzeitansicht der Rechen-, Netzwerk- und Speicherressourcen in einem Cluster.

> **Note**
>
> Sie können keine Betriebsinformationen zu Ihren Containern in Application Manager verwalten oder anzeigen. Sie können nur Betriebsinformationen über die Infrastruktur verwalten und anzeigen, die Ihre Amazon-ECS-Ressourcen hostet.

Aktionen, die Sie auf dieser Seite ausführen können

Hier sind folgende Aktionen möglich:

- Wählen Sie Verwalten von Clustern, um den Cluster in Amazon ECS zu öffnen.
- Wählen Sie Alle anzeigen, um eine Liste der Ressourcen in Ihrem Cluster anzuzeigen.
- Wählen Sie Anzeigen in CloudWatch, um Ressourcenalarme in Amazon CloudWatch anzuzeigen.
- Wählen Sie Manage nodes (Verwalten von Knoten) oder Manage Fargate profiles, (Fargate-Profile verwalten) um diese Ressourcen in Amazon ECS anzuzeigen.
- Wählen Sie eine Ressourcen-ID aus, um detaillierte Informationen darüber in der Konsole anzuzeigen, in der sie erstellt wurde.
- Anzeigen einer Liste der OpsItems im Zusammenhang mit Ihren Clustern.
- Zeigen Sie einen Verlauf von Runbooks an, die auf Ihren Clustern ausgeführt wurden.

So öffnen Sie den ECS-Cluster

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Application Manager aus.
3. Wählen Sie in der Sektion Container-Cluster ECS-Cluster.
4. Wählen Sie einen Cluster in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.

Arbeiten mit Amazon EKS in Application Manager

Application Manager, ein Tool in AWS Systems Manager, lässt sich in Amazon Elastic Kubernetes Service (Amazon EKS) integrieren, um Informationen zum Zustand Ihrer Amazon-EKS-Cluster-Infrastruktur bereitzustellen. Application Manager wendet ein Tag auf Ihren Amazon-EKS-Cluster an und verwendet dabei den Amazon-Ressourcennamen (ARN) des Clusters als Tag-Wert. Application Manager bietet eine Komponentenlaufzeitansicht der Rechen-, Netzwerk- und Speicherressourcen in einem Cluster.

> **Note**
>
> Sie können keine Betriebsinformationen zu Ihren Amazon EKS-Pods oder -Containern in Application Manager verwalten. Sie können nur Betriebsinformationen zu der Infrastruktur verwalten und anzeigen, die Ihre Amazon EKS-Ressourcen hostet.

Aktionen, die Sie auf dieser Seite ausführen können

Hier sind folgende Aktionen möglich:

- Wählen Sie Verwalten von Clustern, um den Cluster in Amazon EKS zu öffnen.
- Wählen Sie Alle anzeigen, um eine Liste der Ressourcen in Ihrem Cluster anzuzeigen.
- Wählen Sie Anzeigen in CloudWatch, um Ressourcenalarme in Amazon CloudWatch anzuzeigen.
- Wählen Sie Manage nodes (Verwalten von Knoten) oder Manage Fargate profiles (Fargate-Profile verwalten), um diese Ressourcen in Amazon EKS anzuzeigen.
- Wählen Sie eine Ressourcen-ID aus, um detaillierte Informationen darüber in der Konsole anzuzeigen, in der sie erstellt wurde.
- Anzeigen einer Liste der OpsItems im Zusammenhang mit Ihren Clustern.
- Zeigen Sie einen Verlauf von Runbooks an, die auf Ihren Clustern ausgeführt wurden.

So öffnen Sie eine EKS-Cluster-Anwendung

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. In der Sektion Container Cluster wählen Sie EKS-Cluster.
4. Wählen Sie einen Cluster in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.

Arbeiten mit Runbooks für Cluster

Sie können Probleme mit AWS-Ressourcen von Application Manager (einem Tool in AWS Systems Manager) mithilfe von Systems Manager Automation Runbooks beheben. Wenn Sie Runbook starten von einem Application Manager-Cluster wählen, zeigt das System eine gefilterte Liste von Runbooks basierend auf dem Typ der Ressourcen in Ihrem Cluster an. Wenn Sie das Runbook auswählen, das Sie starten möchten, öffnet Systems Manager die Seite Ausführen des Automatisierungsdokuments.

Bevor Sie beginnen

Bevor Sie ein Runbook von Application Managerstarten, gehen Sie wie folgt vor:

- Stellen Sie sicher, dass Sie über die richtigen Berechtigungen zum Starten von Runbooks verfügen. Weitere Informationen finden Sie unter Einrichten der Automatisierung.
- Lesen Sie die Dokumentation zur Automatisierungsprozedur zum Starten von Runbooks. Weitere Informationen finden Sie unter Führen Sie einen automatisierten Vorgang aus, der von Systems Manager Automation unterstützt wird.
- Wenn Sie Runbooks auf mehreren Ressourcen gleichzeitig starten möchten, lesen Sie die Dokumentation zur Verwendung von Zielen und Tarifkontrollen. Weitere Informationen finden Sie unter Automatisierte Abläufe in großem Umfang ausführen.

So starten Sie ein Runbook für Cluster von Application Manager

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Application Manager aus.
3. In der Sektion Container-Cluster wählen Sie einen Containertyp aus.
4. Wählen Sie den Cluster in der Liste aus. Application Manager öffnet die Registerkarte Übersicht.

5. Wählen Sie auf der Registerkarte Runbooks die Option Start Runbook (Runbook starten) aus. Application Manager öffnet die Seite Execute automation document (Automatisierungsdokument ausführen) auf einer neuen Registerkarte. Weitere Informationen zu den Optionen auf der Seite Ausführen des Automationsdokuments finden Sie unter Führen Sie einen automatisierten Vorgang aus, der von Systems Manager Automation unterstützt wird.

AWS Systems Manager Parameter Store

Parameter Store, ein Tool in AWS Systems Manager, bietet sicheren, hierarchischen Speicher für die Verwaltung von Konfigurationsdaten und Geheimnissen. Sie können Daten wie Kennwörter, Datenbankzeichenfolgen Amazon Machine Image (AMI) IDs und Lizenzcodes als Parameterwerte speichern. Sie können Werte als Klartext oder als verschlüsselte Daten speichern. Sie können Systems Manager-Parameter in Skripts, Befehlen, SSM-Dokumenten und Konfigurations- und Automatisierungs-Workflows referenzieren, indem Sie den eindeutigen Namen verwenden, den Sie beim Erstellen des Parameters angegeben haben. Um mit Parameter Store zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Parameter Store aus.

Parameter Store ist auch in Secrets Manager integriert. Sie können Secrets-Manager-Geheimnisse abrufen, wenn Sie andere AWS-Services verwenden, die bereits Referenzen zu Parameter Store-Parametern unterstützen. Weitere Informationen finden Sie unter Referenzieren von AWS Secrets Manager-Geheimnissen über Parameter Store-Parameter.

> **Note**
>
> Um Lebenszyklen für die Passwortrotation zu implementieren, verwenden Sie. AWS Secrets Manager Sie können Datenbankanmeldeinformationen, API-Schlüssel und andere geheime Informationen mit Secrets Manager während ihres gesamten Lebenszyklus mühelos rotieren, verwalten und abfragen. Weitere Informationen finden Sie unter Was ist? AWS Secrets Manager im AWS Secrets Manager Benutzerhandbuch.

Welche Vorteile bietet Parameter Store meiner Organisation?

Parameter Store bietet die folgenden Vorteile:

- Verwenden Sie einen sicheren, skalierbaren, gehosteten Verschlüsselungsmanagementservice ohne zu verwaltende Server.
- Verbessern Sie Ihre Sicherheit, indem Sie Ihre Daten von Ihrem Code trennen.

- Speichern Sie Konfigurationsdaten und verschlüsselte Zeichenfolgen in Hierarchien und verfolgen Sie Versionen nach.
- Steuern und prüfen Sie Zugriff genau.
- Speichern Sie Parameter zuverlässig, da Parameter Store in mehreren Availability Zones in einer AWS-Region gehostet wird.

An wen richtet sich Parameter Store?

- Jeder AWS Kunde, der eine zentrale Möglichkeit zur Verwaltung von Konfigurationsdaten haben möchte.
- Softwareentwickler, die verschiedene Logins und Referenzströme speichern möchten.
- Administratoren, die Benachrichtigungen erhalten möchten, wenn ihre Secrets und Passwörter geändert werden oder nicht.

Über welche Features verfügt Parameter Store?

- Änderungsbenachrichtigung

  Sie können Änderungsbenachrichtigungen konfigurieren und automatisierte Aktionen für beide Parameter und Parameterrichtlinien auslösen. Weitere Informationen finden Sie unter Einrichten von Benachrichtigungen oder Auslösen von Aktionen basierend auf Parameter Store-Ereignissen.
- Organisieren von Parametern

  Sie können Ihre Parameter individuell markieren, um anhand der Tags, die Sie ihnen zugewiesen haben, einen oder mehrere Parameter zu identifizieren. Sie können Parameter z. B. nach bestimmten Umgebungen oder Abteilungen taggen.
- Beschriftungsversionen

  Sie können einen Alias für Versionen Ihres Parameters zuordnen, indem Sie Beschriftungen erstellen. Dank Beschriftungen können Sie sich den Zweck einer Parameterversion merken, wenn mehrere Versionen vorhanden sind.
- Datenvalidierung

  Sie können Parameter erstellen, die auf eine Amazon Elastic Compute Cloud (Amazon EC2) - Instance verweisen, und diese Parameter Parameter Store validieren, um sicherzustellen, dass sie auf den erwarteten Ressourcentyp verweisen, dass die Ressource vorhanden ist und dass der

Kunde berechtigt ist, die Ressource zu verwenden. Sie können beispielsweise einen Parameter mit Amazon Machine Image (AMI) ID als einen Wert mit `aws:ec2:image`-Datentyp erstellen und Parameter Store führt einen asynchronen Validierungsvorgang aus, um sicherzustellen, dass der Parameterwert die Formatierungsanforderungen für eine AMI ID erfüllt und dass das angegebene AMI in Ihrem AWS-Konto verfügbar ist.

- Referenz-Secrets

  Parameter Storeist integriert, AWS Secrets Manager sodass Sie Secrets Manager abrufen können, wenn Sie andere verwenden AWS-Services , die bereits Verweise auf Parameter Store Parameter unterstützen.

- Parameter mit anderen Konten teilen

  Sie können die Konfigurationsdaten optional in einer einzigen Datei zentralisieren AWS-Konto und Parameter mit anderen Konten teilen, die darauf zugreifen müssen.

- Zugänglich von anderen AWS-Services

  Sie können Parameter Store Parameter mit anderen Systems Manager Manager-Tools und AWS-Services zum Abrufen von Geheimnissen und Konfigurationsdaten aus einem zentralen Speicher verwenden. Parameter funktionieren mit Systems Manager Manager-Tools wie Run Command Automation und State Manager Tools in AWS Systems Manager. Sie können Parameter auch in einer Reihe anderer referenzieren AWS-Services, z. B. in den folgenden:

  - Amazon Elastic Compute Cloud (Amazon EC2)
  - Amazon Elastic Container Service (Amazon ECS)
  - AWS Secrets Manager
  - AWS Lambda
  - AWS CloudFormation
  - AWS CodeBuild
  - AWS CodePipeline
  - AWS CodeDeploy

- Integrieren Sie mit anderen AWS-Services

  Konfigurieren Sie die Integration mit den folgenden Optionen AWS-Services für Verschlüsselung, Benachrichtigung, Überwachung und Prüfung:

  - AWS Key Management Service (AWS KMS)
  - Amazon-Simple-Notification-Service (Amazon-SNS)

- Amazon CloudWatch: Weitere Informationen finden Sie unterKonfigurieren von EventBridge-Regeln für Parameter und Parameter-Richtlinien.
- Amazon EventBridge: Weitere Informationen finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen undReferenz: EventBridge Amazon-Ereignistypen und -muster für Systems Manager.
- AWS CloudTrail: Weitere Informationen finden Sie unter Protokollierung von AWS Systems Manager-API-Aufrufen mit AWS CloudTrail.

Was ist ein Parameter?

Ein Parameter Store-Parameter ist eine beliebige Datenmenge, die in Parameter Store gespeichert wird, z. B. ein Textblock, eine Namensliste, ein Passwort, eine AMI-ID, ein Lizenzschlüssel usw. Sie können diese Daten zentral und sicher in Ihren Skripten, Befehlen und SSM-Dokumenten referenzieren.

Wenn Sie auf einen Parameter verweisen, geben Sie den Parameternamen unter Verwendung der folgenden Konvention an:

{{ssm:*parameter-name*}}

> **Note**
>
> Parameter können nicht referenziert oder in den Werten anderer Parameter verschachtelt werden. Sie können {{}} oder {{ssm:*parameter-name*}} nicht in einen Parameterwert aufnehmen.

Parameter Store bietet Unterstützung für drei Arten von Parametern: `String`, `StringList` und `SecureString`.

Mit einer Ausnahme geben Sie beim Erstellen oder Aktualisieren eines Parameters den Parameterwert als Klartext ein und Parameter Store führt keine Validierung für den eingegebenen Text aus. Für `String` Parameter können Sie den Datentyp jedoch als angeben`aws:ec2:image`, und es wird Parameter Store überprüft, ob der von Ihnen eingegebene Wert das richtige Format für einen Amazon hat EC2 AMI; zum Beispiel:`ami-12345abcdeEXAMPLE`.

Parametertyp: String

Standardmäßig nimmt der `String`-Parameter jeden beliebigen Textblock auf, den Sie eingeben. Zum Beispiel:

- `abc123`
- `Example Corp`
- `<img src="images/bannerImage1.png"/>`

Parametertyp: StringList

Die Werte der `StringList`-Parameter enthalten eine kommagetrennte Liste von Werten wie in den folgenden Beispielen gezeigt.

```
Monday,Wednesday,Friday
```

```
CSV,TSV,CLF,ELF,JSON
```

Typ des Parameters: SecureString

Der Wert eines `SecureString`-Parameters kann aus beliebigen vertraulichen Daten bestehen, die auf sichere Weise gespeichert und referenziert werden müssen. Wenn Sie Daten haben, die Benutzer nicht ändern oder als Klartext referenzieren sollen (z. B. Passwörter oder Lizenzschlüssel), erstellen Sie diese Parameter mit dem `SecureString`-Datentyp.

> **Important**
>
> Speichern Sie keine vertraulichen Daten in einem `String`- oder `StringList`-Parameter. Verwenden Sie für alle vertraulichen Daten, die verschlüsselt bleiben müssen, nur den `SecureString`-Parametertyp.
> Weitere Informationen finden Sie unter Erstellen eines SecureString Parameters mit dem AWS CLI.

Wir empfehlen die Verwendung von `SecureString`-Parametern in den folgenden Szenarien:

- Sie möchten data/parameters Across verwenden, AWS-Services ohne die Werte als Klartext in Befehlen, Funktionen, Agentenprotokollen oder CloudTrail Logs verfügbar zu machen.

- Sie möchten steuern, welche Personen auf vertrauliche Daten zugreifen können.
- Sie möchten in der Lage sein, zu überprüfen, wann auf vertrauliche Daten zugegriffen wird (CloudTrail).
- Sie möchten Ihre sensiblen Daten verschlüsseln und Sie möchten Ihre eigenen Verschlüsselungsschlüssel für die Zugriffsverwaltung verwenden.

> **Important**
>
> Nur der Wert eines `SecureString`-Parameters wird verschlüsselt. Der Name des Parameters, die Beschreibung und andere Eigenschaften sind nicht verschlüsselt.

Sie können den `SecureString` Parametertyp für Textdaten verwenden, die Sie verschlüsseln möchten, z. B. Kennwörter, Anwendungsgeheimnisse, vertrauliche Konfigurationsdaten oder andere Arten von Daten, die Sie schützen möchten. `SecureString`Daten werden mit einem Schlüssel ver- und entschlüsselt. AWS KMS Sie können entweder einen Standard-KMS-Schlüssel verwenden, der von bereitgestellt wird, AWS oder Sie können Ihren eigenen AWS KMS key erstellen und verwenden. (Verwenden Sie Ihre eigenen AWS KMS key , wenn Sie den Benutzerzugriff auf `SecureString`-Parameter einschränken möchten. Weitere Informationen finden Sie unter IAM-Berechtigungen für die Verwendung von AWS Standardschlüsseln und vom Kunden verwalteten Schlüsseln.)

Sie können `SecureString` Parameter auch zusammen mit anderen verwenden AWS-Services. Im folgenden Beispiel ruft die Lambda-Funktion mithilfe der GetParametersAPI einen `SecureString` Parameter ab.

```
import json
import boto3
ssm = boto3.client('ssm', 'us-east-2')
def get_parameters():
    response = ssm.get_parameters(
        Names=['LambdaSecureString'],WithDecryption=True
    )
    for parameter in response['Parameters']:
        return parameter['Value']

def lambda_handler(event, context):
    value = get_parameters()
    print("value1 = " + value)
```

```
    return value  # Echo back the first key value
```

AWS KMS Verschlüsselung und Preisgestaltung

Wenn Sie bei der Erstellung Ihres `SecureString` Parameters den Parametertyp wählen, verschlüsselt Systems AWS KMS Manager den Parameterwert.

> **Important**
>
> Parameter Store unterstützt nur KMS-Schlüssel zur symmetrischen Verschlüsselung. Sie können keinen KMS-Schlüssel zur asymmetrischen Verschlüsselung verwenden, um Ihre Parameter zu verschlüsseln. Wie Sie feststellen, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, erfahren Sie unter Erkennen symmetrischer und asymmetrischer Schlüssel im AWS Key Management Service -Entwicklerhandbuch.

Für die Erstellung eines `SecureString` Parameters fallen keine Gebühren Parameter Store an, es fallen jedoch Gebühren für die Verwendung der AWS KMS Verschlüsselung an. Weitere Informationen finden Sie unter AWS Key Management Service -Preise.

Weitere Informationen zu Von AWS verwaltete Schlüssel und vom Kunden verwalteten Schlüsseln finden Sie unter AWS Key Management Service Konzepte im AWS Key Management Service Entwicklerhandbuch. Weitere Informationen zur AWS KMS Verschlüsselung Parameter Store und zur Verschlüsselung finden Sie unter AWS Systems ManagerParameter StoreAnwendungsmöglichkeiten AWS KMS.

> **Note**
>
> Verwenden Sie die AWS KMS `DescribeKey` Operation Von AWS verwalteter Schlüssel, um eine anzuzeigen. Dieses AWS Command Line Interface (AWS CLI) Beispiel dient `DescribeKey` zum Anzeigen eines Von AWS verwalteter Schlüssel.
>
> ```
> aws kms describe-key --key-id alias/aws/ssm
> ```

Weitere Informationen

- Einen SecureString Parameter in einer Domain erstellen Parameter Store und einen Knoten mit einer Domain verbinden (PowerShell)

- Verwendung von Parameter Store für den sicheren Zugriff auf Geheimnisse und Config-Daten in CodeDeploy
- Interessante Artikel zu Amazon EC2 Systems Manager Parameter Store

Beschränkungen der Parametergröße

Parameter Store hat je nach verwendeter Parameter-Tier unterschiedliche Größenbeschränkungen für die Parameterwerte:

- Standard-Parameter: Maximale Wertgröße von 4 KB
- Erweiterte Parameter: Maximale Wertgröße von 8 KB

Wenn Sie Parameterwerte speichern müssen, die größer als 4 KB sind, müssen Sie die erweiterte Parameter-Tier verwenden. Erweiterte Parameter bieten zusätzliche Funktionen, sind jedoch mit Gebühren für Ihr AWS Konto verbunden. Weitere Hinweise zu den Parameter-Tiers und den enthaltenen Features finden Sie unter Verwalten von Parameterstufen.

Eine vollständige Liste der Kontingente und Beschränkungen von Parameter Store finden Sie unter AWS Systems Manager Endpunkte und Kontingente in der Allgemeinen Referenz für AWS .

Einrichten von Parameter Store

Um Parameter in Parameter Store, einem Tool in AWS Systems Manager, einrichten zu können, müssen Sie zunächst AWS Identity and Access Management (IAM)-Richtlinien konfigurieren, die den Benutzern in Ihrem Konto die Berechtigung zur Ausführung der von Ihnen festgelegten Aktionen erteilen.

In diesem Abschnitt finden Sie Informationen darüber, wie Sie diese Richtlinien mithilfe der IAM-Konsole manuell konfigurieren und sie Benutzern und Benutzergruppen zuweisen. Darüber hinaus können Sie Richtlinien erstellen und zuordnen, um zu steuern, welche Parameteraktionen auf einem verwalteten Knoten ausgeführt werden dürfen.

Außerdem enthält dieser Abschnitt Informationen dazu, wie Sie Amazon EventBridge-Regeln erstellen, anhand derer Sie Benachrichtigungen über Änderungen an Systems Manager-Parametern erhalten. Anhand der EventBridge-Regeln können Sie zudem andere Aktionen in AWS aufrufen, die auf Änderungen in Parameter Store basieren.

Inhalt

- Einschränken des Zugriffs auf Parameter Store-Parameter mithilfe von IAM-Richtlinien
- Verwalten von Parameterstufen
- Parameter Store-Durchsatz erhöhen oder zurücksetzen
- Einrichten von Benachrichtigungen oder Auslösen von Aktionen basierend auf Parameter Store-Ereignissen

Einschränken des Zugriffs auf Parameter Store-Parameter mithilfe von IAM-Richtlinien

Sie schränken den Zugriff auf AWS Systems Manager Parameter mithilfe von AWS Identity and Access Management (IAM) ein. Genauer gesagt können Sie IAM-Richtlinien erstellen, die den Zugriff auf die folgenden API-Operationen beschränken:

- DeleteParameter
- DeleteParameters
- DescribeParameters
- GetParameter
- GetParameters
- GetParameterHistory
- GetParametersByPath
- PutParameter

Wenn Sie IAM-Richtlinien verwenden, um den Zugriff auf Systems Manager-Parameter einzuschränken, sollten Sie restriktive IAM-Richtlinien erstellen und verwenden. Die folgende Richtlinie ermöglicht z. B. den Aufruf der API-Operationen `DescribeParameters` und `GetParameters` für einen eingeschränkten Satz von Ressourcen. Das bedeutet, dass der Benutzer Informationen zu allen Parametern, die mit `prod-*` beginnen, abrufen und diese verwenden kann.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "ssm:DescribeParameters"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameters"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/prod-*"
        }
    ]
}
```

Important

Wenn ein Benutzer Zugriff auf einen Pfad hat, kann er auf alle Ebenen dieses Pfads zugreifen. Wenn ein Benutzer beispielsweise die Berechtigung für den Zugriff auf den Pfad `/a` besitzt, dann kann er auch auf `/a/b` zugreifen. Selbst wenn einem Benutzer in IAM der Zugriff auf den Parameter `/a/b` ausdrücklich verweigert wurde, kann er dennoch die `GetParametersByPath`-API-Operation rekursiv für `/a` aufrufen und `/a/b` anzeigen.

Vertrauenswürdigen Administratoren können Sie mithilfe einer Richtlinie ähnlich dem folgenden Beispiel Zugriff auf alle API-Operationen für Systems Manager-Parameter gewähren. Mit dieser Richtlinie erhält der Benutzer vollen Zugriff auf alle Produktionsparameter, die mit `dbserver-prod-*` beginnen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter",
                "ssm:DeleteParameter",
```

```
                "ssm:GetParameterHistory",
                "ssm:GetParametersByPath",
                "ssm:GetParameters",
                "ssm:GetParameter",
                "ssm:DeleteParameters"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/dbserver-
prod-*"
        },
        {
            "Effect": "Allow",
            "Action": "ssm:DescribeParameters",
            "Resource": "*"
        }
    ]
}
```

Verweigern von Berechtigungen

Jede API ist einzigartig und verfügt über unterschiedliche Operationen und Berechtigungen, die Sie einzeln zulassen oder verweigern können. Eine explizite Zugriffsverweigerung überschreibt jede Zugriffserlaubnis in einer Richtlinie.

> **Note**
>
> Der Standardschlüssel AWS Key Management Service (AWS KMS) verfügt über `Decrypt` Berechtigungen für alle IAM-Prinzipale innerhalb von. AWS-Konto Wenn Sie unterschiedliche Zugriffsebenen für `SecureString`-Parameter in Ihrem Konto haben möchten, raten wir Ihnen davon ab, den Standardschlüssel zu verwenden.

Wenn Sie möchten, dass alle API-Operationen, die Parameterwerte abrufen, das gleiche Verhalten haben, dann können Sie ein Muster wie `GetParameter*` in einer Richtlinie verwenden. Im folgenden Beispiel wird gezeigt, wie Sie `GetParameter`, `GetParameters`, `GetParameterHistory` und `GetParametersByPath` für alle Parameter, die mit `prod-*` beginnen, verweigern.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ssm:GetParameter*"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/prod-*"
        }
    ]
}
```

Das folgende Beispiel zeigt, wie einige Befehle verweigert werden, während der Benutzer andere Befehle für alle Parameter ausführen kann, die mit `prod-*` beginnen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ssm:PutParameter",
                "ssm:DeleteParameter",
                "ssm:DeleteParameters",
                "ssm:DescribeParameters"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParametersByPath",
                "ssm:GetParameters",
                "ssm:GetParameter",
                "ssm:GetParameterHistory"
```

```
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/prod-*"
        }
    ]
}
```

> **Note**
>
> Der Parameterverlauf umfasst alle Parameterversionen, einschließlich der aktuellen. Wenn einem Benutzer daher die Berechtigung für `GetParameter`, `GetParameters` und `GetParameterByPath` nicht gewährt wird, er aber die Berechtigung für `GetParameterHistory` erhält, kann er den aktuellen Parameter, einschließlich `SecureString`, unter Verwnedung von `GetParameterHistory` sehen.

Erlauben nur bestimmter Parameter für die Ausführung auf Knoten

Sie können den Zugriff so steuern, dass verwaltete Knoten nur von Ihnen angegebene Parameter ausführen können.

Wenn Sie bei der Erstellung Ihres `SecureString` Parameters den Parametertyp wählen, verschlüsselt Systems AWS KMS Manager den Parameterwert. AWS KMS verschlüsselt den Wert entweder mithilfe eines Von AWS verwalteter Schlüssel oder eines vom Kunden verwalteten Schlüssels. Weitere Informationen zu AWS KMS und AWS KMS key finden Sie im AWS Key Management Service Entwicklerhandbuch.

Sie können das anzeigen, Von AWS verwalteter Schlüssel indem Sie den folgenden Befehl von der aus ausführen AWS CLI.

```
aws kms describe-key --key-id alias/aws/ssm
```

Im folgenden Beispiel dürfen Knoten einen Parameterwert nur für Parameter abrufen, die mit `prod-` beginnen. Wenn der Parameter ein `SecureString`-Parameter ist, entschlüsselt der Knoten die Zeichenfolge mit AWS KMS.

> **Note**
>
> Instance-Richtlinien wie im folgenden Beispiel werden der Instance-Rolle in IAM zugeordnet. Weitere Informationen zur Konfiguration des Zugriffs auf Systems Manager-Funktionen

einschließlich einer Anleitung für die Zuweisung von Richtlinien für Benutzer und Instances finden Sie unter Verwalten von EC2-Instances mit Systems Manager.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameters"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:parameter/prod-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/4914ec06-e888-4ea5-
a371-5b88eEXAMPLE"
            ]
        }
    ]
}
```

IAM-Berechtigungen für die Verwendung von AWS Standardschlüsseln und vom Kunden verwalteten Schlüsseln

Parameter Store`SecureString`Parameter werden mithilfe von Schlüsseln ver- und entschlüsselt AWS KMS . Sie können wählen, ob Sie Ihre `SecureString` Parameter entweder mit einem AWS KMS key oder mit dem Standard-KMS-Schlüssel von verschlüsseln möchten. AWS

Wenn Sie einen kundenverwalteten Schlüssel verwenden, muss die IAM-Richtlinie, die einem Benutzer Zugriff auf einen Parameter oder Parameterpfad erteilt, explizite `kms:Encrypt`-

Berechtigungen für den Schlüssel bereitstellen. Die folgende Richtlinie ermöglicht es einem Benutzer beispielsweise, `SecureString` Parameter zu erstellen, zu aktualisieren und anzuzeigen, die mit `prod-` dem angegebenen AWS-Region und AWS-Konto beginnen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter",
                "ssm:GetParameter",
                "ssm:GetParameters"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:parameter/prod-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:Encrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": [
                "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-12345EXAMPLE"
            ]
        }
    ]
}
```

> **Note**
>
> Um verschlüsselte erweiterte Parameter mit dem angegebenen kundenverwalteten Schlüssel zu erstellen, ist die Berechtigung `kms:GenerateDataKey` erforderlich.

Im Gegensatz hierzu haben alle Benutzer innerhalb des Kundenkontos Zugriff auf den standardmäßigen AWS -verwalteten Schlüssel. Wenn Sie diesen Standardschlüssel zum Verschlüsseln von `SecureString`-Parametern verwenden und nicht möchten, dass Benutzer mit `SecureString`-Parametern arbeiten, müssen ihre IAM-Richtlinien den Zugriff auf den Standardschlüssel ausdrücklich ablehnen, wie im folgenden Richtlinienbeispiel gezeigt.

> **Note**
>
> Sie finden den Amazon-Ressourcennamen (ARN) des Standardschlüssels in der AWS KMS -Konsole auf der Seite AWS -verwaltete Schlüssel. Der Standardschlüssel wird mit `aws/ssm` in der Spalte Alias identifiziert.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/abcd1234-ab12-cd34-ef56-
abcdeEXAMPLE"
            ]
        }
    ]
}
```

Wenn Sie in Bezug auf die `SecureString`-Parameter in Ihrem Konto eine granulare Zugriffskontrolle benötigen, sollten Sie einen kundenverwalteten Schlüssel verwenden, um den Zugriff auf diese Parameter zu schützen und einzuschränken. Wir empfehlen außerdem, die Verwendung AWS CloudTrail zur Überwachung von `SecureString` Parameteraktivitäten zu verwenden.

Weitere Informationen finden Sie unter den folgenden Themen:

- Auswertungslogik für Richtlinien im IAM-Benutzerhandbuch
- Verwenden von Schlüsselrichtlinien in AWS KMS im AWS Key Management Service - Benutzerhandbuch
- Ereignisse mit dem CloudTrail Ereignisverlauf im AWS CloudTrail Benutzerhandbuch anzeigen

Verwalten von Parameterstufen

Parameter Store, ein Tool in AWS Systems Manager, beinhaltet Standardparameter und erweiterte Parameter. Parameter werden einzeln konfiguriert, sodass sie entweder die Standardparameterstufe (Standardstufe) oder die erweiterte Parameterstufe verwenden.

Sie können einen Standardparameter jederzeit in einen erweiterten Parameter ändern. Sie können jedoch einen erweiterten Parameter nicht auf einen Standardparameter zurücksetzen. Das Zurücksetzen eines erweiterten Parameters auf einen Standardparameter würde zu Datenverlust führen, weil das System die Größe des Parameters von 8 KB auf 4 KB kürzt. Durch das Zurücksetzen würden auch etwaige dem Parameter angefügte Richtlinien entfernt. Erweiterte Parameter verwenden eine andere Form der Verschlüsselung als Standardparameter. Weitere Informationen finden Sie unter Arbeiten von AWS Systems ManagerParameter Store mit AWS KMS im AWS Key Management Service -Entwicklerhandbuch.

Wenn Sie einen erweiterten Parameter nicht mehr benötigen oder verhindern wollen, dass weitere Gebühren dafür anfallen, löschen Sie ihn und erstellen Sie ihn als Standardparameter neu.

Die folgende Tabelle beschreibt die Unterschiede zwischen den Stufen.

	Standard	Advanced
Gesamtzahl der zulässigen Parameter (pro AWS-Konto und AWS-Region)	10.000	100 000
Maximale Größe eines Parameterwerts.	4 KB	8 KB
Parameterrichtlinien verfügbar	Nein	Ja

	Standard	Advanced
		Weitere Informationen finden Sie unter Zuweisen von Parameterrichtlinien in Parameter Store.
Cost (Kosten)	Keine zusätzlichen Gebühren	Gebührenpflichtig Weitere Informationen finden Sie unter AWS Systems Manager -Preisgestaltung für Parameter Store.

Themen

- Angeben einer Standardparameterstufe
- Ändern eines Standardparameters in einen fortgeschrittenen Parameter

Angeben einer Standardparameterstufe

In Anforderungen zum Erstellen oder Aktualisieren eines Parameters (d. h. der Operation `PutParameter`) können Sie die Parameterstufe angeben, die in der Anforderung verwendet werden soll. Im Folgenden finden Sie ein Beispiel unter Verwendung der AWS Command Line Interface (AWS CLI).

Linux & macOS

```
aws ssm put-parameter \
    --name "default-ami" \
    --type "String" \
    --value "t2.micro" \
    --tier "Standard"
```

Windows

```
aws ssm put-parameter ^
    --name "default-ami" ^
    --type "String" ^
    --value "t2.micro" ^
```

```
    --tier "Standard"
```

Wenn Sie eine Stufe in der Anforderung angeben, erstellt oder aktualisiert Parameter Store den Parameter entsprechend Ihrer Anforderung. Wenn Sie jedoch nicht explizit eine Stufe in einer Anforderung angeben, bestimmt die Parameter Store-Standardstufeneinstellung, in welcher Stufe der Parameter erstellt wird.

Die Standardstufe, wenn Sie beginnen Parameter Store zu verwenden, ist die Standardparameterstufe. Wenn Sie die erweiterte Parameterstufe verwenden, können Sie einen der folgenden als Standardwert angeben:

- Erweitert: Mit dieser Option wertet Parameter Store alle Anforderungen als erweiterte Parameter aus.
- Intelligent-Tiering: Mit dieser Option wertet Parameter Store jede Anforderung aus, um zu ermitteln, ob es sich um einen Standard- oder erweiterten Parameter handelt.

  Wenn die Anforderung keine Optionen enthält, die einen erweiterten Parameter erfordern, wird der Parameter in der Standardparameterstufe erstellt. Wenn eine oder mehrere Optionen, die einen erweiterten Parameter erfordern, in der Anforderung enthalten sind, erstellt Parameter Store einen Parameter in der erweiterten Parameterstufe.

Vorteile von Intelligent-Tiering

Nachstehend sind Gründe, warum Sie Intelligent-Tiering als Standardstufe auswählen können.

Kostenkontrolle – Intelligent-Tiering hilft Ihnen, Ihre parameterbezogenen Kosten zu kontrollieren, indem immer Standardparameter erstellt werden, außer ein erweiterter Parameter ist absolut notwendig.

Automatisches Upgrade auf die erweiterte Parameterstufe – Wenn Sie eine Änderung an Ihrem Code vornehmen, die ein Upgrade eines Standardparameters auf einen erweiterten Parameter erfordert, übernimmt Intelligent-Tiering die Konvertierung für Sie. Sie müssen Ihren Code nicht ändern, um das Upgrade abzuwickeln.

Hier finden Sie einige Beispiele für automatische Upgrades:

- Ihre AWS CloudFormation Vorlagen stellen zahlreiche Parameter bereit, wenn sie ausgeführt werden. Wenn Sie durch diesen Prozess das Kontingent von 10.000 Parametern in der Stufe mit den Standardparametern erreichen, führt Intelligent-Tiering automatisch ein Upgrade auf

die Stufe mit erweiterten Parametern durch, sodass Ihre Prozesse nicht unterbrochen werden. CloudFormation

- Sie speichern einen Zertifikatwert in einem Parameter, drehen den Zertifikatswert regelmäßig und der Inhalt liegt unter dem Limit von 4 KB des Standard-Parameter-Kontingents. Wenn ein Ersatzzertifikatwert 4 KB überschreitet, aktualisiert Intelligent-Tiering den Parameter automatisch auf die erweiterte Parameterstufe.
- Sie möchten einer Parameterrichtlinie zahlreiche vorhandene Standardparameter zuordnen, die die erweiterte Parameterstufe erfordert. Anstatt die Option `--tier Advanced` in allen Aufrufen inkludieren zu müssen, um die Parameter zu aktualisieren, aktualisiert Intelligent-Tiering die Parameter automatisch auf die erweiterte Parameterstufe. Mit der Option „Intelligent-Tiering“ werden Parameter immer dann von „Standard“ auf „erweitert“ aktualisiert, wenn Kriterien für die erweiterte Parameterstufe eingeführt werden.

Optionen, für die ein erweiterter Parameter erforderlich ist, umfassen die folgenden:

- Die Inhaltsgröße des Parameters beträgt mehr als 4 KB.
- Der Parameter verwendet eine Parameterrichtlinie.
- Derzeit sind in Ihrem System bereits mehr als 10.000 Parameter vorhanden. AWS-Konto AWS-Region

Optionen für die Standardstufe

Die Stufenoptionen, die Sie als Standard festlegen können, umfassen die folgenden.

- Standard – Der Die Standardparameterstufe, wenn Sie beginnen Parameter Store zu verwenden. Mithilfe der Ebene mit den Standardparametern können Sie 10.000 Parameter für jeden Parameter AWS-Region in einem erstellen. AWS-Konto Die Inhaltsgröße jedes Parameters darf maximal 4 KB betragen. Standardparameter unterstützen keine Parameterrichtlinien. Für die Nutzung der Standardparameterstufe fallen keine zusätzlichen Gebühren an. Die Auswahl von Standard als Standardstufe bedeutet, dass Parameter Store immer versucht, einen Standardparameter für Anforderungen zu erstellen, die keine Stufe angeben.
- Erweitert — Verwenden Sie die Stufe mit erweiterten Parametern, um maximal 100.000 Parameter für jeden Parameter in einem zu erstellen. AWS-Region AWS-Konto Die Inhaltsgröße jedes Parameters darf maximal 8 KB betragen. Erweiterte Parameter unterstützen Parameterrichtlinien. Um einen Parameter gemeinsam nutzen zu können, muss er sich in der erweiterten Parameterebene befinden. Für die Nutzung der erweiterten Parameterstufe fallen Gebühren an.

Weitere Informationen finden Sie unter AWS Systems Manager -Preisgestaltung für Parameter Store. Die Auswahl von Advanced (Erweitert) als Standardstufe bedeutet, dass Parameter Store immer versucht, einen erweiterten Parameter für Anforderungen zu erstellen, die keine Stufe angeben.

> **Note**
>
> Wenn Sie die erweiterte Parameterstufe auswählen, müssen Sie AWS explizit autorisieren, Ihrem Konto für alle erweiterten Parameter, die Sie erstellen, Gebühren zu verrechnen.

- Intelligent-Tiering – Mit der Option Intelligent-Tiering bestimmt Parameter Store, ob die Standardparameterstufe oder die erweiterte Parameterstufe basierend auf dem Inhalt der Anforderung verwendet werden soll. Wenn Sie beispielsweise einen Befehl ausführen, um einen Parameter mit einem Inhalt unter 4 KB zu erstellen, und der aktuelle Wert AWS-Region in Ihrem AWS-Konto weniger als 10.000 Parameter enthält und Sie keine Parameterrichtlinie angeben, wird ein Standardparameter erstellt. Wenn Sie einen Befehl ausführen, um einen Parameter mit mehr als 4 KB Inhalt zu erstellen, haben Sie bereits mehr als 10.000 Parameter AWS-Region in Ihrem System AWS-Konto, oder wenn Sie eine Parameterrichtlinie angeben, wird ein erweiterter Parameter erstellt.

> **Note**
>
> Wenn Sie Intelligent-Tiering wählen, autorisieren Sie ausdrücklich, Ihr Konto mit allen von Ihnen erstellten erweiterten Parametern AWS zu belasten.

Sie können die Parameter Store-Standardstufeneinstellung jederzeit ändern.

Konfigurieren von Berechtigungen zum Angeben einer Parameter Store-Standardstufe

Stellen Sie sicher, dass Sie in AWS Identity and Access Management (IAM) berechtigt sind, die Standardparameterstufe zu ändern, Parameter Store indem Sie einen der folgenden Schritte ausführen:

- Stellen Sie sicher, dass Sie die `AdministratorAccess`-Richtlinie an Ihre IAM-Entität (z. B. Benutzer, Gruppe oder Rolle) anfügen.
- Stellen Sie sicher, dass Sie über die Berechtigung zum Ändern der Standardstufeneinstellung verfügen, indem Sie die folgenden API-Operationen verwenden:

- GetServiceSetting
- UpdateServiceSetting
- ResetServiceSetting

Gewähren Sie der IAM-Entität die folgenden Berechtigungen, damit ein Benutzer die Standard-Kontingent-Einstellung für Parameter in einer bestimmten AWS-Region in einem AWS-Konto anzeigen und ändern kann.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:servicesetting/ssm/
parameter-store/default-parameter-tier"
        }
    ]
}
```

Administratoren können schreibgeschützte Berechtigungen festlegen, indem sie die folgenden Berechtigungen zuweisen.

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "*"
        }
    ]
}
```

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

  Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center -Benutzerhandbuch.
- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

  Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anleitung unter Eine Rolle für einen externen Identitätsanbieter (Verbund) erstellen im IAM-Benutzerhandbuch.
- IAM-Benutzer:
  - Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Befolgen Sie die Anleitung unter Eine Rolle für einen IAM-Benutzer erstellen im IAM-Benutzerhandbuch.
  - (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter Hinzufügen von Berechtigungen zu einem Benutzer (Konsole) im IAM-Benutzerhandbuch.

Angeben oder Ändern der Parameter Store-Standardstufe mithilfe der Konsole

Das folgende Verfahren zeigt, wie Sie die Systems Manager Manager-Konsole verwenden, um die Standardparameterebene für das aktuelle AWS-Konto und anzugeben oder zu ändern AWS-Region.

> **Tip**
>
> Wenn Sie noch keinen Parameter erstellt haben, können Sie das AWS Command Line Interface (AWS CLI) oder verwenden, AWS Tools for Windows PowerShell um die Standardparameterebene zu ändern. Weitere Informationen finden Sie unter Angeben oder Ändern der Parameter Store Standardstufe mit dem AWS CLI und Angeben oder Ändern der Parameter Store-Standardstufe (PowerShell).

So legen Sie die Parameter Store-Standardstufe fest oder ändern sie

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie die Registerkarte Einstellungen.
4. Klicken Sie auf Ändern der Standardstufe.
5. Wählen Sie eine der folgenden Optionen aus.

   - Standard
   - Advanced
   - Intelligent-Tiering

   Weitere Informationen zu diesen Optionen finden Sie unter Angeben einer Standardparameterstufe.
6. Überprüfen Sie die Nachricht und klicken Sie dann auf Confirm (Bestätigen).

Wenn Sie die Standardstufeneinstellung später ändern möchten, wiederholen Sie diesen Vorgang und geben Sie eine andere Option für die Standardstufe an.

Angeben oder Ändern der Parameter Store Standardstufe mit dem AWS CLI

Das folgende Verfahren zeigt, wie Sie mit dem AWS CLI die Standardeinstellung für die Parameterebene für das aktuelle AWS-Konto und ändern können AWS-Region.

So geben Sie die Parameter Store-Standardstufe mithilfe der AWS CLI an oder ändern sie

1. Öffnen Sie den AWS CLI und führen Sie den folgenden Befehl aus, um die Standardeinstellung für die Parameterebene für eine bestimmte AWS-Region Zeile zu ändern AWS-Konto.

   ```
   aws ssm update-service-setting --setting-id arn:aws:ssm:region:account-
   id:servicesetting/ssm/parameter-store/default-parameter-tier --setting-value tier-
   option
   ```

   *region*stellt den Bezeichner für eine Region dar AWS Systems Manager, die von AWS-Region unterstützt wird, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

   *tier-option*Zu den Werten gehören `Standard`Advanced, und`Intelligent-Tiering`. Weitere Informationen zu diesen Optionen finden Sie unter Angeben einer Standardparameterstufe.

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl aus, um die aktuellen Standardeinstellungen für den Dienst auf Parameterebene für Parameter Store die aktuelle Version AWS-Konto und anzuzeigen AWS-Region.

   ```
   aws ssm get-service-setting --setting-id arn:aws:ssm:region:account-
   id:servicesetting/ssm/parameter-store/default-parameter-tier
   ```

   Die vom System zurückgegebenen Informationen ähneln den Folgenden.

   ```
   {
       "ServiceSetting": {
           "SettingId": "/ssm/parameter-store/default-parameter-tier",
           "SettingValue": "Advanced",
           "LastModifiedDate": 1556551683.923,
           "LastModifiedUser": "arn:aws:sts::123456789012:assumed-role/Administrator/
   Jasper",
   ```

```
        "ARN": "arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/parameter-
store/default-parameter-tier",
        "Status": "Customized"
    }
}
```

Wenn Sie die Standardstufeneinstellung erneut ändern möchten, wiederholen Sie diesen Vorgang und geben Sie eine andere `SettingValue`-Option an.

Angeben oder Ändern der Parameter Store-Standardstufe (PowerShell)

Das folgende Verfahren zeigt, wie Sie die Tools für Windows verwenden, PowerShell um die Standardeinstellung für die Parameterstufe für ein bestimmtes Konto AWS-Region in einem Amazon Web Services Services-Konto zu ändern.

Um die Parameter Store Standardstufe anzugeben oder zu ändern, verwenden Sie PowerShell

1. Ändern Sie die Parameter Store Standardstufe in der aktuellen AWS-Konto und AWS-Region mithilfe von AWS -Tools für PowerShell (Tools für PowerShell).

   ```
   Update-SSMServiceSetting -SettingId "arn:aws:ssm:region:account-id:servicesetting/
   ssm/parameter-store/default-parameter-tier" -SettingValue "tier-option" -
   Region region
   ```

   *region*steht für die Kennung einer Region, die von AWS-Region unterstützt wird AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

   *tier-option*Zu den Werten gehören `Standard`Advanced, und`Intelligent-Tiering`. Weitere Informationen zu diesen Optionen finden Sie unter Angeben einer Standardparameterstufe.

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl aus, um die aktuellen Standardeinstellungen für den Dienst auf Parameterebene für Parameter Store die aktuelle Version AWS-Konto und anzuzeigen AWS-Region.

```
Get-SSMServiceSetting -SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/
parameter-store/default-parameter-tier" -Region region
```

*region*stellt den Bezeichner für eine Region dar AWS Systems Manager, die von AWS-Region unterstützt wird, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

Die vom System zurückgegebenen Informationen ähneln den Folgenden.

```
ARN : arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/parameter-store/
default-parameter-tier
LastModifiedDate : 4/29/2019 3:35:44 PM
LastModifiedUser : arn:aws:sts::123456789012:assumed-role/Administrator/Jasper
SettingId        : /ssm/parameter-store/default-parameter-tier
SettingValue     : Advanced
Status           : Customized
```

Wenn Sie die Standardstufeneinstellung erneut ändern möchten, wiederholen Sie diesen Vorgang und geben Sie eine andere `SettingValue`-Option an.

Ändern eines Standardparameters in einen fortgeschrittenen Parameter

Gehen Sie wie folgt vor, um einen vorhandenen Standardparameter in einen erweiterten Parameter zu ändern. Weitere Informationen zum Erstellen eines neuen erweiterten Parameters finden Sie unter Parameter Store-Parameter im Systems Manager erstellen.

So ändern Sie einen Standardparameter in einen erweiterten Parameter

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie einen Parameter aus und klicken Sie dann auf Edit (Bearbeiten).
4. Geben Sie unter Description (Beschreibung) Informationen zu diesem Parameter ein.
5. Wählen Sie Erweitert aus.
6. Geben Sie unter Value (Wert) den Wert dieses Parameters ein. Erweiterte Parameter haben ein maximales Wertlimit von 8 KB.

7. Wählen Sie Änderungen speichern aus.

Parameter Store-Durchsatz erhöhen oder zurücksetzen

Ein steigender Parameter Store Durchsatz erhöht die maximale Anzahl von Transaktionen pro Sekunde (TPS)Parameter Store, die ein Tool in AWS Systems Manager verarbeiten kann. Ein erhöhter Durchsatz ermöglicht Ihnen, Parameter Store mit höheren Volumina zur Unterstützung von Anwendungen und Arbeitslasten zu betreiben, die gleichzeitigen Zugriff auf mehrere Parameter benötigen. Sie können das Kontingent auf der Registerkarte Einstellungen bis zum maximalen Durchsatz erhöhen.

Die Parameter Store Durchsatzeinstellung gilt für alle Transaktionen, die von allen IAM-Benutzern im aktuellen AWS-Konto und erstellt wurden. AWS-Region Die Durchsatzeinstellung gilt für Standard- und erweiterte Parameter.

> **Note**
>
> In der Regel sind Updates sofort in Service Quotas sichtbar. In seltenen Fällen kann es bis zu 24 Stunden dauern, bis ein Update angezeigt wird.

Weitere Informationen zum maximalen Durchsatzstandard und maximalen Limits finden Sie unter AWS Systems Manager -Endpunkte und -Kontingente.

Wenn Sie das Durchsatzkontingent erhöhen, wird Ihnen eine Gebühr berechnet. AWS-Konto Weitere Informationen finden Sie unter AWS Systems Manager – Preise.

Themen

- Konfiguration von Berechtigungen zur Änderung des Durchsatzes Parameter Store
- Den Durchsatz mithilfe der Konsole erhöhen oder zurücksetzen
- Erhöhen oder Zurücksetzen des Durchsatzes mit dem AWS CLI
- Durchsatz erhöhen oder zurücksetzen () PowerShell

Konfiguration von Berechtigungen zur Änderung des Durchsatzes Parameter Store

Stellen Sie sicher, dass Sie in IAM über die Berechtigung verfügen, den Parameter Store-Durchsatz zu verändern, indem Sie einen der folgenden Schritte ausführen:

- Stellen Sie sicher, dass die `AdministratorAccess`-Richtlinie Ihrer IAM-Entität (z. B. Benutzer, Gruppe oder Rolle) angefügt ist.
- Stellen Sie sicher, dass Sie über die Berechtigung zum Ändern des Servicedurchsatzes verfügen, indem Sie die folgenden API-Operationen verwenden:
  - GetServiceSetting
  - UpdateServiceSetting
  - ResetServiceSetting

Gewähren Sie der IAM-Entität die folgenden Berechtigungen, damit ein Benutzer die Durchsatzeinstellung für Parameter in einer bestimmten AWS-Region in einem AWS-Konto anzeigen und ändern kann.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:servicesetting/ssm/
parameter-store/high-throughput-enabled"
        }
    ]
}
```

Administratoren können schreibgeschützte Berechtigungen festlegen, indem sie die folgenden Berechtigungen zuweisen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetServiceSetting"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ssm:ResetServiceSetting",
                "ssm:UpdateServiceSetting"
            ],
            "Resource": "*"
        }
    ]
}
```

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

  Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center -Benutzerhandbuch.
- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

  Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anleitung unter Eine Rolle für einen externen Identitätsanbieter (Verbund) erstellen im IAM-Benutzerhandbuch.
- IAM-Benutzer:
  - Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Befolgen Sie die Anleitung unter Eine Rolle für einen IAM-Benutzer erstellen im IAM-Benutzerhandbuch.
  - (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter Hinzufügen von Berechtigungen zu einem Benutzer (Konsole) im IAM-Benutzerhandbuch.

Den Durchsatz mithilfe der Konsole erhöhen oder zurücksetzen

Das folgende Verfahren zeigt, wie Sie mithilfe von Systems Manager die Anzahl der Transaktionen pro Sekunde, die Parameter Store verarbeiten kann, für das aktuelle AWS-Konto und die AWS-Region erhöhen können. Außerdem erfahren Sie, wie Sie zu den Standardeinstellungen zurückkehren, wenn Sie die Datenrate nicht mehr benötigen oder keine weiteren Gebühren anfallen möchten.

In der Konsole können Sie den Parameter Store-Durchsatz wie folgt erhöhen oder zurücksetzen:

> **Tip**
>
> Wenn Sie noch keinen Parameter erstellt haben, können Sie das AWS Command Line Interface (AWS CLI) oder verwenden, um den Durchsatz AWS Tools for Windows PowerShell zu erhöhen. Weitere Informationen finden Sie unter Erhöhen oder Zurücksetzen des Durchsatzes mit dem AWS CLI und Durchsatz erhöhen oder zurücksetzen () PowerShell.

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie die Registerkarte Einstellungen.
4. Um den Durchsatz zu erhöhen, wählen Sie Limit festlegen.

   –oder–

   Um zum Standardlimit zurückzukehren, wählen Sie Limit zurücksetzen.
5. Wenn Sie das Limit erhöhen, gehen Sie wie folgt vor:

   - Aktivieren Sie das Kontrollkästchen Ich akzeptiere, dass durch das Ändern dieser Einstellung Gebühren auf meinem AWS-Konto anfallen.
   - Wählen Sie Set limit (Limit festlegen) aus.

   –oder–

   Wenn Sie das Limit auf die Standardeinstellung zurücksetzen, gehen Sie wie folgt vor:

- Aktivieren Sie das Kontrollkästchen Ich akzeptiere, dass das Zurücksetzen auf das Standarddurchsatzlimit dazu führt, dass Parameter Store weniger Transaktionen pro Sekunde verarbeitet.
- Wählen Sie Limit erneut festlegen aus.

Erhöhen oder Zurücksetzen des Durchsatzes mit dem AWS CLI

Das folgende Verfahren zeigt, wie Sie mit dem AWS CLI die Anzahl der Transaktionen pro Sekunde erhöhen Parameter Store können, die für den aktuellen AWS-Konto und verarbeitet werden können. AWS-Region Sie können auch das Standardlimit wiederherstellen.

Um den Parameter Store Durchsatz zu erhöhen, verwenden Sie AWS CLI

1. Öffnen Sie den AWS CLI und führen Sie den folgenden Befehl aus, um die Anzahl der Transaktionen pro Sekunde zu erhöhen, die im aktuellen AWS-Konto und verarbeitet werden Parameter Store können AWS-Region.

   ```
   aws ssm update-service-setting --setting-id arn:aws:ssm:region:account-
   id:servicesetting/ssm/parameter-store/high-throughput-enabled --setting-value true
   ```

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl aus, um die aktuellen Durchsatz-Serviceeinstellungen für Parameter Store im aktuellen AWS-Konto und anzuzeigen AWS-Region.

   ```
   aws ssm get-service-setting --setting-id arn:aws:ssm:region:account-
   id:servicesetting/ssm/parameter-store/high-throughput-enabled
   ```

   Die vom System zurückgegebenen Informationen ähneln den Folgenden:

   ```
   {
       "ServiceSetting": {
           "SettingId": "/ssm/parameter-store/high-throughput-enabled",
           "SettingValue": "true",
           "LastModifiedDate": 1556551683.923,
           "LastModifiedUser": "arn:aws:sts::123456789012:assumed-role/Administrator/
   Jasper",
           "ARN": "arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/parameter-
   store/high-throughput-enabled",
   ```

```
        "Status": "Customized"
    }
}
```

Wenn Sie den erhöhten Durchsatz nicht mehr benötigen oder Kosten vermeiden wollen, können Sie die Standardeinstellungen wiederherstellen. Um Ihre Einstellungen wiederherzustellen, führen Sie den folgenden Befehl aus.

```
aws ssm reset-service-setting --setting-id arn:aws:ssm:region:account-
id:servicesetting/ssm/parameter-store/high-throughput-enabled
```

```
{
    "ServiceSetting": {
        "SettingId": "/ssm/parameter-store/high-throughput-enabled",
        "SettingValue": "false",
        "LastModifiedDate": 1555532818.578,
        "LastModifiedUser": "System",
        "ARN": "arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/parameter-store/
high-throughput-enabled",
        "Status": "Default"
    }
}
```

Durchsatz erhöhen oder zurücksetzen () PowerShell

Das folgende Verfahren zeigt, wie Sie mithilfe der Tools für Windows PowerShell die Anzahl der Transaktionen pro Sekunde erhöhen, die für den aktuellen AWS-Konto und verarbeitet Parameter Store werden können. AWS-Region Sie können auch das Standardlimit wiederherstellen.

Um den Parameter Store Durchsatz zu erhöhen, verwenden Sie PowerShell

1. Erhöhen Sie den Parameter Store Durchsatz im aktuellen AWS-Konto und AWS-Region mithilfe von AWS -Tools für PowerShell (Tools für PowerShell).

   ```
   Update-SSMServiceSetting -SettingId "arn:aws:ssm:region:account-id:servicesetting/
   ssm/parameter-store/high-throughput-enabled" -SettingValue "true" -Region region
   ```

   Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl aus, um die aktuellen Durchsatz-Serviceeinstellungen für Parameter Store die aktuelle Version AWS-Konto und anzuzeigen AWS-Region.

```
Get-SSMServiceSetting -SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/
parameter-store/high-throughput-enabled" -Region region
```

Die von den Systemen zurückgegebenen Informationen ähneln den Folgenden:

```
ARN              : arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/parameter-
store/high-throughput-enabled
LastModifiedDate : 4/29/2019 3:35:44 PM
LastModifiedUser : arn:aws:sts::123456789012:assumed-role/Administrator/Jasper
SettingId        : /ssm/parameter-store/high-throughput-enabled
SettingValue     : true
Status           : Customized
```

Wenn Sie den erhöhten Durchsatz nicht mehr benötigen oder Kosten vermeiden wollen, können Sie die Standardeinstellungen wiederherstellen. Um Ihre Einstellungen wiederherzustellen, führen Sie den folgenden Befehl aus.

```
Reset-SSMServiceSetting -SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/
parameter-store/high-throughput-enabled" -Region region
```

Die vom System zurückgegebenen Informationen ähneln den Folgenden:

```
ARN              : arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/parameter-
store/high-throughput-enabled
LastModifiedDate : 4/17/2019 8:26:58 PM
LastModifiedUser : System
SettingId        : /ssm/parameter-store/high-throughput-enabled
SettingValue     : false
Status           : Default
```

Einrichten von Benachrichtigungen oder Auslösen von Aktionen basierend auf Parameter Store-Ereignissen

Die Themen in diesem Abschnitt erklären, wie Sie Amazon EventBridge und Amazon Simple Notification Service (Amazon SNS) verwenden, um über Änderungen an AWS Systems Manager-Parametern benachrichtigt zu werden. Sie können eine EventBridge-Regel erstellen, damit Sie

benachrichtigt werden, wenn ein Parameter oder eine Parameterbezeichnungsversion erstellt, aktualisiert oder gelöscht wird. Ereignisse werden auf bestmögliche Weise ausgegeben. Sie können über Änderungen oder den Status der Parameterrichtlinien benachrichtigt werden, wenn beispielsweise ein Parameter abgelaufen ist, bald abläuft oder sich seit einem bestimmten Zeitraum nicht geändert hat.

> **Note**
>
> Parameterrichtlinien sind nur verfügbar für Parameter, die das Kontingent für erweiterte Parameter verwenden. Gebührenpflichtig. Weitere Informationen erhalten Sie unter Zuweisen von Parameterrichtlinien in Parameter Store und Verwalten von Parameterstufen.

Das Thema in diesem Abschnitt erläutert zudem, wie Sie andere Aktionen auf einem Ziel anhand bestimmter Parameter-Ereignisse auslösen. So können Sie z. B. eine AWS Lambda-Funktion ausführen, die einen Parameter automatisch neu erstellt, wenn dieser abgelaufen ist oder gelöscht wird. Außerdem können Sie eine Benachrichtigung einrichten, die eine Lambda-Funktion aufruft, wenn dass Passwort für Ihre Datenbank aktualisiert wird. Die Lambda-Funktion kann ein Zurücksetzen Ihrer Datenbankverbindungen erzwingen oder neue Verbindungen herstellen und dabei das neue Passwort verwenden. EventBridge unterstützt auch das Ausführen von Run Command-Befehlen und Automation-Ausführungen sowie Aktionen in vielen anderen AWS-Services. Run Command und Automation sind beide Tools in AWS Systems Manager. Weitere Informationen finden Sie im Benutzerhandbuch für Amazon EventBridge.

Bevor Sie beginnen

Erstellen Sie alle Ressourcen, die Sie zum Festlegen der Zielaktion für die Regel benötigen, die Sie erstellen möchten. Möchten Sie beispielsweise eine Regel zum Senden von Benachrichtigungen erstellen, müssen Sie zunächst ein Amazon SNS-Thema anlegen. Weitere Informationen finden Sie unter Erste Schritte mit Amazon SNS im Benutzerhandbuch für Amazon Simple Notification Service.

Konfigurieren von EventBridge-Regeln für Parameter und Parameter-Richtlinien

Dieses Thema erklärt Folgendes:

- So erstellen Sie eine EventBridge-Regel, durch die ein Ziel basierend auf Ereignissen aufgerufen wird, die sich auf einem oder mehreren Parametern in Ihrem AWS-Konto ereignen.
- So erstellen Sie EventBridge-Regeln, durch die Ziele basierend auf Ereignissen aufgerufen werden, die sich auf einer oder mehreren Parameter-Richtlinien in Ihrem AWS-Konto ereignen.

Wenn Sie einen erweiterten Parameter erstellen, geben Sie an, wann ein Parameter abläuft, wann eine Benachrichtigung gesendet wird, dass ein Parameter abläuft, und wie lange gewartet werden soll, bevor eine Benachrichtigung darüber gesendet wird, dass ein Parameter sich nicht verändert hat. Sie richten die Benachrichtigungen für diese Ereignisse anhand der folgenden Schritte ein. Weitere Informationen erhalten Sie unter Zuweisen von Parameterrichtlinien in Parameter Store und Verwalten von Parameterstufen.

So konfigurieren Sie eine EventBridge-Regel für einen Systems-Manager-Parameter oder eine Parameter-Richtlinie

1. Öffnen Sie die Amazon-EventBridge-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Rules (Regeln) und anschließend Create rule (Regel erstellen) aus.

   –oder–

   Wenn die Amazon EventBridge-Homepage zuerst geöffnet wird, wählen Sie Create rule (Regel erstellen).
3. Geben Sie einen Namen und eine Beschreibung für die Regel ein.

   Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben Region und auf demselben Event Bus haben.
4. Wählen Sie für Event Bus den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel bei übereinstimmenden Ereignissen, die von Ihrem eigenen AWS-Konto stammen, ausgelöst wird, wählen Sie Standard. Wenn ein AWS-Service in Ihrem Konto ein Ereignis ausgibt, wird es stets an den standardmäßigen Event Bus Ihres Kontos weitergeleitet.
5. Lassen Sie für Rule type (Regeltyp) die Standardoption Rule with an event pattern (Regel mit einem Ereignismuster) ausgewählt.
6. Wählen Sie Weiter aus.
7. Lassen Sie für Ereignisquelle die Standardoption AWS-Ereignisse oder EventBridge-Partnerereignisse ausgewählt. Sie können den Abschnitt Beispielereignis überspringen.
8. Gehen Sie bei Event pattern (Ereignismuster) wie folgt vor:

   - Wählen Sie Custom patterns (JSON editor) (Benutzerdefinierte Muster (JSON-Editor)) aus.

- Fügen Sie für Event pattern (Ereignismuster) einen der folgenden Inhalte in das Feld ein, je nachdem, ob Sie eine Regel für einen Parameter oder eine Parameter-Richtlinie erstellen:

  Parameter

```
{
    "source": [
        "aws.ssm"
    ],
    "detail-type": [
        "Parameter Store Change"
    ],
    "detail": {
        "name": [
            "parameter-1-name",
            "/parameter-2-name/level-2",
            "/parameter-3-name/level-2/level-3"
        ],
        "operation": [
            "Create",
            "Update",
            "Delete",
            "LabelParameterVersion"
        ]
    }
}
```

  Parameter policy

```
{
    "source": [
        "aws.ssm"
    ],
    "detail-type": [
        "Parameter Store Policy Action"
    ],
    "detail": {
        "parameter-name": [
            "parameter-1-name",
            "/parameter-2-name/level-2",
            "/parameter-3-name/level-2/level-3"
        ],
        "policy-type": [
```

```
            "Expiration",
            "ExpirationNotification",
            "NoChangeNotification"
        ]
    }
}
```

- Ändern Sie den Inhalt für die Parameter und die Operationen, auf die Sie reagieren möchten, wie in den folgenden Beispielen gezeigt.

  Parameter

  In diesem Beispiel wird eine Aktion ausgeführt, wenn einer der Parameter namens `/Oncall` und `/Project/Teamlead` aktualisiert wird:

```
{
    "source": [
        "aws.ssm"
    ],
    "detail-type": [
        "Parameter Store Change"
    ],
    "detail": {
        "name": [
            "/Oncall",
            "/Project/Teamlead"
        ],
        "operation": [
            "Update"
        ]
    }
}
```

  Parameter policy

  In diesem Beispiel wird immer dann eine Aktion ausgeführt, wenn der Parameter mit dem Namen `/OncallDuties` abläuft und gelöscht wird:

```
{
    "source": [
        "aws.ssm"
    ],
    "detail-type": [
```

```
        "Parameter Store Policy Action"
    ],
    "detail": {
        "parameter-name": [
            "/OncallDuties"
        ],
        "policy-type": [
            "Expiration"
        ]
    }
}
```

9. Wählen Sie Weiter aus.
10. Für Target 1 (Ziel 1) wählen Sie einen Zieltyp und eine unterstützte Ressource aus. Wenn Sie beispielsweise SNS-Thema auswählen, treffen Sie eine Auswahl für Topic (Thema). Wenn Sie CodePipeline wählen geben Sie einen Pipeline-ARN für Pipeline ARN ein. Geben Sie bei Bedarf zusätzliche Konfigurationswerte an.

    > **Tip**
    >
    > Wählen Sie Add another target (Weiteres Ziel hinzufügen), wenn Sie zusätzliche Ziele für die Regel benötigen.

11. Wählen Sie Weiter aus.
12. (Optional) Geben Sie ein oder mehrere Tags für die Regel ein. Weitere Informationen finden Sie unter Amazon-EventBridge-Tags im Benutzerhandbuch zu Amazon EventBridge.
13. Wählen Sie Weiter aus.
14. Wählen Sie Regel erstellen aus.

Weitere Informationen

- Verwenden von Parameterbezeichnungen für die einfache Aktualisierung der Konfiguration über mehrere Umgebungen hinweg
- Tutorial: Verwenden von EventBridge zum Weiterleiten von Ereignissen an AWS Systems Manager Run Command im Benutzerhandbuch zu Amazon EventBridge
- Tutorial: Einrichten von AWS Systems Manager-Automatisierung als EventBridge-Ziel im Benutzerhandbuch zu Amazon EventBridge

Arbeiten mit Parameter Store

In diesem Abschnitt wird beschrieben, wie Sie Parameter organisieren, erstellen und markieren und verschiedene Versionen von Parametern erstellen.

Sie können die AWS Systems Manager-Konsole, die AWS Command Line Interface (AWS CLI), die AWS -Tools für PowerShell und die AWS SDKs verwenden, um Parameter zu erstellen und mit ihnen zu arbeiten. Weitere Informationen zu Parametern finden Sie unter Was ist ein Parameter?.

Themen

- Parameter Store-Parameter im Systems Manager erstellen
- Suchen nach Parameter Store-Parametern im Systems Manager
- Zuweisen von Parameterrichtlinien in Parameter Store
- Arbeiten mit Parameterhierarchien in Parameter Store
- Zugriff auf Parameter Store-API-Vorgänge verhindern
- Arbeiten mit Parameterbezeichnungen in Parameter Store
- Arbeiten mit Parameterversionen in Parameter Store
- Arbeiten mit gemeinsam genutzten Parametern in Parameter Store
- Arbeiten mit Parametern in Parameter Store mithilfe von Run Command-Befehlen
- Verwendung der nativen Parameterunterstützung Parameter Store für Amazon Machine Image IDs
- Löschen von Parametern in Parameter Store

Parameter Store-Parameter im Systems Manager erstellen

Verwenden Sie die Informationen in den folgenden Themen, um Ihnen beim Erstellen von Systems Manager-Parametern über die AWS Systems Manager-Konsole, der AWS Command Line Interface (AWS CLI) oder den AWS Tools for Windows PowerShell (Tools for Windows PowerShell) zu helfen.

Dieser Abschnitt zeigt, wie Sie Parameter mit Parameter Store in einer Testumgebung erstellen, speichern und ausführen können. Es zeigt auch, wie Sie Parameter Store mit anderen Systems-Manager-Tools in AWS-Services verwenden. Weitere Informationen finden Sie unter Was ist ein Parameter?.

Informationen zu Anforderungen und Einschränkungen für Parameternamen

Die Informationen in diesem Thema sind hilfreich bei der Angabe gültiger Werte für Parameternamen, wenn Sie einen Parameter erstellen.

Diese Informationen ergänzen die Details im Thema PutParameter in der AWS Systems Manager API-Referenz, das auch Informationen zu den Werten AllowedPattern, Description, KeyId, Overwrite, Type und Value bereitstellt.

Die Anforderungen und Einschränkungen für Parameternamen umfassen Folgendes:

- Berücksichtigung der Groß-/Kleinschreibung: Bei Parameternamen werden Groß- und Kleinschreibung berücksichtigt.
- Leerstellen: Parameternamen dürfen keine Leerzeichen enthalten.
- Gültige Zeichen: Parameternamen können nur die folgenden Symbole und Buchstaben enthalten: `a-zA-Z0-9_.-`

  Darüber hinaus wird der Schrägstrich (/) verwendet, um Hierarchien in Parameternamen zu beschreiben. Zum Beispiel: `/Dev/Production/East/Project-ABC/MyParameter`
- Gültiges AMI-Format: Wenn Sie `aws:ec2:image` als Datentyp für einen `String`-Parameter wählen, muss die eingegebene ID für das AMI-ID-Format `ami-12345abcdeEXAMPLE` validiert werden.
- Vollständig qualifiziert: Wenn Sie einen Parameter in einer Hierarchie anlegen oder darauf verweisen, müssen Sie einen vorangehenden Schrägstrich (/) einfügen. Wenn Sie auf einen Parameter verweisen, der Teil einer Hierarchie ist, müssen Sie den gesamten Hierarchiepfad einschließlich des ersten Schrägstrichs (/) angeben.
  - Vollständig qualifizierte Parameternamen: `MyParameter1`, `/MyParameter2`, `/Dev/Production/East/Project-ABC/MyParameter`
  - Nicht vollständig qualifizierter Parametername: `MyParameter3/L1`
- Länge: Die maximale Länge für einen Parameternamen, den Sie angeben, beträgt 1011 Zeichen. Dazu gehören die Zeichen im ARN, die vor dem von Ihnen angegebenen Namen stehen, z. B. die 45 Zeichen in `arn:aws:ssm:us-east-2:111122223333:parameter/`.
- Präfixe: Einem Parameternamen darf kein „`aws`“- oder „`ssm`„“-Präfix vorangestellt werden (ohne Berücksichtigung der Groß-/Kleinschreibung). Beispiel: Versuche, Parameter mit den folgenden Namen zu erstellen, schlagen mit einer Ausnahme fehl:
  - `awsTestParameter`
  - `SSM-testparameter`
  - `/aws/testparam1`

> **Note**
>
> Wenn Sie einen Parameter in einem SSM-Dokument, -Befehl oder -Skript angeben, schließen Sie `ssm` als Teil der Syntax ein. Zum Beispiel {{ssm:*parameter-name*}} und {{ ssm:*parameter-name* }}, z. B. `{{ssm:MyParameter}}` und `{{ ssm:MyParameter }}`.

- Eindeutigkeit: Ein Parametername muss innerhalb einer AWS-Regioneindeutig sein. Systems Manager behandelt beispielsweise die folgenden Parameter als separate Parameter, wenn sie sich in derselben Region befinden:
  - `/Test/TestParam1`
  - `/TestParam1`

  Die folgenden Beispiele sind ebenfalls eindeutig:
  - `/Test/TestParam1/Logpath1`
  - `/Test/TestParam1`

  Die folgenden Beispiele sind, sofern sie sich in derselben Region befinden, jedoch nicht eindeutig:
  - `/TestParam1`
  - `TestParam1`
- Hierarchietiefe: Wenn Sie eine Parameterhierarchie angeben, darf die Hierarchie maximal fünfzehn Ebenen tief sein. Sie können auf jeder Ebene der Hierarchie einen Parameter definieren. Die beiden folgenden Beispiele zeigen strukturell gültige Parameter:
  - `/Level-1/L2/L3/L4/L5/L6/L7/L8/L9/L10/L11/L12/L13/L14/parameter-name`
  - `parameter-name`

  Der Versuch, den folgenden Parameter zu erstellen, löst eine `HierarchyLevelLimitExceededException`-Ausnahme aus:
  - `/Level-1/L2/L3/L4/L5/L6/L7/L8/L9/L10/L11/L12/L13/L14/L15/L16/parameter-name`

> **Important**
>
> Wenn ein Benutzer Zugriff auf einen Pfad hat, kann er auf alle Ebenen dieses Pfads zugreifen. Wenn ein Benutzer beispielsweise die Berechtigung für den Zugriff auf den Pfad

`/a` besitzt, dann kann er auch auf `/a/b` zugreifen. Selbst wenn einem Benutzer in AWS Identity and Access Management (IAM) der Zugriff auf den Parameter `/a/b` ausdrücklich verweigert wurde, kann er dennoch die API-Operation GetParametersByPath rekursiv für `/a` aufrufen und `/a/b` anzeigen.

Themen

- Erstellen eines Parameter Store-Paramers mithilfe der Konsole
- Erstellen eines Parameter Store-Parameters mithilfe der AWS CLI
- Einen Parameter Store Parameter mit Tools für Windows erstellen PowerShell

Erstellen eines Parameter Store-Paramers mithilfe der Konsole

Sie können die AWS Systems Manager Konsole verwenden, um `SecureString` Parametertypen und -typen zu erstellen und auszuführen`String`. `StringList` Warten Sie nach dem Löschen eines Parameters mindestens 30 Sekunden, um einen Parameter mit dem gleichen Namen zu erstellen.

> **Note**
>
> Parameter sind nur dort verfügbar AWS-Region , wo sie erstellt wurden.

Das folgende Verfahren führt Sie durch die Schritte zum Erstellen eines Parameters mithilfe der Parameter Store-Konsole. Sie können `String`-, `StringList`- und `SecureString`-Parametertypen über die Konsole erstellen.

So erstellen Sie einen Parameter

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie Parameter erstellen aus.
4. Geben Sie in das Feld Name (Name) eine Hierarchie und einen Namen ein. Geben Sie z. B. ei **`/Test/helloWorld`**.

   Weitere Informationen zu Parameterhierarchien finden Sie unter Arbeiten mit Parameterhierarchien in Parameter Store.

5. Geben Sie im Feld Description (Beschreibung) eine Beschreibung ein, anhand der dieser Parameters als Test-Parameter erkannt werden kann.
6. Wählen Sie für Parameter tier (Parameterstufe) entweder Standard oder Advanced (Erweitert) aus. Weitere Informationen zu erweiterten Parameter finden Sie unter Verwalten von Parameterstufen.
7. Wählen Sie als Typ die Option Zeichenfolge StringList, oder aus SecureString.

   - Wenn Sie String (Zeichenfolge) wählen, wird das Feld Data type (Datentyp) angezeigt. Wenn Sie einen Parameter für die Ressourcen-ID für ein Amazon Machine Image (AMI) erstellen, wählen Sie `aws:ec2:image` aus. Behalten Sie andernfalls die Standardeinstellung `text` bei.
   - Wenn Sie möchten SecureString, wird das Feld KMS-Schlüssel-ID angezeigt. Wenn Sie keine AWS Key Management Service AWS KMS key ID, keinen AWS KMS key Amazon-Ressourcennamen (ARN), einen Aliasnamen oder einen Alias-ARN angeben`alias/aws/ssm`, verwendet das System den Von AWS verwalteter Schlüssel für Systems Manager. Wenn Sie diesen Schlüssel nicht verwenden möchten, können Sie einen kundenverwalteten Schlüssel verwenden. Weitere Informationen über Von AWS verwaltete Schlüssel und kundenverwaltete Schlüssel finden Sie unter AWS Key Management Service -Konzepte im AWS Key Management Service -Entwicklerhandbuch. Weitere Informationen Parameter Store zur AWS KMS Verschlüsselung finden Sie unter AWS Systems ManagerParameter StoreAnwendungsmöglichkeiten AWS KMS.

     > **Important**
     >
     > Parameter Store unterstützt nur KMS-Schlüssel zur symmetrischen Verschlüsselung. Sie können keinen KMS-Schlüssel zur asymmetrischen Verschlüsselung verwenden, um Ihre Parameter zu verschlüsseln. Wie Sie feststellen, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, erfahren Sie unter Erkennen symmetrischer und asymmetrischer Schlüssel im AWS Key Management Service -Entwicklerhandbuch.

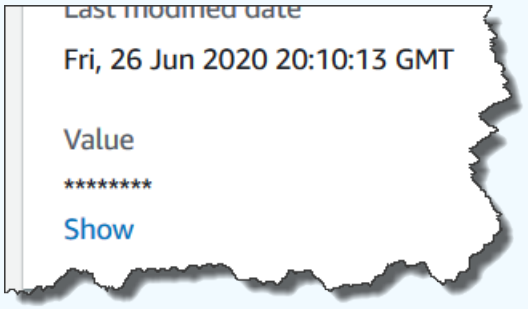
   - Wenn Sie unter Verwendung des Parameters `key-id` mit einem kundenverwalteten Schlüssel-Aliasnamen oder -Alias-ARN in der Konsole einen `SecureString`-Parameter erstellen, müssen Sie vor dem Alias das Präfix `alias/` angeben. Nachfolgend ein ARN-Beispiel.

     ```
     arn:aws:kms:us-east-2:123456789012:alias/abcd1234-ab12-cd34-ef56-abcdeEXAMPLE
     ```

     Im Folgenden finden Sie ein Beispiel für einen Aliasnamen.

```
alias/MyAliasName
```

8. Geben Sie im Feld Value (Wert) einen Wert ein. Geben Sie beispielsweise **`This is my first parameter`** oder **`ami-0dbf5ea29aEXAMPLE`** ein.

> **Note**
>
> Parameter können nicht referenziert oder in den Werten anderer Parameter verschachtelt werden. Sie können `{{}}` oder `{{ssm:parameter-name}}` nicht in einen Parameterwert aufnehmen.
> Wenn Sie sich dafür entscheiden SecureString, wird der Wert des Parameters standardmäßig maskiert („******“), wenn Sie ihn später auf der Registerkarte „Parameterübersicht“ anzeigen, wie in der folgenden Abbildung dargestellt. Klicken Sie auf Anzeigen, um den Parameterwert anzuzeigen.
>
> 
>

9. (Optional) Wenden Sie im Bereich Tags ein oder mehrere Tag-Schlüssel-Wert-Paare auf den Parameter an.

   Tags sind optionale Metadaten, die Sie einer Ressource zuweisen. Mithilfe von Tags können Sie eine Ressource unterschiedlich kategorisieren, beispielsweise nach Zweck, Besitzer oder Umgebung. Sie können beispielsweise einen Systems Manager-Parameter markieren, um den Typ der Ressource, für die er gilt, die Umgebung oder den Typ der Konfigurationsdaten, auf die vom Parameter verwiesen wird, zu identifizieren. In diesem Fall können Sie die folgenden Schlüssel-Wert-Paare angeben:

   - `Key=Resource,Value=S3bucket`
   - `Key=OS,Value=Windows`
   - `Key=ParameterType,Value=LicenseKey`

10. Wählen Sie Parameter erstellen aus.

11. Wählen Sie in der Liste der Parameter den Namen des Parameters aus, den Sie gerade erstellt haben. Überprüfen Sie die Details auf der Registerkarte Overview. Wenn Sie einen `SecureString`-Parameter erstellt haben, wählen Sie Show aus, um die unverschlüsselten Werte anzuzeigen.

> **Note**
>
> Sie können einen erweiterten Parameter nicht in einen Standardparameter ändern. Wenn Sie einen erweiterten Parameter nicht mehr benötigen oder verhindern wollen, dass weitere Gebühren dafür anfallen, löschen Sie ihn und erstellen Sie ihn als Standardparameter neu.

Erstellen eines Parameter Store-Parameters mithilfe der AWS CLI

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um `SecureString` Parametertypen `StringStringList`, und zu erstellen. Warten Sie nach dem Löschen eines Parameters mindestens 30 Sekunden, um einen Parameter mit dem gleichen Namen zu erstellen.

Parameter können nicht referenziert oder in den Werten anderer Parameter verschachtelt werden. Sie können `{{}}` oder `{{ssm:parameter-name}}` nicht in einen Parameterwert aufnehmen.

> **Note**
>
> Parameter sind nur dort verfügbar AWS-Region , wo sie erstellt wurden.

Themen

- Erstellen eines String Parameters mit dem AWS CLI
- Erstellen eines StringList Parameters mit dem AWS CLI
- Erstellen eines SecureString Parameters mit dem AWS CLI
- Erstellen eines mehrzeiligen Parameters mit dem AWS CLI

Erstellen eines **`String`** Parameters mit dem AWS CLI

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um einen Parameter vom `String`-Typ zu erstellen. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen.

   Linux & macOS

   ```
   aws ssm put-parameter \
       --name "parameter-name" \
       --value "parameter-value" \
       --type String \
       --tags "Key=tag-key,Value=tag-value"
   ```

   Windows

   ```
   aws ssm put-parameter ^
       --name "parameter-name" ^
       --value "parameter-value" ^
       --type String ^
       --tags "Key=tag-key,Value=tag-value"
   ```

   –oder–

   Führen Sie den folgenden Befehl aus, um einen Parameter zu erstellen, der eine Amazon Machine Image (AMI)-ID als Parameterwert enthält.

   Linux & macOS

   ```
   aws ssm put-parameter \
       --name "parameter-name" \
       --value "an-AMI-id" \
       --type String \
       --data-type "aws:ec2:image" \
       --tags "Key=tag-key,Value=tag-value"
   ```

   Windows

   ```
   aws ssm put-parameter ^
       --name "parameter-name" ^
   ```

```
    --value "an-AMI-id" ^
    --type String ^
    --data-type "aws:ec2:image" ^
    --tags "Key=tag-key,Value=tag-value"
```

Die Option `--name` unterstützt Hierarchien. Weitere Informationen zu Hierarchien finden Sie unter Arbeiten mit Parameterhierarchien in Parameter Store.

Die Option `--data-type` muss nur angegeben werden, wenn Sie einen Parameter erstellen, der eine AMI-ID enthält. Es überprüft, ob es sich bei dem von Ihnen eingegebenen Parameterwert um eine ordnungsgemäß formatierte Amazon Elastic Compute Cloud (Amazon EC2) AMI -ID handelt. Für alle anderen Parameter lautet der Standarddatentyp `text` und Sie können optional einen Wert angeben. Weitere Informationen finden Sie unter Verwendung der nativen Parameterunterstützung Parameter Store für Amazon Machine Image IDs.

> **Important**
>
> Bei Erfolg gibt der Befehl die Versionsnummer des Parameters zurück. Ausnahme: Wenn Sie `aws:ec2:image` als Datentyp angegeben haben, bedeutet eine neue Versionsnummer in der Antwort nicht, dass der Parameterwert bereits validiert wurde. Weitere Informationen finden Sie unter Verwendung der nativen Parameterunterstützung Parameter Store für Amazon Machine Image IDs.

Im folgenden Beispiel werden einem Parameter die Tags zweier Schlüssel-Wert-Paare hinzugefügt.

Linux & macOS

```
aws ssm put-parameter \
    --name parameter-name \
    --value "parameter-value" \
    --type "String" \
    --tags '[{"Key":"Region","Value":"East"},{"Key":"Environment",
 "Value":"Production"}]'
```

Windows

```
aws ssm put-parameter ^
    --name parameter-name ^
    --value "parameter-value" ^
    --type "String" ^
    --tags [{\"Key\":\"Region1\",\"Value\":\"East1\"},{\"Key\":\"Environment1\",
\"Value\":\"Production1\"}]
```

Im folgenden Beispiel wird eine Parameterhierarchie im Namen verwendet, um einen `String`-Klartext-Parameter zu erstellen. Er gibt die Versionsnummer des Parameters zurück. Weitere Informationen zu Parameterhierarchien finden Sie unter Arbeiten mit Parameterhierarchien in Parameter Store.

Linux & macOS

Parameter nicht in einer Hierarchie

```
aws ssm put-parameter \
    --name "golden-ami" \
    --type "String" \
    --value "ami-12345abcdeEXAMPLE"
```

Parameter in einer Hierarchie

```
aws ssm put-parameter \
    --name "/amis/linux/golden-ami" \
    --type "String" \
    --value "ami-12345abcdeEXAMPLE"
```

Windows

Parameter nicht in einer Hierarchie

```
aws ssm put-parameter ^
    --name "golden-ami" ^
    --type "String" ^
    --value "ami-12345abcdeEXAMPLE"
```

Parameter in einer Hierarchie

```
aws ssm put-parameter ^
    --name "/amis/windows/golden-ami" ^
    --type "String" ^
    --value "ami-12345abcdeEXAMPLE"
```

3. Führen Sie den folgenden Befehl aus, um den letzten Parameterwert anzuzeigen und die Details des neuen Parameters zu überprüfen.

```
aws ssm get-parameters --names "/Test/IAD/helloWorld"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "InvalidParameters": [],
    "Parameters": [
        {
            "Name": "/Test/IAD/helloWorld",
            "Type": "String",
            "Value": "My updated parameter value",
            "Version": 2,
            "LastModifiedDate": "2020-02-25T15:55:33.677000-08:00",
            "ARN": "arn:aws:ssm:us-east-2:123456789012:parameter/Test/IAD/
helloWorld"
        }
    ]
}
```

Führen Sie den folgenden Befehl aus, um den Parameterwert zu ändern. Er gibt die Versionsnummer des Parameters zurück.

```
aws ssm put-parameter --name "/Test/IAD/helloWorld" --value "My updated 1st parameter"
 --type String --overwrite
```

Führen Sie den folgenden Befehl aus, um den Verlauf der Parameterwerte anzuzeigen.

```
aws ssm get-parameter-history --name "/Test/IAD/helloWorld"
```

Führen Sie den folgenden Befehl aus, um diesen Parameter in einem Befehl zu verwenden.

```
aws ssm send-command --document-name "AWS-RunShellScript" --parameters '{"commands":
["echo {{ssm:/Test/IAD/helloWorld}}"]}' --targets "Key=instanceids,Values=instance-ids"
```

Führen Sie den folgenden Befehl aus, wenn Sie nur den Parameterwert abrufen möchten.

```
aws ssm get-parameter --name testDataTypeParameter --query "Parameter.Value"
```

Führen Sie den folgenden Befehl aus, wenn Sie nur den Parameterwert mithilfe von `get-parameters` abrufen möchten.

```
aws ssm get-parameters --names "testDataTypeParameter" --query "Parameters[*].Value"
```

Führen Sie den folgenden Befehl aus, um die Metadaten zum Parameter anzuzeigen.

```
aws ssm describe-parameters --filters "Key=Name,Values=/Test/IAD/helloWorld"
```

> **Note**
>
> Der Name muss großgeschrieben werden.

Das System gibt unter anderem folgende Informationen zurück

```
{
    "Parameters": [
        {
            "Name": "helloworld",
            "Type": "String",
            "LastModifiedUser": "arn:aws:iam::123456789012:user/JohnDoe",
            "LastModifiedDate": 1494529763.156,
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        }
    ]
}
```

Erstellen eines StringList Parameters mit dem AWS CLI

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um einen Parameter zu erstellen. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   Linux & macOS

   ```
   aws ssm put-parameter \
       --name "parameter-name" \
       --value "a-comma-separated-list-of-values" \
       --type StringList \
       --tags "Key=tag-key,Value=tag-value"
   ```

   Windows

   ```
   aws ssm put-parameter ^
       --name "parameter-name" ^
       --value "a-comma-separated-list-of-values" ^
       --type StringList ^
       --tags "Key=tag-key,Value=tag-value"
   ```

   > **Note**
   >
   > Bei Erfolg gibt der Befehl die Versionsnummer des Parameters zurück.

   In diesem Beispiel werden die Tags zweier Schlüssel-Wert-Paare an ein Parameter hinzugefügt. (Führen Sie, abhängig von der Art des Betriebssystems auf Ihrem lokalen Computer, einen der folgenden Befehle aus. Die von einer lokalen Windows-Maschine auszuführende Version enthält die Escape-Zeichen ("\"), mit denen Sie den Befehl von Ihrem Befehlszeilen-Tool aus ausführen.)

   Im folgenden Beispiel für `StringList` wird eine Parameterhierarchie verwendet.

Linux & macOS

```
aws ssm put-parameter \
    --name /IAD/ERP/Oracle/addUsers \
    --value "Milana,Mariana,Mark,Miguel" \
    --type StringList
```

Windows

```
aws ssm put-parameter ^
    --name /IAD/ERP/Oracle/addUsers ^
    --value "Milana,Mariana,Mark,Miguel" ^
    --type StringList
```

> **Note**
>
> Die Elemente einer `StringList` müssen durch ein Komma (,) getrennt werden. Sie können keine anderen Satzzeichen oder Sonderzeichen als Escape-Zeichen für Elemente in der Liste verwenden. Verwenden Sie den Typ `String`, wenn ein Parameterwert ein Komma erfordert.

3. Führen Sie den Befehl `get-parameters` aus, um die Details zu einem Parameter zu überprüfen. Zum Beispiel:

```
aws ssm get-parameters --name "/IAD/ERP/Oracle/addUsers"
```

Erstellen eines SecureString Parameters mit dem AWS CLI

Gehen Sie folgendermaßen vor, um einen `SecureString`-Parameter zu erstellen. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen.

> **Important**
>
> Nur der Wert eines `SecureString`-Parameters wird verschlüsselt. Der Name des Parameters, die Beschreibung und andere Eigenschaften sind nicht verschlüsselt.

> **Important**
>
> Parameter Store unterstützt nur KMS-Schlüssel zur symmetrischen Verschlüsselung. Sie können keinen KMS-Schlüssel zur asymmetrischen Verschlüsselung verwenden, um Ihre Parameter zu verschlüsseln. Wie Sie feststellen, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, erfahren Sie unter Erkennen symmetrischer und asymmetrischer Schlüssel im AWS Key Management Service -Entwicklerhandbuch.

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie einen der folgenden Befehle aus, um einen Parameter zu erstellen, der den Datentyp `SecureString` verwendet.

   Linux & macOS

   Erstellen Sie einen **`SecureString`** Parameter mit der Standardeinstellung Von AWS verwalteter Schlüssel

   ```
   aws ssm put-parameter \
       --name "parameter-name" \
       --value "parameter-value" \
       --type "SecureString"
   ```

   Einen **`SecureString`**-Parameter erstellen, der einen vom Kunden verwalteten Schlüssel verwendet

   ```
   aws ssm put-parameter \
       --name "parameter-name" \
       --value "a-parameter-value, for example P@ssW%rd#1" \
       --type "SecureString"
       --tags "Key=tag-key,Value=tag-value"
   ```

   Erstellen Sie einen **`SecureString`** Parameter, der einen benutzerdefinierten AWS KMS Schlüssel verwendet

```
aws ssm put-parameter \
    --name "parameter-name" \
    --value "a-parameter-value, for example P@ssW%rd#1" \
    --type "SecureString" \
    --key-id "your-account-ID/the-custom-AWS KMS-key" \
    --tags "Key=tag-key,Value=tag-value"
```

Windows

Erstellen Sie einen **SecureString** Parameter unter Verwendung der Standardeinstellung Von AWS verwalteter Schlüssel

```
aws ssm put-parameter ^
    --name "parameter-name" ^
    --value "parameter-value" ^
    --type "SecureString"
```

Einen **SecureString**-Parameter erstellen, der einen vom Kunden verwalteten Schlüssel verwendet

```
aws ssm put-parameter ^
    --name "parameter-name" ^
    --value "a-parameter-value, for example P@ssW%rd#1" ^
    --type "SecureString" ^
    --tags "Key=tag-key,Value=tag-value"
```

Erstellen Sie einen **SecureString** Parameter, der einen benutzerdefinierten AWS KMS Schlüssel verwendet

```
aws ssm put-parameter ^
    --name "parameter-name" ^
    --value "a-parameter-value, for example P@ssW%rd#1" ^
    --type "SecureString" ^
    --key-id " ^
    --tags "Key=tag-key,Value=tag-value"account-ID/the-custom-AWS KMS-key"
```

Wenn Sie einen `SecureString`-Parameter erstellen, indem Sie den Von AWS verwalteter Schlüssel -Schlüssel in Ihrem Konto und Ihrer Region verwenden, müssen Sie keinen Wert für den `--key-id`-Parameter angeben.

> **Note**
>
> Um den Ihrem AWS-Konto und AWS KMS key zugewiesenen Wert zu verwenden AWS-Region, entfernen Sie den `key-id` Parameter aus dem Befehl. Weitere Informationen zum Konfigurieren einer Regel in AWS KMS keys finden Sie unter AWS Key Management Service im AWS Key Management Service -Entwicklerhandbuch.

Um einen vom Kunden verwalteten Schlüssel anstelle des Ihrem Konto Von AWS verwalteter Schlüssel zugewiesenen Schlüssels zu verwenden, geben Sie den Schlüssel mithilfe des `--key-id` Parameters an. Der Parameter unterstützt die folgenden KMS-Parameterformate.

- Beispiel für Schlüssel-Amazon-Ressourcenname (ARN):

  `arn:aws:kms:us-east-2:123456789012:key/key-id`

- Beispiel für den Alias-ARN:

  `arn:aws:kms:us-east-2:123456789012:alias/alias-name`

- Beispiel: Key-ID

  `12345678-1234-1234-1234-123456789012`

- Beispiel für den Aliasnamen:

  `alias/MyAliasName`

Sie können einen vom Kunden verwalteten Schlüssel mithilfe der AWS-Managementkonsole oder der AWS KMS API erstellen. Die folgenden AWS CLI Befehle erstellen einen vom Kunden verwalteten Schlüssel in AWS-Region Ihrer aktuellen Version AWS-Konto.

```
aws kms create-key
```

Verwenden Sie einen Befehl im folgenden Format, um einen `SecureString`-Parameter mit dem Schlüssel zu erstellen, den Sie gerade generiert haben.

Im folgenden Beispiel wird ein verschleierter Name (`3l3vat3131`) für einen Passwortparameter und einen AWS KMS key verwendet.

Linux & macOS

```
aws ssm put-parameter \
    --name /Finance/Payroll/3l3vat3131 \
    --value "P@sSwW)rd" \
    --type SecureString \
    --key-id arn:aws:kms:us-
east-2:123456789012:key/1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e
```

Windows

```
aws ssm put-parameter ^
    --name /Finance/Payroll/3l3vat3131 ^
    --value "P@sSwW)rd" ^
    --type SecureString ^
    --key-id arn:aws:kms:us-
east-2:123456789012:key/1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e
```

3. Führen Sie den folgenden Befehl aus, um die Details zu einem Parameter zu überprüfen.

Wenn Sie keinen `with-decryption`-Parameter bzw. den `no-with-decryption`-Parameter angeben, gibt der Befehl eine verschlüsselte GUID zurück.

Linux & macOS

```
aws ssm get-parameters \
    --name "the-parameter-name-you-specified" \
    --with-decryption
```

Windows

```
aws ssm get-parameters ^
    --name "the-parameter-name-you-specified" ^
    --with-decryption
```

4. Führen Sie den folgenden Befehl aus, um die Metadaten zum Parameter anzuzeigen.

   Linux & macOS

   ```
   aws ssm describe-parameters \
       --filters "Key=Name,Values=the-name-that-you-specified"
   ```

   Windows

   ```
   aws ssm describe-parameters ^
       --filters "Key=Name,Values=the-name-that-you-specified"
   ```

5. Führen Sie den folgenden Befehl aus, um den Parameterwert zu ändern, wenn Sie keinen vom Kunden verwalteten AWS KMS key verwenden.

   Linux & macOS

   ```
   aws ssm put-parameter \
       --name "the-name-that-you-specified" \
       --value "a-new-parameter-value" \
       --type "SecureString" \
       --overwrite
   ```

   Windows

   ```
   aws ssm put-parameter ^
       --name "the-name-that-you-specified" ^
       --value "a-new-parameter-value" ^
       --type "SecureString" ^
       --overwrite
   ```

   –oder–

   Führen Sie einen der folgenden Befehle aus, um den Parameterwert zu ändern, wenn Sie einen vom Kunden verwalteten AWS KMS key verwenden.

   Linux & macOS

   ```
   aws ssm put-parameter \
       --name "the-name-that-you-specified" \
   ```

```
    --value "a-new-parameter-value" \
    --type "SecureString" \
    --key-id "the-KMSkey-ID" \
    --overwrite
```

```
aws ssm put-parameter \
    --name "the-name-that-you-specified" \
    --value "a-new-parameter-value" \
    --type "SecureString" \
    --key-id "account-alias/the-KMSkey-ID" \
    --overwrite
```

Windows

```
aws ssm put-parameter ^
    --name "the-name-that-you-specified" ^
    --value "a-new-parameter-value" ^
    --type "SecureString" ^
    --key-id "the-KMSkey-ID" ^
    --overwrite
```

```
aws ssm put-parameter ^
    --name "the-name-that-you-specified" ^
    --value "a-new-parameter-value" ^
    --type "SecureString" ^
    --key-id "account-alias/the-KMSkey-ID" ^
    --overwrite
```

6. Führen Sie den folgenden Befehl aus, um den letzten Parameterwert anzuzeigen.

Linux & macOS

```
aws ssm get-parameters \
    --name "the-name-that-you-specified" \
    --with-decryption
```

Windows

```
aws ssm get-parameters ^
    --name "the-name-that-you-specified" ^
```

```
    --with-decryption
```

7. Führen Sie den folgenden Befehl aus, um den Verlauf der Parameterwerte anzuzeigen.

   Linux & macOS

   ```
   aws ssm get-parameter-history \
       --name "the-name-that-you-specified"
   ```

   Windows

   ```
   aws ssm get-parameter-history ^
       --name "the-name-that-you-specified"
   ```

> **Note**
>
> Sie können einen Parameter mit einem verschlüsselten Wert manuell erstellen. Da der Wert in diesem Fall bereits verschlüsselt ist, müssen Sie den `SecureString`-Parametertyp nicht auswählen. Wenn Sie `SecureString` dennoch auswählen, wird Ihr Parameter zweifach verschlüsselt.

Alle `SecureString`-Werte werden standardmäßig als verschlüsselter Text angezeigt. Um einen `SecureString` Wert zu entschlüsseln, muss ein Benutzer über die Berechtigung zum Aufrufen des API-Vorgangs AWS KMS Decrypt verfügen. Weitere Informationen zur Konfiguration der AWS KMS -Zugriffskontrolle finden Sie unter Authentifizierung und Zugriffskontrolle für AWS KMS im AWS Key Management Service -Entwicklerhandbuch.

> **Important**
>
> Wenn Sie den KMS-Schlüsselalias für den KMS-Schlüssel, der zum Verschlüsseln eines Parameters verwendet wird, ändern, müssen Sie auch den Schlüsselalias aktualisieren, mit dem der Parameter AWS KMS referenziert. Dies gilt nur für den KMS-Schlüsselalias; die Schlüssel-ID, die ein Alias anfügt, bleibt unverändert, es sei denn, Sie löschen den gesamten Schlüssel.

Erstellen eines mehrzeiligen Parameters mit dem AWS CLI

Sie können den verwenden AWS CLI , um einen Parameter mit Zeilenumbrüchen zu erstellen. Verwenden Sie Zeilenumbrüche, um den Text in längere Parameterwerte aufzuteilen, um die Lesbarkeit zu verbessern, oder aktualisieren Sie beispielsweise den Inhalt von Parametern mit mehreren Absätzen für eine Webseite. Sie können den Inhalt in eine JSON-Datei einschließen und die `--cli-input-json`-Option verwenden, indem Sie Zeilenumbruchzeichen wie `\n` verwenden, wie im folgenden Beispiel gezeigt.

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus, um einen mehrzeiligen Parameter zu erstellen.

   Linux & macOS

   ```
   aws ssm put-parameter \
       --name "MultiLineParameter" \
       --type String \
       --cli-input-json file://MultiLineParameter.json
   ```

   Windows

   ```
   aws ssm put-parameter ^
       --name "MultiLineParameter" ^
       --type String ^
       --cli-input-json file://MultiLineParameter.json
   ```

   Im folgenden Beispiel werden die Inhalte der Datei `MultiLineParameter.json` angezeigt.

   ```
   {
       "Value": "<para>Paragraph One</para>\n<para>Paragraph Two</para>
   \n<para>Paragraph Three</para>"
   }
   ```

Der gespeicherte Parameterwert wird wie folgt gespeichert.

```
<para>Paragraph One</para>
<para>Paragraph Two</para>
<para>Paragraph Three</para>
```

Einen Parameter Store Parameter mit Tools für Windows erstellen PowerShell

Sie können AWS Tools for Windows PowerShell sie zum Erstellen von `SecureString` Parametertypen `StringStringList`, und verwenden. Warten Sie nach dem Löschen eines Parameters mindestens 30 Sekunden, um einen Parameter mit dem gleichen Namen zu erstellen.

Parameter können nicht referenziert oder in den Werten anderer Parameter verschachtelt werden. Sie können `{{}}` oder `{{ssm:parameter-name}}` nicht in einen Parameterwert aufnehmen.

> **Note**
>
> Parameter sind nur dort verfügbar AWS-Region , wo sie erstellt wurden.

Themen

- Einen Zeichenkettenparameter erstellen (Tools für Windows PowerShell)
- Einen StringList Parameter erstellen (Tools für Windows PowerShell)
- Einen SecureString Parameter erstellen (Tools für Windows PowerShell)

Einen Zeichenkettenparameter erstellen (Tools für Windows PowerShell)

1. Installieren und konfigurieren Sie die AWS -Tools für PowerShell (Tools für Windows PowerShell), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren des AWS -Tools für PowerShell.
2. Führen Sie den folgenden Befehl aus, um einen Parameter zu erstellen, der einen Klartext-Wert enthält. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   ```
   Write-SSMParameter `
       -Name "parameter-name" `
       -Value "parameter-value" `
       -Type "String"
   ```

   –oder–

Führen Sie den folgenden Befehl aus, um einen Parameter zu erstellen, der eine Amazon Machine Image (AMI)-ID als Parameterwert enthält.

> **Note**
>
> Um einen Parameter mit einem Tag zu erstellen, erstellen Sie den service.model.tag vorher als Variable. Ein Beispiel.
>
> ```
> $tag = New-Object Amazon.SimpleSystemsManagement.Model.Tag
> $tag.Key = "tag-key"
> $tag.Value = "tag-value"
> ```

```
Write-SSMParameter `
    -Name "parameter-name" `
    -Value "an-AMI-id" `
    -Type "String" `
    -DataType "aws:ec2:image" `
    -Tags $tag
```

Die Option `-DataType` muss nur angegeben werden, wenn Sie einen Parameter erstellen, der eine AMI-ID enthält. Für alle anderen Parameter lautet der Standarddatentyp `text`. Weitere Informationen finden Sie unter Verwendung der nativen Parameterunterstützung Parameter Store für Amazon Machine Image IDs.

Im folgenden Beispiel wird eine Parameterhierarchie verwendet.

```
Write-SSMParameter `
    -Name "/IAD/Web/SQL/IPaddress" `
    -Value "99.99.99.999" `
    -Type "String" `
    -Tags $tag
```

3. Führen Sie den folgenden Befehl aus, um die Details zu einem Parameter zu überprüfen.

```
(Get-SSMParameterValue -Name "the-parameter-name-you-specified").Parameters
```

Einen StringList Parameter erstellen (Tools für Windows PowerShell)

1. Installieren und konfigurieren Sie die AWS -Tools für PowerShell (Tools für Windows PowerShell), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren des AWS -Tools für PowerShell.

2. Führen Sie den folgenden Befehl aus, um einen StringList Parameter zu erstellen. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   > **Note**
   >
   > Um einen Parameter mit einem Tag zu erstellen, erstellen Sie den service.model.tag vorher als Variable. Ein Beispiel.
   >
   > ```
   > $tag = New-Object Amazon.SimpleSystemsManagement.Model.Tag
   > $tag.Key = "tag-key"
   > $tag.Value = "tag-value"
   > ```

   ```
   Write-SSMParameter `
       -Name "parameter-name" `
       -Value "a-comma-separated-list-of-values" `
       -Type "StringList" `
       -Tags $tag
   ```

   Bei Erfolg gibt der Befehl die Versionsnummer des Parameters zurück.

   Ein Beispiel.

   ```
   Write-SSMParameter `
       -Name "stringlist-parameter" `
       -Value "Milana,Mariana,Mark,Miguel" `
       -Type "StringList" `
       -Tags $tag
   ```

   > **Note**
   >
   > Die Elemente einer `StringList` müssen durch ein Komma (,) getrennt werden. Sie können keine anderen Satzzeichen oder Sonderzeichen als Escape-Zeichen

für Elemente in der Liste verwenden. Verwenden Sie den Typ `String`, wenn ein Parameterwert ein Komma erfordert.

3. Führen Sie den folgenden Befehl aus, um die Details zu einem Parameter zu überprüfen.

```
(Get-SSMParameterValue -Name "the-parameter-name-you-specified").Parameters
```

Einen SecureString Parameter erstellen (Tools für Windows PowerShell)

Bevor Sie einen `SecureString`-Parameter erstellen, informieren Sie sich über die Voraussetzungen für diese Art von Parameter. Weitere Informationen finden Sie unter Erstellen eines SecureString Parameters mit dem AWS CLI.

Important

Nur der Wert eines `SecureString`-Parameters wird verschlüsselt. Der Name des Parameters, die Beschreibung und andere Eigenschaften sind nicht verschlüsselt.

Important

Parameter Store unterstützt nur KMS-Schlüssel zur symmetrischen Verschlüsselung. Sie können keinen KMS-Schlüssel zur asymmetrischen Verschlüsselung verwenden, um Ihre Parameter zu verschlüsseln. Wie Sie feststellen, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, erfahren Sie unter Erkennen symmetrischer und asymmetrischer Schlüssel im AWS Key Management Service -Entwicklerhandbuch.

1. Installieren und konfigurieren Sie die AWS -Tools für PowerShell (Tools für Windows PowerShell), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren des AWS -Tools für PowerShell.

2. Führen Sie den folgenden Befehl aus, um einen Parameter zu erstellen. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen.

> **Note**
>
> Um einen Parameter mit einem Tag zu erstellen, erstellen Sie zuerst den service.model.tag als Variable. Ein Beispiel.
>
> ```
> $tag = New-Object Amazon.SimpleSystemsManagement.Model.Tag
> $tag.Key = "tag-key"
> $tag.Value = "tag-value"
> ```

```
Write-SSMParameter `
    -Name "parameter-name" `
    -Value "parameter-value" `
    -Type "SecureString"  `
    -KeyId "an AWS KMS key ID, an AWS KMS key ARN, an alias name, or an alias ARN"
 `
    -Tags $tag
```

Bei Erfolg gibt der Befehl die Versionsnummer des Parameters zurück.

> **Note**
>
> Um den Ihrem Konto Von AWS verwalteter Schlüssel zugewiesenen zu verwenden, entfernen Sie den `-KeyId` Parameter aus dem Befehl.

Im folgenden Beispiel wird ein verschleierter Name (3l3vat3131) für einen Passwortparameter und einen Von AWS verwalteter Schlüssel verwendet.

```
Write-SSMParameter `
    -Name "/Finance/Payroll/3l3vat3131" `
    -Value "P@sSwW)rd" `
    -Type "SecureString"`
    -Tags $tag
```

3. Führen Sie den folgenden Befehl aus, um die Details zu einem Parameter zu überprüfen.

```
(Get-SSMParameterValue -Name "the-parameter-name-you-specified" -WithDecryption
 $true).Parameters
```

Alle `SecureString`-Werte werden standardmäßig als verschlüsselter Text angezeigt. Um einen `SecureString` Wert zu entschlüsseln, muss ein Benutzer über die Berechtigung zum Aufrufen des API-Vorgangs AWS KMS Decrypt verfügen. Weitere Informationen zur Konfiguration der AWS KMS -Zugriffskontrolle finden Sie unter Authentifizierung und Zugriffskontrolle für AWS KMS im AWS Key Management Service -Entwicklerhandbuch.

> **Important**
>
> Wenn Sie den KMS-Schlüsselalias für den KMS-Schlüssel, der zum Verschlüsseln eines Parameters verwendet wird, ändern, müssen Sie auch den Schlüsselalias aktualisieren, mit dem der Parameter AWS KMS referenziert. Dies gilt nur für den KMS-Schlüsselalias; die Schlüssel-ID, die ein Alias anfügt, bleibt unverändert, es sei denn, Sie löschen den gesamten Schlüssel.

Suchen nach Parameter Store-Parametern im Systems Manager

Wenn Sie viele Parameter in Ihrem Konto haben, kann es schwierig sein, Informationen zu nur einem oder einigen wenigen Parametern gleichzeitig zu finden. In diesem Fall können Sie Filterwerkzeuge verwenden, um mithilfe von Suchkriterien nach den gewünschten Parametern zu suchen. Sie können die AWS Systems Manager Konsole, die AWS Command Line Interface (AWS CLI), die oder die DescribeParametersAPI verwenden AWS -Tools für PowerShell, um nach Parametern zu suchen.

Themen

- Mithilfe der Konsole nach einem Parameter suchen
- Suche nach einem Parameter mit AWS CLI

Mithilfe der Konsole nach einem Parameter suchen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie das Suchfeld und die gewünschte Suchmethode aus. Zum Beispiel `Type` oder `Name`.

4. Geben Sie Informationen für den ausgewählten Suchtyp an. Zum Beispiel:

   - Wenn Sie nach `Type` suchen, wählen Sie `String`, `StringList` oder `SecureString` aus.
   - Wenn Sie nach `Name` suchen, wählen Sie `contains`, `equals` oder `begins-with` aus und geben Sie den Parameternamen ganz oder teilweise ein.

   > **Note**
   >
   > In der Konsole ist `contains` der Standardsuchtyp für `Name`.

5. Drücken Sie die Eingabetaste.

Die Liste der Parameter wird mit den Ergebnissen Ihrer Suche aktualisiert.

> **Note**
>
> Ihre Suche enthält möglicherweise mehr Ergebnisse, als auf der ersten Ergebnisseite angezeigt werden. Verwenden Sie den Rechtspfeil (>) neben dem Thema der Parameterliste (falls verfügbar), um die nächste Ergebnisseite anzuzeigen.

Suche nach einem Parameter mit AWS CLI

Verwenden Sie den Befehl `describe-parameters`, um Informationen zu einem oder mehreren Parametern in der AWS CLI anzuzeigen.

Die folgenden Beispiele zeigen verschiedene Optionen, mit denen Sie Informationen zu den Parametern in Ihrem anzeigen können AWS-Konto. Weitere Informationen zu diesen Optionen finden Sie unter describe-parameters im AWS Command Line Interface -Handbuch.

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Ersetzen Sie die Beispielwerte in den folgenden Befehlen durch Werte, die in Ihrem Konto erstellten Parametern entsprechen.

Linux & macOS

```
aws ssm describe-parameters \
    --parameter-filters "Key=Name,Values=MyParameterName"
```

Windows

```
aws ssm describe-parameters ^
    --parameter-filters "Key=Name,Values=MyParameterName"
```

> **Note**
>
> Für `describe-parameters` ist `Equals` der Standardsuchtyp für `Name`. In den Parameterfiltern ist die Angabe von `"Key=Name,Values=MyParameterName"` identisch mit der Angabe `"Key=Name,Option=Equals,Values=MyParameterName"`.

```
aws ssm describe-parameters \
    --parameter-filters "Key=Name,Option=Contains,Values=Product"
```

```
aws ssm describe-parameters \
    --parameter-filters "Key=Type,Values=String"
```

```
aws ssm describe-parameters \
    --parameter-filters "Key=Path,Values=/Production/West"
```

```
aws ssm describe-parameters \
    --parameter-filters "Key=Tier,Values=Standard"
```

```
aws ssm describe-parameters \
    --parameter-filters "Key=tag:tag-key,Values=tag-value"
```

```
aws ssm describe-parameters \
```

```
    --parameter-filters "Key=KeyId,Values=key-id"
```

> **Note**
>
> Im letzten Beispiel *key-id* steht es für die ID eines AWS Key Management Service (AWS KMS) -Schlüssels, der zum Verschlüsseln eines in Ihrem Konto erstellten `SecureString` Parameters verwendet wird. Alternativ können Sie eingeben, **alias/aws/ssm** um den AWS KMS Standardschlüssel für Ihr Konto zu verwenden. Weitere Informationen finden Sie unter Erstellen eines SecureString Parameters mit dem AWS CLI.

Bei erfolgreicher Ausführung gibt der Befehl eine Ausgabe zurück, die in etwa wie folgt aussieht:

```
{
    "Parameters": [
        {
            "Name": "/Production/West/Manager",
            "Type": "String",
            "LastModifiedDate": 1573438580.703,
            "LastModifiedUser": "arn:aws:iam::111122223333:user/Mateo.Jackson",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "/Production/West/TeamLead",
            "Type": "String",
            "LastModifiedDate": 1572363610.175,
            "LastModifiedUser": "arn:aws:iam::111122223333:user/Mateo.Jackson",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "/Production/West/HR",
            "Type": "String",
            "LastModifiedDate": 1572363680.503,
            "LastModifiedUser": "arn:aws:iam::111122223333:user/Mateo.Jackson",
            "Version": 1,
            "Tier": "Standard",
```

```
            "Policies": []
        }
    ]
}
```

Zuweisen von Parameterrichtlinien in Parameter Store

Parameterrichtlinien unterstützen Sie bei der Verwaltung einer wachsenden Menge von Parametern, indem Sie einem Parameter bestimmte Kriterien zuweisen können, wie etwa Ablaufdatum oder Time to Live (Gültigkeitsdauer). Parameterrichtlinien sind besonders hilfreich, da sie Sie zwingen, in Parameter Store, einem Tool in AWS Systems Manager, gespeicherte Passwörter und Konfigurationsdaten zu aktualisieren oder zu löschen. Parameter Store bietet folgende Arten von Richtlinien: `Expiration`, `ExpirationNotification` und `NoChangeNotification`.

> **Note**
>
> Verwenden Sie, um Lebenszyklen für die Passwortrotation zu implementieren. AWS Secrets Manager Sie können Datenbankanmeldeinformationen, API-Schlüssel und andere geheime Informationen mit Secrets Manager während ihres gesamten Lebenszyklus mühelos rotieren, verwalten und abfragen. Weitere Informationen finden Sie unter Was ist? AWS Secrets Manager im AWS Secrets Manager Benutzerhandbuch.

Parameter Store erzwingt Parameterrichtlinien durch asynchrone, periodische Scans. Nachdem Sie eine Richtlinie erstellt haben, müssen Sie weitere Aktionen ausführen, um die Richtlinie zu erzwingen. Parameter Store führt die von der Richtlinie definierte Aktion gemäß den von Ihnen angegebenen Kriterien unabhängig aus.

> **Note**
>
> Parameterrichtlinien sind nur verfügbar für Parameter, die das Kontingent für erweiterte Parameter verwenden. Weitere Informationen finden Sie unter Verwalten von Parameterstufen.

Eine Parameterrichtlinie ist ein JSON-Array, wie in der folgenden Tabelle gezeigt. Sie können eine Richtlinie zuweisen, wenn Sie einen neuen erweiterten Parameter erstellen, oder Sie können eine

Richtlinie anwenden, indem Sie einen Parameter aktualisieren. Parameter Store unterstützt die folgenden Arten von Parameterrichtlinien.

Richtlinie	Details	Beispiele
Ablauf	Diese Richtlinie löscht den Parameter. Sie können ein bestimmtes Datum und eine bestimmte Uhrzeit im Format `ISO_INSTANT` oder `ISO_OFFSET_DATE_TIME` angeben. Wenn Sie den Zeitpunkt für das Löschen des Parameters ändern möchten, aktualisieren Sie die Richtlinie. Das Aktualisieren eines Parameters hat keine Auswirkungen auf das Ablaufdatum oder die Uhrzeit der angefügten Richtlinie. Wenn das Ablaufdatum und die Uhrzeit erreicht ist, löscht Parameter Store den Parameter. **Note** Für das Beispiel wird das Format `ISO_INSTANT` verwendet. Sie können auch ein Datum und eine Uhrzeit im Format `ISO_OFFSET_DATE_TIME` angeben. Hier ist ein Beispiel: `2019-11-0`	`{ "Type": "Expiration", "Version": "1.0", "Attributes": { "Timestamp": "2018-12-02T21:34:33.000Z" } }`

Richtlinie	Details	Beispiele
	1T22:13:4 8.87+10:30:00 .	

ExpirationNotification	Diese Richtlinie löst ein Ereignis in Amazon EventBridge (EventBridge) aus, das Sie über den Ablauf informiert. Mithilfe dieser Richtlinie können Sie Benachrichtigungen erhalten, bevor die Ablaufzeit erreicht ist, und zwar in Einheiten von Tagen oder Stunden.	<pre>{
		    "Type": "ExpirationNotification",
		    "Version": "1.0",
		    "Attributes": {
		        "Before": "15",
		        "Unit": "Days"
		    }
		}</pre>

Richtlinie	Details	Beispiele

NoChangeNotification	Diese Richtlinie löst ein Ereignis aus, EventBrid ge wenn ein Parameter für einen bestimmten Zeitraum nicht geändert wurde. Dieser Richtlinientyp ist beispiels weise nützlich, wenn ein Passwort in einem bestimmte n Zeitraum geändert werden muss.
	
	Diese Richtlinie bestimmt anhand des `LastModif iedTime` -Attributs des Parameters, wann eine Benachrichtigung gesendet wird. Wenn Sie einen Parameter ändern oder bearbeiten, setzt das System den Benachrichtigungsz eitraum basierend auf dem neuen Wert für `LastModif iedTime` zurück.
		<pre>{
		    "Type": "NoChange Notification",
		    "Version": "1.0",
		    "Attributes": {
		        "After": "20",
		        "Unit": "Days"
		    }
		}</pre>

Sie können einem Parameter mehrere Richtlinien zuweisen. Sie können beispielsweise `ExpirationNotification` Richtlinien zuweisen`Expiration`, sodass das System ein EventBridge Ereignis auslöst, um Sie über das bevorstehende Löschen eines Parameters zu informieren. Sie können einem Parameter maximal zehn (10) Richtlinien zuweisen.

Das folgende Beispiel zeigt die Anforderungssyntax für eine PutParameterAPI-Anfrage, die einem neuen `SecureString` Parameter mit dem Namen vier Richtlinien zuweist. `ProdDB3`

```
{
    "Name": "ProdDB3",
    "Description": "Parameter with policies",
```

```
    "Value": "P@ssW*rd21",
    "Type": "SecureString",
    "Overwrite": "True",
    "Policies": [
        {
            "Type": "Expiration",
            "Version": "1.0",
            "Attributes": {
                "Timestamp": "2018-12-02T21:34:33.000Z"
            }
        },
        {
            "Type": "ExpirationNotification",
            "Version": "1.0",
            "Attributes": {
                "Before": "30",
                "Unit": "Days"
            }
        },
        {
            "Type": "ExpirationNotification",
            "Version": "1.0",
            "Attributes": {
                "Before": "15",
                "Unit": "Days"
            }
        },
        {
            "Type": "NoChangeNotification",
            "Version": "1.0",
            "Attributes": {
                "After": "20",
                "Unit": "Days"
            }
        }
    ]
}
```

Hinzufügen von Richtlinien zu einem vorhandenen Parameter

Dieser Abschnitt enthält Informationen zum Hinzufügen von Richtlinien zu einem vorhandenen Parameter mithilfe der AWS Systems Manager Konsole, der AWS Command Line Interface (AWS CLI) und AWS Tools for Windows PowerShell . Weitere Informationen zum Erstellen eines neuen

Parameters mit Richtlinien finden Sie unter Parameter Store-Parameter im Systems Manager erstellen.

Themen

- Hinzufügen von Richtlinien zu einem vorhandenen Parameter mit der Konsole
- Hinzufügen von Richtlinien zu einem vorhandenen Parameter mithilfe des AWS CLI
- Hinzufügen von Richtlinien zu einem vorhandenen Parameter (Tools für Windows PowerShell)

Hinzufügen von Richtlinien zu einem vorhandenen Parameter mit der Konsole

Gehen Sie wie folgt vor, um Richtlinien zu einem vorhandenen Parameter über die Systems Manager-Konsole hinzufügen.

So fügen Sie einem vorhandenen Parameter Richtlinien hinzu

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie die Option neben dem Parameter, den Sie aktualisieren möchten, um Richtlinien einzuschließen, und klicken Sie anschließend auf Edit (Bearbeiten).
4. Wählen Sie Erweitert aus.
5. (Optional) Wählen Sie im Abschnitt Parameter policies (Parameterrichtlinien) die Option Enabled (Aktiviert) aus. Sie können ein Ablaufdatum und ein oder mehrere Benachrichtigungsrichtlinien für diesen Parameter angeben.
6. Wählen Sie Änderungen speichern aus.

> **Important**
>
> - Parameter Store behält Richtlinien für einen Parameter bei, bis Sie entweder die Richtlinien mit neuen Richtlinien überschreiben oder die Richtlinien entfernen.
> - Um alle Richtlinien aus einem vorhandenen Parameter zu entfernen, bearbeiten Sie den Parameter und wenden Sie eine leere Richtlinie mithilfe von eckigen und geschweiften Klammern wie folgt an: `[{}]`
> - Wenn Sie einem Parameter mit Richtlinien eine neue Richtlinie hinzufügen, überschreibt Systems Manager die dem Parameter angefügten Richtlinien. Die vorhandenen Richtlinien

werden gelöscht. Wenn Sie einem Parameter mit einer oder mehrere Richtlinien eine neue Richtlinie hinzufügen möchten, müssen Sie die ursprünglichen Richtlinien kopieren und einfügen, die neue Richtlinie eingeben und Ihre Änderungen speichern.

Hinzufügen von Richtlinien zu einem vorhandenen Parameter mithilfe des AWS CLI

Gehen Sie wie folgt vor, um einem vorhandenen Parameter mit der AWS CLI Richtlinien hinzuzufügen.

So fügen Sie einem vorhandenen Parameter Richtlinien hinzu

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl zum Hinzufügen von Richtlinien zu einem vorhandenen Parameter aus. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   Linux & macOS

   ```
   aws ssm put-parameter
       --name "parameter name" \
       --value 'parameter value' \
       --type parameter type \
       --overwrite \
       --policies "[{policies-enclosed-in-brackets-and-curly-braces}]"
   ```

   Windows

   ```
   aws ssm put-parameter
       --name "parameter name" ^
       --value 'parameter value' ^
       --type parameter type ^
       --overwrite ^
       --policies "[{policies-enclosed-in-brackets-and-curly-braces}]"
   ```

Hier sehen Sie ein Beispiel mit einer Ablaufrichtlinie, mit der der Parameter nach 15 Tagen gelöscht wird. Das Beispiel enthält auch eine Benachrichtigungsrichtlinie, die fünf (5) Tage vor dem Löschen des Parameters ein EventBridge Ereignis generiert. Außerdem umfasst es eine `NoChangeNotification`-Richtlinie für den Fall, dass an diesem Parameter nach 60 Tagen keine Änderungen vorgenommen werden. Im folgenden Beispiel wird ein verschleierter Name (`3l3vat3131`) für ein Passwort und einen AWS Key Management Service (AWS KMS key) verwendet. Weitere Informationen zu AWS KMS keys finden Sie unter AWS Key Management Service Konzepte im AWS Key Management Service Entwicklerhandbuch.

Linux & macOS

```
aws ssm put-parameter \
    --name "/Finance/Payroll/3l3vat3131" \
    --value "P@sSwW)rd" \
    --type "SecureString" \
    --overwrite \
    --policies "[{\"Type\":\"Expiration\",\"Version\":\"1.0\",\"Attributes\":
{\"Timestamp\":\"2020-05-13T00:00:00.000Z\"}},{\"Type\":\"ExpirationNotification
\",\"Version\":\"1.0\",\"Attributes\":{\"Before\":\"5\",\"Unit\":\"Days\"}},
{\"Type\":\"NoChangeNotification\",\"Version\":\"1.0\",\"Attributes\":{\"After
\":\"60\",\"Unit\":\"Days\"}}]"
```

Windows

```
aws ssm put-parameter ^
    --name "/Finance/Payroll/3l3vat3131" ^
    --value "P@sSwW)rd" ^
    --type "SecureString" ^
    --overwrite ^
    --policies "[{\"Type\":\"Expiration\",\"Version\":\"1.0\",\"Attributes\":
{\"Timestamp\":\"2020-05-13T00:00:00.000Z\"}},{\"Type\":\"ExpirationNotification
\",\"Version\":\"1.0\",\"Attributes\":{\"Before\":\"5\",\"Unit\":\"Days\"}},
{\"Type\":\"NoChangeNotification\",\"Version\":\"1.0\",\"Attributes\":{\"After
\":\"60\",\"Unit\":\"Days\"}}]"
```

3. Führen Sie den folgenden Befehl aus, um die Details zu einem Parameter zu überprüfen. *`parameter name`*Ersetzen Sie es durch Ihre eigenen Informationen.

Linux & macOS

```
aws ssm describe-parameters  \
    --parameter-filters "Key=Name,Values=parameter name"
```

Windows

```
aws ssm describe-parameters  ^
    --parameter-filters "Key=Name,Values=parameter name"
```

Important

- Parameter Store behält Richtlinien für einen Parameter bei, bis Sie entweder die Richtlinien mit neuen Richtlinien überschreiben oder die Richtlinien entfernen.
- Um alle Richtlinien aus einem vorhandenen Parameter zu entfernen, bearbeiten Sie den Parameter und wenden Sie eine leere Richtlinie mithilfe von eckigen und geschweiften Klammern an. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen. Zum Beispiel:

  Linux & macOS

  ```
  aws ssm put-parameter \
      --name parameter name \
      --type parameter type  \
      --value 'parameter value' \
      --policies "[{}]"
  ```

  Windows

  ```
  aws ssm put-parameter ^
      --name parameter name ^
      --type parameter type  ^
      --value 'parameter value' ^
      --policies "[{}]"
  ```

- Wenn Sie einem Parameter mit Richtlinien eine neue Richtlinie hinzufügen, überschreibt Systems Manager die dem Parameter angefügten Richtlinien. Die vorhandenen Richtlinien werden gelöscht. Wenn Sie einem Parameter mit einer oder mehrere Richtlinien eine neue

Richtlinie hinzufügen möchten, müssen Sie die ursprünglichen Richtlinien kopieren und einfügen, die neue Richtlinie eingeben und Ihre Änderungen speichern.

Hinzufügen von Richtlinien zu einem vorhandenen Parameter (Tools für Windows PowerShell)

Gehen Sie wie folgt vor, um mithilfe von Tools für Windows einem vorhandenen Parameter Richtlinien hinzuzufügen PowerShell. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

So fügen Sie einem vorhandenen Parameter Richtlinien hinzu

1. Öffnen Sie Tools für Windows PowerShell und führen Sie den folgenden Befehl aus, um Ihre Anmeldeinformationen anzugeben. Sie müssen entweder über Administratorrechte in Amazon Elastic Compute Cloud (Amazon EC2) verfügen oder Ihnen müssen die entsprechenden Berechtigungen in AWS Identity and Access Management (IAM) erteilt worden sein.

   ```
   Set-AWSCredentials `
       -AccessKey access-key-name `
       -SecretKey secret-key-name
   ```

2. Führen Sie den folgenden Befehl aus, um die Region für Ihre PowerShell Sitzung festzulegen. Im Beispiel wird die Region USA Ost (Ohio) (us-east-2) verwendet.

   ```
   Set-DefaultAWSRegion `
       -Region us-east-2
   ```

3. Führen Sie den folgenden Befehl zum Hinzufügen von Richtlinien zu einem vorhandenen Parameter aus. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   ```
   Write-SSMParameter `
       -Name "parameter name" `
       -Value "parameter value" `
       -Type "parameter type" `
       -Policies "[{policies-enclosed-in-brackets-and-curly-braces}]" `
       -Overwrite
   ```

   Hier sehen Sie ein Beispiel mit einer Ablaufrichtlinie, mit der der Parameter am 13. Mai 2020 um Mitternacht (GMT) gelöscht wird. Das Beispiel enthält auch eine Benachrichtigungsrichtlinie,

die fünf (5) Tage vor dem Löschen des Parameters ein EventBridge Ereignis generiert. Außerdem umfasst es eine `NoChangeNotification`-Richtlinie für den Fall, dass an diesem Parameter nach 60 Tagen keine Änderungen vorgenommen werden. Im folgenden Beispiel wird ein verschleierter Name (`3l3vat3131`) für einen Passwortparameter und einen Von AWS verwalteter Schlüssel verwendet.

```
Write-SSMParameter `
    -Name "/Finance/Payroll/3l3vat3131" `
    -Value "P@sSwW)rd" `
    -Type "SecureString" `
    -Policies "[{\"Type\":\"Expiration\",\"Version\":\"1.0\",\"Attributes\":
{\"Timestamp\":\"2018-05-13T00:00:00.000Z\"}},{\"Type\":\"ExpirationNotification
\",\"Version\":\"1.0\",\"Attributes\":{\"Before\":\"5\",\"Unit\":\"Days\"}},{\"Type
\":\"NoChangeNotification\",\"Version\":\"1.0\",\"Attributes\":{\"After\":\"60\",
\"Unit\":\"Days\"}}]" `
    -Overwrite
```

4. Führen Sie den folgenden Befehl aus, um die Details zu einem Parameter zu überprüfen. *`parameter name`*Ersetzen Sie es durch Ihre eigenen Informationen.

```
(Get-SSMParameterValue -Name "parameter name").Parameters
```

Important

- Parameter Store behält Richtlinien für einen Parameter bei, bis Sie entweder die Richtlinien mit neuen Richtlinien überschreiben oder die Richtlinien entfernen.
- Um alle Richtlinien aus einem vorhandenen Parameter zu entfernen, bearbeiten Sie den Parameter und wenden Sie eine leere Richtlinie mithilfe von eckigen und geschweiften Klammern an. Zum Beispiel:

  ```
  Write-SSMParameter `
      -Name "parameter name" `
      -Value "parameter value" `
      -Type "parameter type" `
      -Policies "[{}]"
  ```

- Wenn Sie einem Parameter mit Richtlinien eine neue Richtlinie hinzufügen, überschreibt Systems Manager die dem Parameter angefügten Richtlinien. Die vorhandenen Richtlinien

werden gelöscht. Wenn Sie einem Parameter mit einer oder mehrere Richtlinien eine neue Richtlinie hinzufügen möchten, müssen Sie die ursprünglichen Richtlinien kopieren und einfügen, die neue Richtlinie eingeben und Ihre Änderungen speichern.

Arbeiten mit Parameterhierarchien in Parameter Store

Das Verwalten Dutzender oder Hunderter Parameter als unsortierte Liste ist zeitaufwendig und fehleranfällig. Außerdem kann es sich als schwierig erweisen, für eine bestimmte Aufgabe den korrekten Parameter zu bestimmen. Sie könnten versehentlich den falschen Parameter verwenden, oder Sie erstellen möglicherweise mehrere Parameter, die dieselben Konfigurationsdaten verwenden.

Mit Parameterhierarchien können Sie -Parameter leichter organisieren und verwalten. Bei einer Hierarchie handelt es sich um einen Parameternamen mit einem Pfad, den Sie mit Schrägstrichen (/) definieren.

Themen

- Erläutern der Parameterhierarchie anhand von Beispielen
- Abfragen von Parametern in einer Hierarchie
- Verwaltung von Parametern mithilfe von Hierarchien mit dem AWS CLI

Erläutern der Parameterhierarchie anhand von Beispielen

Im folgendne Beispiel werden drei Hierarchieebenen im Namen verwendet. Damit wird Folgendes identifiziert:

```
/Environment/Type of computer/Application/Data
```

```
/Dev/DBServer/MySQL/db-string13
```

Sie können eine Hierarchie mit maximal 15 Ebenen erstellen. Wir empfehlen, dass Sie Hierarchien erstellen, die eine vorhandene hierarchische Struktur in Ihrer Umgebung abbilden, wie in den folgenden Beispielen gezeigt:

- Ihre Umgebung für kontinuierliche Integration und kontinuierliche Bereitstellung (CI/CD-Workflows)

  ```
  /Dev/DBServer/MySQL/db-string
  ```

  ```
  /Staging/DBServer/MySQL/db-string
  ```

```
/Prod/DBServer/MySQL/db-string
```

- Die Anwendungen, die Container verwenden

```
/MyApp/.NET/Libraries/my-password
```

- Die Unternehmensstruktur

```
/Finance/Accountants/UserList
```

```
/Finance/Analysts/UserList
```

```
/HR/Employees/EU/UserList
```

Parameterhierarchien standardisieren die Möglichkeiten für die Erstellung von Parameter und vereinfachen mit der Zeit die Verwaltung von Parametern. Eine Parameterhierarchie kann außerdem dazu beitragen, den richtigen Parameter für eine Konfigurationsaufgabe zu bestimmen. Auf diese Weise können Sie vermeiden, dass mehrere Parameter mit denselben Konfigurationsdaten erstellt werden.

Sie können eine Hierarchie erstellen, mit der Sie wie in den folgenden Beispielen gezeigt Parameter über verschiedene Umgebungen hinweg freigeben können; hier werden Passwörter in Entwicklungs- und Staging-Umgebungen verwendet.

```
/DevTest/MyApp/database/my-password
```

Sie könnten anschließend ein eindeutiges Passwort für Ihre produktive Umgebung erstellen, wie im folgenden Beispiel gezeigt:

```
/prod/MyApp/database/my-password
```

Sie müssen dabei nicht unbedingt eine Parameterhierarchie angeben. Sie können Parameter auf Ebene 1 erstellen. Diese werden als Root-Parameter bezeichnet. Um die Abwärtskompatibilität zu gewährleisten, sind alle Parameter, die in Parameter Store erstellt wurden, bevor Hierarchien eingeführt wurden, Root-Parameter. Die Systeme behandelt die folgenden beiden Parameter als Root-Parameter.

```
/parameter-name
```

```
parameter-name
```

Abfragen von Parametern in einer Hierarchie

Ein weiterer Vorteil der Verwendung von Hierarchien ist die Möglichkeit, mithilfe der GetParametersByPathAPI-Operation alle Parameter unter einer bestimmten Ebene in einer Hierarchie abzufragen. Wenn Sie beispielsweise den folgenden Befehl von AWS Command Line Interface (AWS CLI) aus ausführen, gibt das System alle Parameter unter der `Oncall` Ebene zurück:

```
aws ssm get-parameters-by-path --path /Dev/Web/Oncall
```

Beispielausgabe:

```
{
    "Parameters": [
        {
            "Name": "/Dev/Web/Oncall/Week1",
            "Type": "String",
            "Value": "John Doe",
            "Version": 1,
            "LastModifiedDate": "2024-11-22T07:18:53.510000-08:00",
            "ARN": "arn:aws:ssm:us-east-2:123456789012:parameter/Dev/Web/Oncall/Week1",
            "DataType": "text"
        },
        {
            "Name": "/Dev/Web/Oncall/Week2",
            "Type": "String",
            "Value": "Mary Major",
            "Version": 1,
            "LastModifiedDate": "2024-11-22T07:21:25.325000-08:00",
            "ARN": "arn:aws:ssm:us-east-2:123456789012:parameter/Dev/Web/Oncall/Week2",
            "DataType": "text"
        }
    ]
}
```

Sie können den entschlüsselten `SecureString`-Parameter in einer Hierarchie anzeigen, indem Sie den Pfad und den `--with-decryption`-Parameter angeben, wie im folgenden Beispiel gezeigt.

```
aws ssm get-parameters-by-path --path /Prod/ERP/SAP --with-decryption
```

Verwaltung von Parametern mithilfe von Hierarchien mit dem AWS CLI

In dieser Anleitung wird beschrieben, wie Sie mit Parametern und Parameterhierarchien arbeiten können, indem Sie die AWS CLI verwenden.

So verwalten Sie Parameter mithilfe von Hierarchien

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um einen Parameter zu erstellen, der den `allowedPattern`-Parameter und den Parametertyp `String` verwendet. Das zulässige Muster in diesem Beispiel bedeutet, dass der Wert für den Parameter zwischen 1 und 4 Zeichen lang sein muss.

   Linux & macOS

   ```
   aws ssm put-parameter \
       --name "/MyService/Test/MaxConnections" \
       --value 100 --allowed-pattern "\d{1,4}" \
       --type String
   ```

   Windows

   ```
   aws ssm put-parameter ^
       --name "/MyService/Test/MaxConnections" ^
       --value 100 --allowed-pattern "\d{1,4}" ^
       --type String
   ```

   Der Befehl gibt die Versionsnummer des Parameters zurück.

3. Führen Sie den folgenden Befehl aus und versuchen Sie, den gerade erstellten Parameter mit einem neuen Wert zu überschreiben.

   Linux & macOS

   ```
   aws ssm put-parameter \
       --name "/MyService/Test/MaxConnections" \
   ```

```
    --value 10,000 \
    --type String \
    --overwrite
```

Windows

```
aws ssm put-parameter ^
    --name "/MyService/Test/MaxConnections" ^
    --value 10,000 ^
    --type String ^
    --overwrite
```

Das System gibt den folgenden Fehler zurück, da der neue Wert die Anforderungen des zulässigen Musters nicht erfüllt, das Sie im letzten Schritt angegeben haben.

```
An error occurred (ParameterPatternMismatchException) when calling the PutParameter
 operation: Parameter value, cannot be validated against allowedPattern: \d{1,4}
```

4. Führen Sie den folgenden Befehl aus, um einen `SecureString`-Parameter zu erstellen, der den Datentyp Von AWS verwalteter Schlüssel verwendet. Das zulässige Muster in diesem Beispiel bedeutet, dass der Benutzer beliebige Zeichen eingeben kann, und der Wert zwischen 8 und 20 Zeichen lang sein muss.

   Linux & macOS

   ```
   aws ssm put-parameter \
       --name "/MyService/Test/my-password" \
       --value "p#sW*rd33" \
       --allowed-pattern ".{8,20}" \
       --type SecureString
   ```

   Windows

   ```
   aws ssm put-parameter ^
       --name "/MyService/Test/my-password" ^
       --value "p#sW*rd33" ^
       --allowed-pattern ".{8,20}" ^
       --type SecureString
   ```

5. Führen Sie die folgenden Befehle aus, um mehrere Parameter zu erstellen, die die Hierarchiestruktur aus dem letzten Schritt verwenden.

   Linux & macOS

   ```
   aws ssm put-parameter \
       --name "/MyService/Test/DBname" \
       --value "SQLDevDb" \
       --type String
   ```

   ```
   aws ssm put-parameter \
       --name "/MyService/Test/user" \
       --value "SA" \
       --type String
   ```

   ```
   aws ssm put-parameter \
       --name "/MyService/Test/userType" \
       --value "SQLuser" \
       --type String
   ```

   Windows

   ```
   aws ssm put-parameter ^
       --name "/MyService/Test/DBname" ^
       --value "SQLDevDb" ^
       --type String
   ```

   ```
   aws ssm put-parameter ^
       --name "/MyService/Test/user" ^
       --value "SA" ^
       --type String
   ```

   ```
   aws ssm put-parameter ^
       --name "/MyService/Test/userType" ^
       --value "SQLuser" ^
       --type String
   ```

6. Führen Sie den folgenden Befehl aus, um den Wert zweier Parameter abzurufen.

Linux & macOS

```
aws ssm get-parameters \
    --names "/MyService/Test/user" "/MyService/Test/userType"
```

Windows

```
aws ssm get-parameters ^
    --names "/MyService/Test/user" "/MyService/Test/userType"
```

7. Führen Sie den folgenden Befehl aus, um alle Parameter unter einer bestimmten Ebene abzufragen.

   Linux & macOS

   ```
   aws ssm get-parameters-by-path \
       --path "/MyService/Test"
   ```

   Windows

   ```
   aws ssm get-parameters-by-path ^
       --path "/MyService/Test"
   ```

8. Führen Sie den folgenden Befehl aus, um zwei Parameter zu löschen.

   Linux & macOS

   ```
   aws ssm delete-parameters \
       --names "/IADRegion/Dev/user" "/IADRegion/Dev/userType"
   ```

   Windows

   ```
   aws ssm delete-parameters ^
       --names "/IADRegion/Dev/user" "/IADRegion/Dev/userType"
   ```

Zugriff auf Parameter Store-API-Vorgänge verhindern

Mithilfe von dienstspezifischen Bedingungen, die von Systems Manager for AWS Identity and Access Management (IAM) -Richtlinien unterstützt werden, können Sie den Zugriff auf Parameter Store API-

Operationen und -Inhalte explizit zulassen oder verweigern. Mithilfe dieser Bedingungen können Sie nur bestimmten IAM-Entitäten (Benutzern und Rollen) in Ihrer Organisation erlauben, bestimmte API-Aktionen aufzurufen, oder Sie können verhindern, dass bestimmte IAM-Entitäten diese ausführen. Dazu gehören Aktionen, die über die Parameter Store Konsole, die AWS Command Line Interface (AWS CLI) und ausgeführt werden. SDKs

Systems Manager unterstützt derzeit drei Bedingungen, die spezifisch für Parameter Store sind.

Themen

- Verhinderung von Änderungen an vorhandenen Parametern mithilfe von ssm:Overwrite
- Verhinderung der Erstellung oder Aktualisierung von Parametern, die eine Parameterrichtlinie verwenden, unter Verwendung von ssm:Policies
- Den Zugriff auf Ebenen in einem hierarchischen Parameter verhindern mit ssm:Recursive

Verhinderung von Änderungen an vorhandenen Parametern mithilfe von **`ssm:Overwrite`**

Verwenden Sie die `ssm:Overwrite`-Bedingung, um zu steuern, ob IAM-Entitäten vorhandene Parameter aktualisieren können.

In der folgenden Beispielrichtlinie erteilt die `"Allow"` Anweisung die Erlaubnis, Parameter zu erstellen, indem der `PutParameter` API-Vorgang in der AWS-Konto 123456789012 in der Region USA Ost (Ohio) (us-east-2) ausgeführt wird.

Die `"Deny"`-Anweisung verhindert jedoch, dass Entitäten die Werte vorhandener Parameter ändern, da die `Overwrite`-Option für die `PutParameter`-Aktion ausdrücklich abgelehnt wird. Daher können Entitäten, denen diese Richtlinie zugewiesen ist, Parameter erstellen, aber keine Änderungen an vorhandenen Parametern vornehmen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/*"
```

```
        },
        {
            "Effect": "Deny",
            "Action": [
                "ssm:PutParameter"
            ],
            "Condition": {
                "StringEquals": {
                    "ssm:Overwrite": [
                        "true"
                    ]
                }
            },
            "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/*"
        }
    ]
}
```

Verhinderung der Erstellung oder Aktualisierung von Parametern, die eine Parameterrichtlinie verwenden, unter Verwendung von **`ssm:Policies`**

Verwenden Sie die `ssm:Policies`-Bedingung, um zu steuern, ob Entitäten Parameter erstellen können, die eine Parameterrichtlinie enthalten, und bestehende Parameter aktualisieren können, die eine Parameterrichtlinie enthalten.

Im folgenden Richtlinienbeispiel gewährt die `"Allow"` Anweisung allgemeine Berechtigungen zum Erstellen von Parametern, verhindert jedoch, dass Entitäten Parameter erstellen oder aktualisieren, die `"Deny"` eine Parameterrichtlinie in der AWS-Konto 123456789012 in der Region USA Ost (Ohio) (us-east-2) enthalten. Entitäten können weiterhin Parameter erstellen oder aktualisieren, denen keine Parameterrichtlinie zugewiesen ist.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
```

```
            "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ssm:PutParameter"
            ],
            "Condition": {
                "StringEquals": {
                    "ssm:Policies": [
                        "true"
                    ]
                }
            },
            "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/*"
        }
    ]
}
```

Den Zugriff auf Ebenen in einem hierarchischen Parameter verhindern mit **`ssm:Recursive`**

Verwenden Sie die `ssm:Recursive`-Bedingung, um zu steuern, ob IAM-Entitäten Ebenen in einem hierarchischen Parameter anzeigen oder darauf verweisen können. Sie können den Zugriff auf alle Parameter außerhalb einer bestimmten Hierarchieebene gewähren oder einschränken.

In der folgenden Beispielrichtlinie gewährt die `"Allow"`-Anweisung Zugriff auf Parameter Store-Operationen mit allen Parametern im Pfad `/Code/Departments/Finance/*` für AWS-Konto 123456789012 in der Region USA Ost (Ohio) (us-east-2).

Danach verhindert die `"Deny"`-Anweisung, dass IAM-Entitäten Parameterdaten auf oder unter der Ebene von `/Code/Departments/*` anzeigen oder abrufen. Entitäten können jedoch weiterhin Parameter in diesem Pfad erstellen oder aktualisieren. Das Beispiel wurde erstellt, um zu verdeutlichen, dass die rekursive Verweigerung von Zugriffen unterhalb einer bestimmten Ebene in einer Parameterhierarchie Vorrang vor erlaubteren Zugriffen in derselben Richtlinie hat.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "ssm:*"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ssm:GetParametersByPath"
            ],
            "Condition": {
                "StringEquals": {
                    "ssm:Recursive": [
                        "true"
                    ]
                }
            },
            "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/Code/
Departments/*"
        }
    ]
}
```

Important

Wenn ein Benutzer Zugriff auf einen Pfad hat, kann er auf alle Ebenen dieses Pfads zugreifen. Wenn ein Benutzer beispielsweise die Berechtigung für den Zugriff auf den Pfad `/a` besitzt, dann kann er auch auf `/a/b` zugreifen. Dies gilt, sofern dem Benutzer nicht ausdrücklich der Zugriff auf Parameter `/b` in IAM verweigert wurde, wie oben dargestellt.

Arbeiten mit Parameterbezeichnungen in Parameter Store

Eine Parameter-Bezeichnung ist ein benutzerdefinierter Alias, mit dem Sie verschiedene Versionen eines Parameters verwalten können. Wenn Sie einen Parameter ändern, AWS Systems Manager wird automatisch eine neue Version gespeichert und die Versionsnummer um eins erhöht. Dank einer Bezeichnung können Sie sich den Zweck einer Parameterversion merken, wenn mehrere Versionen vorhanden sind.

Nehmen wir beispielsweise an, Sie haben einen Parameter mit dem Namen `/MyApp/DB/ConnectionString`. Der Wert des Parameters ist eine Verbindungszeichenfolge mit einem MySQL-Server in einer lokalen Datenbank einer Testumgebung. Nachdem Sie die Anwendung aktualisiert haben, möchten Sie festlegen, dass der Parameter eine Verbindungszeichenfolge für eine Produktionsdatenbank verwendet. Sie ändern den Wert von `/MyApp/DB/ConnectionString`. Systems Manager erstellt automatisch Version 2 mit der neuen Verbindungszeichenfolge. Damit Sie sich den Zweck der einzelnen Versionen besser merken können, fügen Sie jedem Parameter eine Bezeichnung an. Für Version eins fügen Sie die Bezeichnung Test an. Für Version zwei fügen Sie die Bezeichnung Production an.

Sie können Bezeichnungen von einer Version eines Parameters in eine andere Version verschieben. Wenn Sie beispielsweise Version drei des Parameters `/MyApp/DB/ConnectionString` mit einer Verbindungszeichenfolge für eine neue Produktionsdatenbank erstellen, können Sie die Bezeichnung Production von Version zwei des Parameters zu Version drei des Parameters verschieben.

Parameterbezeichnungen stellen eine einfache Alternative zu Parameter-Tags dar. Ihre Organisation verfügt u. U. über strenge Richtlinien für Tags, die auf verschiedene AWS -Ressourcen angewendet werden müssen. Im Gegensatz dazu ist eine Bezeichnung einfach eine Textzuordnung für eine bestimmte Version eines Parameters.

Ähnlich wie Tags können Sie Parameter mithilfe von Bezeichnungen abfragen. Sie können eine Liste bestimmter Parameterversionen anzeigen, die alle dieselbe Bezeichnung verwenden, wenn Sie Ihren Parametersatz mithilfe der GetParametersByPathAPI-Operation abfragen, wie weiter unten in diesem Abschnitt beschrieben.

> **Note**
>
> Wenn Sie einen Befehl ausführen, der eine Version eines Parameters angibt, die nicht existiert, schlägt der Befehl fehl. Es greift nicht auf den letzten oder Standardwert des Parameters zurück.

Anforderungen und Einschränkungen für Bezeichnungen

Für Parameterbezeichnungen gelten die folgenden Anforderungen und Einschränkungen:

- Für eine Version eines Parameters sind maximal 10 Bezeichnungen zulässig.

- Es ist nicht möglich, die gleiche Bezeichnung verschiedenen Versionen desselben Parameters anzufügen. Wenn Version 1 des Parameters beispielsweise die Bezeichnung Production hat, können Sie Production nicht an Version 2 anfügen.
- Sie können eine Bezeichnung von einer Version eines Parameters zu einer anderen Version verschieben.
- Es ist nicht möglich, eine Bezeichnung festzulegen, wenn Sie einen Parameter erstellen. Sie müssen eine Bezeichnung einer bestimmten Version eines Parameters anfügen.
- Wenn Sie eine Parameterbezeichnung nicht mehr verwenden möchten, können Sie sie zu einer anderen Version eines Parameters verschieben oder sie löschen.
- Eine Bezeichnung darf höchstens 100 Zeichen lang sein.
- Bezeichnungen können Buchstaben (Unterscheidung nach Groß- und Kleinschreibung), Ziffern, Punkte (.), Bindestriche (-) und Unterstriche (_) enthalten.
- Bezeichnungen dürfen nicht mit einer Zahl, „aws“ oder „ssm“ (keine Unterscheidung nach Groß- und Kleinschreibung) beginnen. Wenn eine Bezeichnung diese Anforderungen nicht erfüllt, wird sie der Parameterversion nicht angefügt und vom System in der Liste `InvalidLabels` angezeigt.

Themen

- Arbeiten mit Parameterbezeichnungen mithilfe der Konsole
- Arbeiten mit Parameterbeschriftungen unter Verwendung des AWS CLI

Arbeiten mit Parameterbezeichnungen mithilfe der Konsole

In diesem Abschnitt wird beschrieben, wie Sie die folgenden Aufgaben mithilfe der Systems Manager-Konsole durchführen.

- Eerstellen eines Parameterlabels mithilfe der Konsole
- So zeigen Sie Bezeichnungen, die einem Parameter angefügt sind, mithilfe der Konsole an
- Verschieben von Parameterbezeichnungen mithilfe der Konsole
- Löschen der Parameterbezeichnungen mithilfe der Konsole

Eerstellen eines Parameterlabels mithilfe der Konsole

Im folgenden Verfahren wird beschrieben, wie Sie einer bestimmten Version eines vorhandenen Parameters über die Systems Manager-Konsole eine Bezeichnung anfügen. Es ist nicht möglich, eine Bezeichnung anzufügen, wenn Sie einen neuen Parameter erstellen.

Anfügen einer Bezeichnung an die aktuelle Version eines Parameters

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie den Namen eines Parameters aus, um die Detailseite für diesen Parameter anzuzeigen.
4. Wählen Sie die Registerkarte History (Verlauf) aus.
5. Wählen Sie die Parameterversion aus, der Sie eine Bezeichnung anfügen möchten.
6. Klicken Sie auf Verwalten von Bezeichnungen.
7. Klicken Sie auf Hinzufügen einer neuen Bezeichnung.
8. Geben Sie den Namen der Bezeichnung in das Textfeld ein. Wählen Sie Add new label (Neue Bezeichnung hinzufügen) aus, um weitere Bezeichnungen hinzuzufügen. Sie können maximal zehn Bezeichnungen anfügen.
9. Wenn Sie fertig sind, wählen Sie Änderungen speichern aus.

So zeigen Sie Bezeichnungen, die einem Parameter angefügt sind, mithilfe der Konsole an

Für eine Parameterversion sind maximal 10 Bezeichnungen zulässig. Im folgenden Verfahren wird beschrieben, wie Sie alle Bezeichnungen, die einer Parameterversion angefügt sind, mithilfe der Systems Manager-Konsole anzeigen.

Anzeigen von Bezeichnungen, die einer Parameterversion angefügt sind

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie den Namen eines Parameters aus, um die Detailseite für diesen Parameter anzuzeigen.
4. Wählen Sie die Registerkarte History (Verlauf) aus.
5. Suchen Sie die Parameterversion, für die Sie die angefügten Bezeichnungen anzeigen möchten. Die Spalte Labels (Bezeichnungen) enthält alle Bezeichnungen, die der Parameterversion angefügt sind.

Verschieben von Parameterbezeichnungen mithilfe der Konsole

Im folgenden Verfahren wird beschrieben, wie Sie eine Parameterbezeichnung zu einer anderen Version desselben Parameters mithilfe der Systems Manager-Konsole verschieben.

So verschieben Sie eine Bezeichnung zu einer Parameterversion

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie den Namen eines Parameters aus, um die Detailseite für diesen Parameter anzuzeigen.
4. Wählen Sie die Registerkarte History (Verlauf) aus.
5. Wählen Sie die Parameterversion aus, deren Bezeichnung Sie verschieben möchten.
6. Klicken Sie auf Verwalten von Bezeichnungen.
7. Klicken Sie auf Hinzufügen einer neuen Bezeichnung.
8. Geben Sie den Namen der Bezeichnung in das Textfeld ein.
9. Wenn Sie fertig sind, wählen Sie Änderungen speichern aus.

Löschen der Parameterbezeichnungen mithilfe der Konsole

Im folgenden Verfahren wird beschrieben, wie Sie über die Systems Manager-Konsole eine oder mehrere Parameterbezeichnungen löschen.

So löschen Sie Bezeichnungen aus einem Parameter

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie den Namen eines Parameters aus, um die Detailseite für diesen Parameter anzuzeigen.
4. Wählen Sie die Registerkarte History (Verlauf) aus.
5. Wählen Sie die Parameterversion aus, deren Bezeichnungen Sie löschen möchten.
6. Klicken Sie auf Verwalten von Bezeichnungen.
7. Klicken Sie neben jeder Bezeichnung, die Sie löschen möchten, auf Remove (Entfernen).

8. Wenn Sie fertig sind, wählen Sie Änderungen speichern aus.
9. Bestätigen Sie, dass Ihre Änderungen korrekt sind, geben Sie `Confirm` in das Textfeld ein und wählen Sie Bestätigen aus.

Arbeiten mit Parameterbeschriftungen unter Verwendung des AWS CLI

In diesem Abschnitt wird beschrieben, wie Sie die folgenden Aufgaben mithilfe der AWS Command Line Interface (AWS CLI) durchführen.

- Erstellen einer neuen Parameterbezeichnung mit dem AWS CLI
- Beschriftungen für einen Parameter anzeigen mit dem AWS CLI
- Anzeigen einer Liste von Parametern, denen ein Label zugewiesen wurde, mithilfe der AWS CLI
- Verschieben einer Parameterbezeichnung mit dem AWS CLI
- Löschen von Parameterbeschriftungen mit dem AWS CLI

Erstellen einer neuen Parameterbezeichnung mit dem AWS CLI

Im folgenden Verfahren wird beschrieben, wie Sie einer bestimmten Version eines vorhandenen Parameters über die AWS CLI eine Bezeichnung anfügen. Es ist nicht möglich, eine Bezeichnung anzufügen, wenn Sie einen neuen Parameter erstellen.

So erstellen Sie eine neue Parameterbezeichnung

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus, um eine Liste der Parameter anzuzeigen, für die Sie über die Berechtigung zum Anfügen einer Bezeichnung verfügen.

> **Note**
>
> Parameter sind nur dort verfügbar AWS-Region , wo sie erstellt wurden. Falls Sie einen Parameter, dem Sie eine Bezeichnung anfügen möchten, nicht finden können, prüfen Sie Ihre Region.

```
aws ssm describe-parameters
```

Notieren Sie den Namen eines Parameters, dem Sie eine Bezeichnung anfügen möchten.

3. Führen Sie den folgenden Befehl aus, um alle Versionen des Parameters anzuzeigen.

```
aws ssm get-parameter-history --name "parameter-name"
```

Notieren Sie die Parameterversion, der Sie eine Bezeichnung anfügen möchten.

4. Führen Sie den folgenden Befehl aus, um anhand der Versionsnummer Informationen zu einem Parameter abzurufen.

```
aws ssm get-parameters --names "parameter-name:version-number"
```

Ein Beispiel.

```
aws ssm get-parameters --names "/Production/SQLConnectionString:3"
```

5. Führen Sie einen der folgenden Befehle aus, um einer Parameterversion eine Bezeichnung anzufügen. Wenn Sie mehrere Bezeichnungen anzufügen, müssen Sie die Namen der Bezeichnungen durch ein Leerzeichen trennen.

Anfügen einer Bezeichnung an die aktuelle Version eines Parameters

```
aws ssm label-parameter-version --name parameter-name  --labels label-name
```

Anfügen einer Bezeichnung an eine bestimmte Version eines Parameters

```
aws ssm label-parameter-version --name parameter-name --parameter-version version-
number --labels label-name
```

Hier sind einige Beispiele.

```
aws ssm label-parameter-version --name /config/endpoint --labels production east-
region finance
```

```
aws ssm label-parameter-version --name /config/endpoint --parameter-version 3 --
labels MySQL-test
```

> **Note**
>
> Wenn die Ausgabe die Bezeichnung zeigt, die Sie in der Liste `InvalidLabels` erstellt haben, entspricht die Bezeichnung nicht den weiter oben in diesem Thema beschriebenen Anforderungen. Überprüfen Sie die Anforderungen und versuchen Sie es erneut. Wenn die Liste `InvalidLabels` leer ist, wurde Ihre Bezeichnung der Version des Parameters erfolgreich angefügt.

6. Sie können die Details des Parameters entweder mithilfe einer Versionsnummer oder eines Bezeichnungsnamens anzeigen. Führen Sie den folgenden Befehl aus und geben Sie die im vorherigen Schritt erstellte Bezeichnung an.

```
aws ssm get-parameter --name parameter-name:label-name --with-decryption
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "Parameter": {
        "Version": version-number,
        "Type": "parameter-type",
        "Name": "parameter-name",
        "Value": "parameter-value",
        "Selector": ":label-name"
    }
}
```

> **Note**
>
> Selector (Auswahl) in der Ausgabe ist entweder die Versionsnummer oder die Bezeichnung, die Sie im Eingabefeld `Name` angegeben haben.

Beschriftungen für einen Parameter anzeigen mit dem AWS CLI

Sie können den GetParameterHistoryAPI-Vorgang verwenden, um den vollständigen Verlauf und alle Labels anzuzeigen, die einem bestimmten Parameter zugeordnet sind. Oder Sie können den GetParametersByPathAPI-Vorgang verwenden, um eine Liste aller Parameter anzuzeigen, denen ein bestimmtes Label zugewiesen wurde.

Um Beschriftungen für einen Parameter mithilfe der GetParameterHistory API-Operation anzuzeigen

1. Führen Sie den folgenden Befehl aus, um eine Liste der Parameter anzuzeigen, für die Sie Bezeichnungen anzeigen können.

   > **Note**
   >
   > Parameter sind nur in den Regionen verfügbar, in denen sie erstellt wurden. Falls Sie einen Parameter, für den Sie eine Bezeichnung verschieben möchten, nicht finden können, prüfen Sie Ihre Region.

   ```
   aws ssm describe-parameters
   ```

   Notieren Sie sich den Namen des Parameters, dessen Bezeichnungen Sie anzeigen möchten.

2. Führen Sie den folgenden Befehl aus, um alle Versionen des Parameters anzuzeigen.

   ```
   aws ssm get-parameter-history --name parameter-name --with-decryption
   ```

   Das System gibt unter anderem folgende Informationen zurück

   ```
   {
       "Parameters": [
           {
               "Name": "/Config/endpoint",
               "LastModifiedDate": 1528932105.382,
               "Labels": [
                   "Deprecated"
               ],
               "Value": "MyTestService-June-Release.example.com",
               "Version": 1,
               "LastModifiedUser": "arn:aws:iam::123456789012:user/test",
   ```

```
            "Type": "String"
        },
        {
            "Name": "/Config/endpoint",
            "LastModifiedDate": 1528932111.222,
            "Labels": [
                "Current"
            ],
            "Value": "MyTestService-July-Release.example.com",
            "Version": 2,
            "LastModifiedUser": "arn:aws:iam::123456789012:user/test",
            "Type": "String"
        }
    ]
}
```

Anzeigen einer Liste von Parametern, denen ein Label zugewiesen wurde, mithilfe der AWS CLI

Sie können den GetParametersByPathAPI-Vorgang verwenden, um eine Liste aller Parameter in einem Pfad anzuzeigen, denen ein bestimmtes Label zugewiesen wurde.

Führen Sie den folgenden Befehl aus, um eine Liste der Parameter in einem Pfad anzuzeigen, denen eine bestimmte Bezeichnung zugeordnet wurde. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

```
aws ssm get-parameters-by-path \
    --path parameter-path \
    --parameter-filters Key=Label,Values=label-name,Option=Equals \
    --max-results a-number \
    --with-decryption --recursive
```

Das System gibt unter anderem folgende Informationen zurück In diesem Beispiel durchsuchte der Benutzer den `/Config`-Pfad.

```
{
    "Parameters": [
        {
            "Version": 3,
            "Type": "SecureString",
            "Name": "/Config/DBpwd",
            "Value": "MyS@perGr&pass33"
```

```
        },
        {
            "Version": 2,
            "Type": "String",
            "Name": "/Config/DBusername",
            "Value": "TestUserDB"
        },
        {
            "Version": 2,
            "Type": "String",
            "Name": "/Config/endpoint",
            "Value": "MyTestService-July-Release.example.com"
        }
    ]
}
```

Verschieben einer Parameterbezeichnung mit dem AWS CLI

Im folgenden Verfahren wird beschrieben, wie Sie eine Parameterbezeichnung zu einer anderen Version desselben Parameters verschieben.

So verschieben Sie eine Parameterbezeichnung

1. Führen Sie den folgenden Befehl aus, um alle Versionen des Parameters anzuzeigen. *parameter name*Ersetzen Sie es durch Ihre eigenen Informationen.

   ```
   aws ssm get-parameter-history \
       --name "parameter name"
   ```

   Beachten Sie die Parameterversionen, aus denen Sie die Bezeichnung verschieben möchten.

2. Führen Sie den folgenden Befehl aus, um eine vorhandene Bezeichnung einer anderen Version eines Parameters zuzuweisen. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   ```
   aws ssm label-parameter-version \
       --name parameter name \
       --parameter-version version number \
       --labels name-of-existing-label
   ```

> **Note**
>
> Wenn Sie eine vorhandene Bezeichnung zur neuesten Version eines Parameters verschieben möchten, entfernen Sie `--parameter-version` aus dem Befehl.

Löschen von Parameterbeschriftungen mit dem AWS CLI

Im folgenden Verfahren wird beschrieben, wie Sie Parameterbezeichnungen mithilfe der AWS CLI löschen.

So löschen Sie eine Parameterbezeichnung

1. Führen Sie den folgenden Befehl aus, um alle Versionen des Parameters anzuzeigen. *`parameter name`*Ersetzen Sie durch Ihre eigenen Informationen.

```
aws ssm get-parameter-history \
    --name "parameter name"
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "Parameters": [
        {
            "Name": "foo",
            "DataType": "text",
            "LastModifiedDate": 1607380761.11,
            "Labels": [
                "l3",
                "l2"
            ],
            "Value": "test",
            "Version": 1,
            "LastModifiedUser": "arn:aws:iam::123456789012:user/test",
            "Policies": [],
            "Tier": "Standard",
            "Type": "String"
        },
        {
            "Name": "foo",
```

```
            "DataType": "text",
            "LastModifiedDate": 1607380763.11,
            "Labels": [
                "l1"
            ],
            "Value": "test",
            "Version": 2,
            "LastModifiedUser": "arn:aws:iam::123456789012:user/test",
            "Policies": [],
            "Tier": "Standard",
            "Type": "String"
        }
    ]
}
```

Notieren Sie die Parameterversion, für die Sie eine oder mehrere Bezeichnungen löschen möchten.

2. Führen Sie den folgenden Befehl aus, um die Bezeichnungen zu löschen, die Sie aus diesem Parameter auswählen. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

```
aws ssm unlabel-parameter-version \
    --name parameter name \
    --parameter-version version \
    --labels label 1,label 2,label 3
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "InvalidLabels": ["invalid"],
    "DeletedLabels" : ["Prod"]
 }
```

Arbeiten mit Parameterversionen in Parameter Store

Jedes Mal, wenn Sie den Wert eines Parameters bearbeiten, erstellt Parameter Store, ein Tool in AWS Systems Manager, eine neue Version des Parameters und behält die vorherigen Versionen bei. Zu Beginn der Erstellung eines Parameters weist Parameter Store diesem Parameter die Version 1 zu. Wenn Sie den Wert des Parameters ändern, erhöht Parameter Store die Versionsnummer

automatisch um eins. Sie können die Details, einschließlich der Werte, aller Versionen im Verlauf eines Parameters anzeigen.

Sie können auch die Version eines Parameters angeben, der in API-Befehlen und SSM-Dokumenten verwendet werden soll. Beispiel: `ssm:MyParameter:3`. Sie können einen Parameternamen und eine bestimmte Versionsnummer in API-Aufrufen und SSM-Dokumenten angeben. Wenn Sie keine Versionsnummer angeben, verwendet das System automatisch die neueste Version. Wenn Sie die Nummer für eine nicht vorhandene Version angeben, gibt das System einen Fehler zurück, anstatt auf die neueste oder Standardversion des Parameters zurückzugreifen.

Sie können Parameterversionen verwenden, um zu sehen, wie oft ein Parameter im Lauf eines bestimmten Zeitraums geändert wurde. Parameterversionen bieten auch eine Schutzebene, wenn ein Parameterwert versehentlich geändert wird.

Sie können maximal 100 Versionen eines Parameters erstellen verwalten. Nachdem Sie 100 Versionen eines Parameters erstellt haben, wird jedes Mal, wenn Sie eine neue Version erstellen, die älteste Version des Parameters aus dem Verlauf entfernt, um Platz für die neue Version zu schaffen.

Eine Ausnahme ist, wenn bereits 100 Parameterversionen im Verlauf vorhanden sind und der ältesten Version eines Parameters eine Parameterbezeichnung zugewiesen wird. In diesem Fall wird diese Version nicht aus dem Verlauf entfernt, und die Anforderung, eine neue Parameterversion zu erstellen, schlägt fehl. Diese Schutzmaßnahme soll verhindern, dass Parameterversionen mit ihnen zugewiesenen geschäftskritischen Bezeichnungen gelöscht werden. Um mit dem Erstellen neuer Parameter fortzufahren, verschieben Sie die Bezeichnung zuerst von der ältesten Version des Parameters in eine neuere Version, um sie in Ihren Operationen verwenden zu können. Informationen zum Verschieben von Parameterbezeichnungen finden Sie unter Verschieben von Parameterbezeichnungen mithilfe der Konsole und Verschieben einer Parameterbezeichnung mit dem AWS CLI.

Das folgende Verfahren zeigt, wie Sie einen Parameter bearbeiten und dann überprüfen, ob Sie eine neue Version erstellt haben. Sie können die Befehle `get-parameter` und `get-parameters` verwenden, um Parameterversionen anzuzeigen. Weitere Informationen zur Verwendung dieser Befehle finden Sie unter GetParameter und GetParameters in der AWS Systems Manager-API-Referenz

Themen

- Erstellen einer neuen Version eines Parameters mithilfe der Konsole
- Verweisen auf eine Parameterversion

Erstellen einer neuen Version eines Parameters mithilfe der Konsole

Sie können die Systems Manager-Konsole verwenden, um eine neue Version eines Parameters zu erstellen und den Versionsverlauf eines Parameters anzuzeigen.

So erstellen Sie eine neue Version eines Parameters

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie den Namen eines Parameters aus, den Sie vorher erstellt haben. Weitere Informationen zum Erstellen eines neuen Parameters finden Sie unter Parameter Store-Parameter im Systems Manager erstellen.
4. Wählen Sie Bearbeiten aus.
5. Geben Sie im Feld Value einen neuen Wert ein und klicken Sie auf Save changes.
6. Wählen Sie den Namen des Parameters aus, den Sie gerade aktualisiert haben. Prüfen Sie auf der Registerkarte Overview, dass die Versionsnummer um 1 erhöht wurde, und überprüfen Sie den neuen Wert.
7. Um den Verlauf aller Versionen eines Parameters anzuzeigen, wählen Sie die Registerkarte History (Verlauf) aus.

Verweisen auf eine Parameterversion

Sie können in Befehlen, API-Aufrufen und SSM-Dokumenten mithilfe des folgenden Formats auf spezifische Parameterversionen verweisen: ssm: *`parameter-name`*`:`*`version-number`*.

Im folgenden Beispiel verwendet die Amazon Elastic Compute Cloud (Amazon EC2) `run-instances command` Version 3 des Parameters `golden-ami`.

Linux & macOS

```
aws ec2 run-instances \
    --image-id resolve:ssm:/golden-ami:3 \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --security-groups my-security-group
```

Windows

```
aws ec2 run-instances ^
    --image-id resolve:ssm:/golden-ami:3 ^
    --count 1 ^
    --instance-type t2.micro ^
    --key-name my-key-pair ^
    --security-groups my-security-group
```

> **Note**
>
> Das Verwenden von `resolve` und einem Parameterwert ist nur mit der Option `--image-id` und einem Parameter, der ein Amazon Machine Image (AMI) als Wert enthält, möglich. Weitere Informationen finden Sie unter Verwendung der nativen Parameterunterstützung Parameter Store für Amazon Machine Image IDs.

Hier ist ein Beispiel für die Angabe von Version 2 eines Parameters mit dem Namen `MyRunCommandParameter` in einem SSM-Dokument.

YAML

```
---
schemaVersion: '2.2'
description: Run a shell script or specify the commands to run.
parameters:
  commands:
    type: String
    description: "(Required) Specify a shell script or a command to run."
    displayType: textarea
    default: "{{ssm:MyRunCommandParameter:2}}"
mainSteps:
- action: aws:runShellScript
  name: RunScript
  inputs:
    runCommand:
    - "{{commands}}"
```

JSON

```
{
    "schemaVersion": "2.2",
    "description": "Run a shell script or specify the commands to run.",
    "parameters": {
        "commands": {
            "type": "String",
            "description": "(Required) Specify a shell script or a command to run.",
            "displayType": "textarea",
            "default": "{{ssm:MyRunCommandParameter:2}}"
        }
    },
    "mainSteps": [
        {
            "action": "aws:runShellScript",
            "name": "RunScript",
            "inputs": {
                "runCommand": [
                    "{{commands}}"
                ]
            }
        }
    ]
}
```

Arbeiten mit gemeinsam genutzten Parametern in Parameter Store

Die gemeinsame Nutzung erweiterter Parameter vereinfacht die Verwaltung von Konfigurationsdaten in einer Umgebung mit mehreren Konten. Sie können Ihre Parameter zentral speichern und verwalten und sie mit anderen AWS-Konten teilen, die sie referenzieren müssen.

Parameter Store lässt sich in AWS Resource Access Manager (AWS RAM) integrieren, um die erweiterte gemeinsame Nutzung von Parametern zu ermöglichen. AWS RAM ist ein Service, der es Ihnen ermöglicht, Ressourcen mit anderen AWS-Konten oder über andere AWS Organizations zu teilen.

Mit AWS RAM geben Sie Ressourcen in Ihrem Besitz frei, indem Sie eine Ressourcenfreigabe erstellen. Eine Ressourcenfreigabe gibt die freizugebenden Ressourcen, die zu erteilenden Berechtigungen und die Verbraucher an, mit denen die Freigabe erfolgen soll. Zu den Verbrauchern können folgende Angaben zählen:

- Spezifische AWS-Konten innerhalb oder außerhalb seiner AWS Organizations-Organisation
- Eine Organisationseinheit innerhalb seiner Organisation in AWS Organizations
- Seine gesamte Organisation in AWS Organizations

Weitere Informationen zu AWS RAM finden Sie im AWS RAM-Benutzerhandbuch.

In diesem Thema wird erklärt, wie Sie Parameter teilen, deren Eigentümer Sie sind, und wie Sie Parameter verwenden, die mit Ihnen gemeinsam genutzt werden.

Inhalt

- Voraussetzungen für die Freigabe von Parametern
- Freigabe eines Parameters
- Beenden der Freigabe eines Parameters
- Identifizieren freigegebenen Parametern
- Zugreifen auf freigegebene Parameter
- Berechtigungssätze für die gemeinsame Nutzung von Parametern
- Maximaler Durchsatz für freigegebene Parameter
- Preisgestaltung für freigegebene Parameter
- Kontoübergreifender Zugriff für geschlossene AWS-Konten

Voraussetzungen für die Freigabe von Parametern

Die folgenden Voraussetzungen müssen erfüllt sein, bevor Sie die Parameter von Ihrem Konto freigegebenen:

- Um einen Parameter freigeben zu können, müssen Sie diesen in Ihrem AWS-Konto besitzen. Sie können keinen Parameter freigeben, der für Sie freigegeben wurde.
- Um einen Parameter gemeinsam nutzen zu können, muss er sich in der erweiterten Parameterebene befinden. Weitere Informationen zu den Parameterebenen finden Sie unter Verwalten von Parameterstufen. Hinweise zum Ändern eines vorhandenen Standardparameters in einen erweiterten Parameter finden Sie unter Ändern eines Standardparameters in einen fortgeschrittenen Parameter.
- Um einen `SecureString`-Parameter gemeinsam zu nutzen, muss er mit einem vom Kunden verwalteten Schlüssel verschlüsselt werden und Sie müssen den Schlüssel separat durch AWS

Key Management Service weitergeben. Von AWS verwaltete Schlüssel kann nicht geteilt werden. Mit dem Standard verschlüsselte Parameter Von AWS verwalteter Schlüssel können aktualisiert werden, sodass sie stattdessen einen kundenverwalteten Schlüssel verwenden. AWS KMS-Schlüsseldefinitionen finden Sie unter AWS KMS-Konzepte im AWS Key Management Service-Entwicklerhandbuch.

- Um einen Parameter für Ihre Organisation oder eine Organisationseinheit in AWS Organizations freigeben zu können, müssen Sie die Freigabe mit AWS Organizations aktivieren. Weitere Informationen finden Sie unter Freigabe für AWS Organizations aktivieren im AWS RAM-Benutzerhandbuch.

Freigabe eines Parameters

Um einen Parameter freigeben zu können, müssen Sie ihn einer Ressourcenfreigabe hinzufügen. Eine Ressourcenfreigabe ist eine AWS RAM-Ressource, mit der Sie Ihre Ressourcen in mehreren AWS-Konten gemeinsam nutzen können. Eine Ressourcenfreigabe gibt die freizugebenden Ressourcen und die Konsumenten an, für die sie freigegeben werden.

Wenn Sie einen Parameter, dessen Eigentümer Sie sind, mit anderen AWS-Konten teilen, können Sie zwischen zwei von AWS verwalteten Berechtigungen wählen, die Sie den Verbrauchern gewähren möchten. Weitere Informationen finden Sie unter Berechtigungssätze für die gemeinsame Nutzung von Parametern.

Wenn Sie Teil einer Organisation in AWS Organizations sind und die gemeinsame Nutzung innerhalb Ihrer Organisation aktiviert ist, können Sie Konsumenten in Ihrer Organisation von der AWS RAM-Konsole aus Zugriff auf den gemeinsamen Parameter gewähren. Andernfalls erhalten Konsumenten eine Einladung zur Teilnahme an der Ressourcenfreigabe und nach Annahme der Einladung wird ihnen Zugriff auf den freigegebenen Parameter gewährt.

Sie können einen Parameter, den Sie besitzen, über die AWS RAM oder die AWS CLI freigeben.

> **Note**
>
> Sie können einen Parameter zwar mithilfe des PutResourcePolicy-API-Vorgangs von Systems Manager gemeinsam nutzen, wir empfehlen jedoch, stattdessen AWS Resource Access Manager (AWS RAM) zu verwenden. Das liegt daran, dass die Verwendung des Parameters den zusätzlichen Schritt `PutResourcePolicy` erfordert, den Parameter mithilfe des API-Vorgangs AWS RAM PromoteResourceShareCreatedFromPolicy auf eine standardmäßige Resource Share hochzustufen. Andernfalls wird der Parameter nicht

vom DescribeParametern-API-Vorgang von Systems Manager zurückgegeben, der die `--shared`-Option verwendet.

So geben Sie einen Parameter in Ihrem Besitz mithilfe der AWS RAM-Konsole frei

Siehe Erstellen einer Ressourcenfreigabe in AWS RAM im AWS RAM-Benutzerhandbuch.

Treffen Sie die folgenden Auswahlen, während Sie das Verfahren abschließen:

- Wählen Sie auf der Seite Schritt 1 unter Ressourcen `Parameter Store Advanced Parameter` „aus und aktivieren Sie dann das Kontrollkästchen jedes Parameters in der erweiterten Parameterebene, den Sie freigeben möchten.
- Wählen Sie auf der Seite Schritt 2 für Verwaltete Berechtigungen die Berechtigung aus, die Verbrauchern gewährt werden soll, wie weiter unten unter Berechtigungssätze für die gemeinsame Nutzung von Parametern in diesem Thema beschrieben.

Wählen Sie andere Optionen auf der Grundlage Ihrer Ziele für die gemeinsame Nutzung von Parametern aus.

So geben Sie einen Parameter in Ihrem Besitz mithilfe der AWS CLI frei

Verwenden Sie den create-resource-share-Befehl, um Parameter zu einer neuen Ressourcenfreigabe hinzuzufügen.

Verwenden Sie den associate-resource-share-Befehl, um einer vorhandenen Ressourcenfreigabe Parameter hinzuzufügen.

Im folgenden Beispiel wird eine neue Ressourcenfreigabe erstellt, um Parameter mit Verbrauchern in einer Organisation und in einem Einzelkonto gemeinsam zu nutzen.

```
aws ram create-resource-share \
    --name "MyParameter" \
    --resource-arns "arn:aws:ssm:us-east-2:123456789012:parameter/MyParameter" \
    --principals "arn:aws:organizations::123456789012:ou/o-63bEXAMPLE/ou-46xi-rEXAMPLE"
 "987654321098"
```

Beenden der Freigabe eines Parameters

Wenn Sie die gemeinsame Nutzung eines gemeinsam genutzten Parameters beenden, kann das Verbraucherkonto nicht mehr auf den Parameter zugreifen.

Um die Freigabe eines Parameters in Ihrem Besitz zu beenden, müssen Sie diesen aus der Ressourcenfreigabe entfernen. Hierzu können Sie die Systems Manager-Konsole, die AWS RAM-Konsole oder die AWS CLI verwenden.

So beenden Sie die Freigabe eines Parameters in Ihrem Besitz mithilfe der AWS RAM-Konsole

Siehe Aktualisieren einer Ressourcenfreigabe in AWS RAM im AWS RAM-Benutzerhandbuch.

So beenden Sie die Freigabe eines Parameters in Ihrem Besitz mithilfe der AWS CLI

Verwenden Sie den Befehl disassociate-resource-share.

Identifizieren freigegebenen Parametern

Besitzer und Konsumenten können freigegebene Parameter mithilfe der AWS CLI identifizieren.

So identifizieren Sie freigegebene Parameter mithilfe der AWS CLI

Um freigegebene Parameter mit der AWS CLI zu identifizieren, können Sie zwischen dem Systems-Manager-`describe-parameters`-Befehl und dem AWS RAM-`list-resources`-Befehl wählen.

Wenn Sie die `--shared`-Option mit `describe-parameters` verwenden, gibt der Befehl die Parameter zurück, die mit Ihnen gemeinsam genutzt werden.

Im Folgenden wird ein Beispiel gezeigt:

```
aws ssm describe-parameters --shared
```

Zugreifen auf freigegebene Parameter

Verbraucher können mithilfe der AWS-Befehlszeilentools und AWS-SDKs auf gemeinsam genutzte Parameter zugreifen. Bei Verbraucherkonten sind Parameter, die mit diesem Konto gemeinsam genutzt werden, nicht auf der Seite Meine Parameter enthalten.

CLI-Beispiel: Zugreifen auf gemeinsam genutzte Parameterdetails mit dem AWS CLI

Um mit dem AWS CLI auf Details gemeinsam genutzter Parameter zuzugreifen, können Sie die Befehle get-parameters oder get-parameter verwenden. Sie müssen den vollständigen Parameter-ARN als angeben, um den `--name`-Parameter von einem anderen Konto abzurufen.

Im Folgenden wird ein Beispiel gezeigt.

```
aws ssm get-parameter \
    --name arn:aws:ssm:us-east-2:123456789012:parameter/MySharedParameter
```

Unterstützte und nicht unterstützte Integrationen für gemeinsam genutzte Parameter

Derzeit können Sie gemeinsam genutzte Parameter in den folgenden Integrationsszenarien verwenden:

- AWS CloudFormation-Vorlagenparameter
- Die AWS-Parameter und Secrets-Lambda-Erweiterung
- Amazon Elastic Compute Cloud (EC2)-Startvorlagen
- Werte für `ImageID` mit dem EC2-Befehl RunInstances zum Erstellen von Instances aus einem Amazon Machine Image (AMI)
- Abrufen von Parameterwerten in Runbooks für Automation, ein Tool in Systems Manager

Die folgenden Szenarien und integrierten Services unterstützen derzeit nicht die Verwendung von freigegebenen Parametern:

- Parameter in Befehlen in Run Command, einem Tool in Systems Manager
- AWS CloudFormation dynamische Verweise
- Der Wert der Umgebungsvariable in AWS CodeBuild
- Der Wert der Umgebungsvariable in AWS App Runner
- Der Wert eines Secrets in Amazon Elastic Container Service

Berechtigungssätze für die gemeinsame Nutzung von Parametern

Verbraucherkonten erhalten schreibgeschützten Zugriff auf die Parameter, die Sie mit ihnen teilen. Der Verbraucher kann den Parameter nicht aktualisieren oder löschen. Der Verbraucher kann den Parameter nicht mit einem dritten Konto teilen.

Wenn Sie eine Ressourcenfreigabe AWS Resource Access Manager für die gemeinsame Nutzung Ihrer Parameter erstellen, können Sie aus zwei von AWS verwalteten Berechtigungssätzen wählen, um diesen schreibgeschützten Zugriff zu gewähren:

AWSRAMDefaultPermissionSSMParameterReadOnly

Erlaubte Aktionen: `DescribeParameters`, `GetParameter`, `GetParameters`

AWSRAMPermissionSSMParameterReadOnlyWithHistory

Erlaubte Aktionen: `DescribeParameters`, `GetParameter`, `GetParameters`, `GetParameterHistory`

Wenn Sie die Schritte unter Erstellen einer gemeinsamen Ressource in AWS RAM im AWS RAM-Benutzerhandbuch ausführen, wählen Sie `Parameter Store Advanced Parameters` als Ressourcentyp und eine dieser verwalteten Berechtigungen aus, je nachdem, ob Benutzer den Parameterverlauf anzeigen sollen oder nicht.

> **Note**
>
> Wenn Sie gemeinsam genutzte Parameter programmgesteuert abrufen (z. B. mithilfe von AWS Lambda), müssen Sie möglicherweise allen IAM-Rollen, die API-Aktionen AWS Resource Access Manager aufrufen, die `ssm:GetResourcePolicies`- und `ssm:PutResourcePolicy`-Berechtigungen und hinzufügen.

Maximaler Durchsatz für freigegebene Parameter

Systems Manager begrenzt den maximalen Durchsatz (Transaktionen pro Sekunde) für die Operationen GetParameter und GetParameters. Der Durchsatz wird auf der Ebene der einzelnen Konten durchgesetzt. Daher kann jedes Konto, das einen gemeinsamen Parameter verwendet, seinen maximal zulässigen Durchsatz nutzen, ohne von anderen Konten beeinflusst zu werden. Weitere Informationen zum maximalen Durchsatz für Parameter finden Sie in den folgenden Themen:

- Erhöhen des Parameter Store-Durchsatzes
- Systems-Manager-Service Quotas im Allgemeine Amazon Web Services-Referenz.

Preisgestaltung für freigegebene Parameter

Kontoübergreifendes Teilen ist nur in der erweiterten Parameterstufe verfügbar. Für erweiterte Parameter fallen Gebühren zum aktuellen Preis für den Speicher und die API-Nutzung für jeden erweiterten Parameter an. Die Kosten für die Speicherung der erweiterten Parameter werden dem Eigentümerkonto angerechnet. Für jedes nutzende Konto, das einen API-Aufruf an einen gemeinsam genutzten erweiterten Parameter tätigt, wird die Nutzung des Parameters in Rechnung gestellt.

Wenn Konto A beispielsweise einen erweiterten Parameter, `MyAdvancedParameter`, erstellt, werden diesem Konto 0,05 USD pro Monat für die Speicherung des Parameters berechnet.

Konto A teilt dann `MyAdvancedParameter` mit Konto B und Konto C. Während eines Monats tätigen die drei Konten Anrufe an `MyAdvancedParameter`. In der folgenden Tabelle sind die Gebühren aufgeführt, die für sie je nach Anzahl der von ihnen getätigten Anrufe anfallen würden.

> **Note**
>
> Die Gebühren in der folgenden Tabelle dienen nur zur Veranschaulichung. Informationen zur Überprüfung der aktuellen Preise finden Sie unter AWS Systems Manager-Preise für Parameter Store.

Account	Anzahl der Aufrufe	Gebühren
Konto A (Besitzkonto)	10 000 Anrufe	• Erweiterter Parameter speicher für einen Monat: 0,05 USD • 10 000 Anrufe zu `MyAdvancedParameter` : 0,05 USD • Insgesamt: 0,10 USD
Konto B (Verbraucherkonto)	20 000 Anrufe	• 20 000 Anrufe zu `MyAdvancedParameter` : 0,10 USD • Insgesamt: 0,10 USD
Konto C (Verbraucherkonto)	30 000 Anrufe	• 30 000 Anrufe zu `MyAdvancedParameter` : 0,15 USD • Insgesamt: 0,15 USD

Kontoübergreifender Zugriff für geschlossene AWS-Konten

Wenn das AWS-Konto, das einen gemeinsamen Parameter besitzt, geschlossen wird, verlieren alle Benutzerkonten den Zugriff auf den freigegebenen Parameter. Wenn das Konto, das Eigentümer ist, innerhalb von 90 Tagen nach der Schließung des Kontos wieder geöffnet wird, erhalten die verbrauchenden Konten wieder Zugriff auf die zuvor freigegebenen Parameter. Weitere Informationen zur Wiedereröffnung eines Kontos während der Zeit nach der Schließung finden Sie unter Zugriff auf Ihr AWS-Konto, nachdem Sie es geschlossen haben im AWS -Kontenverwaltung-Referenzhandbuch.

Arbeiten mit Parametern in Parameter Store mithilfe von Run Command-Befehlen

Sie können mit Parametern in Run Command, einem Tool in AWS Systems Manager arbeiten. Weitere Informationen finden Sie unter AWS Systems Manager Run Command.

Ausführen eines String-Parameters mithilfe der Konsole

Das folgende Verfahren führt Sie durch die Schritte zum Ausführen eines Befehls, der einen `String`-Parameter verwendet.

Ausführen eines String-Parameters mithilfe der Parameter Store

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Befehl ausführen aus.
4. Wählen Sie in der Liste Command document (Befehlsdokument) die Option `AWS-RunPowerShellScript` (Windows) oder `AWS-RunShellScript` (Linux) aus.
5. Geben Sie für Command parameters (Befehlsparameter) Folgendes ein: **`echo {{ssm:parameter-name}}`**. Beispiel: **`echo {{ssm:/Test/helloWorld}}`**.
6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

> **Tip**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Weitere Parameter:

   - Geben Sie im Feld Kommentar Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (Sekunden) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

8. Für Ratenregelung:

   - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, sind diejenigen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle (hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, nicht die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die

IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

10. Aktivieren Sie das Kontrollkästchen SNS-Benachrichtigungen aktivieren im Abschnitt SNS-Benachrichtigungen, wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon-SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

11. Klicken Sie auf Ausführen.
12. Wählen Sie auf der Seite Command ID (Befehls-ID) im Bereich Targets and outputs (Ziele und Ausgaben) auf die Schaltfläche neben der ID eines Knotens, auf dem Sie den Befehl ausgeführt haben, und wählen Sie dann View output (Ausgabe anzeigen). Vergewissern Sie sich, dass der Befehl den Wert ausgibt, den Sie für den Parameter angegeben haben, z. B. **`This is my first parameter`**.

Ausführen eines Parameters mit dem AWS CLI

Beispiel 1: Einfacher Befehl

Der folgende Beispielbefehl enthält einen Systems Manager-Parameter mit der Bezeichnung `DNS-IP`. Der Wert dieses Parameters entspricht der IP-Adresse eines Knotens. In diesem Beispiel wird ein AWS Command Line Interface (AWS CLI) -Befehl verwendet, um den Parameterwert wiederzugeben.

Linux & macOS

```
aws ssm send-command \
    --document-name "AWS-RunShellScript" \
    --document-version "1" \
    --targets "Key=instanceids,Values=i-02573cafcfEXAMPLE" \
    --parameters "commands='echo {{ssm:DNS-IP}}'" \
    --timeout-seconds 600 \
    --max-concurrency "50" \
    --max-errors "0" \
    --region us-east-2
```

Windows

```
aws ssm send-command ^
    --document-name "AWS-RunPowerShellScript" ^
    --document-version "1" ^
    --targets "Key=instanceids,Values=i-02573cafcfEXAMPLE" ^
    --parameters "commands='echo {{ssm:DNS-IP}}'" ^
    --timeout-seconds 600 ^
    --max-concurrency "50" ^
    --max-errors "0" ^
    --region us-east-2
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "Command": {
        "CommandId": "c70a4671-8098-42da-b885-89716EXAMPLE",
        "DocumentName": "AWS-RunShellScript",
        "DocumentVersion": "1",
        "Comment": "",
        "ExpiresAfter": "2023-12-26T15:19:17.771000-05:00",
        "Parameters": {
            "commands": [
                "echo {{ssm:DNS-IP}}"
            ]
        },
        "InstanceIds": [],
        "Targets": [
            {
                "Key": "instanceids",
                "Values": [
                    "i-02573cafcfEXAMPLE"
                ]
            }
        ],
        "RequestedDateTime": "2023-12-26T14:09:17.771000-05:00",
        "Status": "Pending",
        "StatusDetails": "Pending",
        "OutputS3Region": "us-east-2",
        "OutputS3BucketName": "",
        "OutputS3KeyPrefix": "",
        "MaxConcurrency": "50",
```

```
        "MaxErrors": "0",
        "TargetCount": 0,
        "CompletedCount": 0,
        "ErrorCount": 0,
        "DeliveryTimedOutCount": 0,
        "ServiceRole": "",
        "NotificationConfig": {
            "NotificationArn": "",
            "NotificationEvents": [],
            "NotificationType": ""
        },
        "CloudWatchOutputConfig": {
            "CloudWatchLogGroupName": "",
            "CloudWatchOutputEnabled": false
        },
        "TimeoutSeconds": 600,
        "AlarmConfiguration": {
            "IgnorePollAlarmFailure": false,
            "Alarms": []
        },
        "TriggeredAlarms": []
    }
}
```

Nachdem die Ausführung eines Befehls abgeschlossen ist, können Sie mit den folgenden Befehlen weitere Informationen dazu anzeigen:

- get-command-invocation – Zeigt detaillierte Informationen zur Befehlsausführung an.
- list-command-invocations – Zeigt den Status der Befehlsausführung auf einem bestimmten verwalteten Knoten an.
- list-commands – Zeigt den Status der Befehlsausführung in verwalteten Knoten an.

Beispiel 2: Einen **`SecureString`**-Parameterwert entschlüsseln

Der nächste Beispielbefehl verwendet einen `SecureString` Parameter mit dem Namen SecurePassword. Mit dem im `parameters`-Feld verwendeten Befehl wird der Wert des `SecureString`-Parameters abgerufen und entschlüsselt. Anschließend wird das lokale Administratorpasswort zurückgesetzt, ohne dass das Passwort als Klartext übergeben wird.

Linux

```
aws ssm send-command \
        --document-name "AWS-RunShellScript" \
        --document-version "1" \
        --targets "Key=instanceids,Values=i-02573cafcfEXAMPLE" \
        --parameters '{"commands":["secure=$(aws ssm get-parameters --names
 SecurePassword --with-decryption --query Parameters[0].Value --output text --region
 us-east-2)","echo $secure | passwd myuser --stdin"]}' \
        --timeout-seconds 600 \
        --max-concurrency "50" \
        --max-errors "0" \
        --region us-east-2
```

Windows

```
aws ssm send-command ^
        --document-name "AWS-RunPowerShellScript" ^
        --document-version "1" ^
        --targets "Key=instanceids,Values=i-02573cafcfEXAMPLE" ^
        --parameters "commands=['$secure = (Get-SSMParameterValue -Names
 SecurePassword -WithDecryption $True).Parameters[0].Value','net user administrator
 $secure']" ^
        --timeout-seconds 600 ^
        --max-concurrency "50" ^
        --max-errors "0" ^
        --region us-east-2
```

Beispiel 3: Auf einen Parameter in einem SSM-Dokument verweisen

Sie können Systems Manager-Parameter auch im Abschnitt Parameters in einem SSM-Dokument referenzieren, wie im folgenden Beispiel gezeigt.

```
{
   "schemaVersion":"2.0",
   "description":"Sample version 2.0 document v2",
   "parameters":{
      "commands" : {
        "type": "StringList",
        "default": ["{{ssm:parameter-name}}"]
      }
    },
```

```
    "mainSteps":[
       {
          "action":"aws:runShellScript",
          "name":"runShellScript",
          "inputs":{
             "runCommand": "{{commands}}"
          }
       }
    ]
}
```

Verwechseln Sie die ähnliche Syntax für lokale Parameter, die im `runtimeConfig`-Abschnitt von SSM-Dokumenten verwendet werden, nicht mit Parameter Store-Parametern. Ein lokaler Parameter ist nicht dasselbe wie ein Systems Manager-Parameter. Sie können lokale Parameter daran erkennen, dass diese (im Gegensatz zu Systems Manager-Parametern) über kein `ssm:`-Präfix verfügen.

```
"runtimeConfig":{
        "aws:runShellScript":{
            "properties":[
                {
                    "id":"0.aws:runShellScript",
                    "runCommand":"{{ commands }}",
                    "workingDirectory":"{{ workingDirectory }}",
                    "timeoutSeconds":"{{ executionTimeout }}"
```

> **Note**
>
> SSM-Dokumente unterstützen keine Referenzen auf `SecureString`-Parameter. Um `SecureString`-Parameter beispielsweise mit Run Command verwenden zu können, müssen Sie daher den Parameterwert vor der Übergabe an Run Command wie in den folgenden Beispielen gezeigt abrufen.
>
> Linux & macOS
>
> ```
> value=$(aws ssm get-parameters --names parameter-name --with-decryption)
> ```
>
> ```
> aws ssm send-command \
>     --name AWS-JoinDomain \
>     --parameters password=$value \
> ```

```
    --instance-id instance-id
```

Windows

```
aws ssm send-command ^
    --name AWS-JoinDomain ^
    --parameters password=$value ^
    --instance-id instance-id
```

Powershell

```
$secure = (Get-SSMParameterValue -Names parameter-name -WithDecryption
 $True).Parameters[0].Value | ConvertTo-SecureString -AsPlainText -Force
```

```
$cred = New-Object System.Management.Automation.PSCredential -
argumentlist user-name,$secure
```

Verwendung der nativen Parameterunterstützung Parameter Store für Amazon Machine Image IDs

Wenn Sie einen `String`-Parameter erstellen, können Sie einen Datentyp als `aws:ec2:image` angeben, um sicherzustellen, dass der eingegebene Parameterwert ein gültiges Amazon Machine Image (AMI)-ID-Format aufweist.

Durch die Unterstützung von AMI-ID-Formaten können Sie vermeiden, dass alle Skripts und Vorlagen jedes Mal mit einer neuen ID aktualisiert werden, wenn sich das AMI ändert, das Sie in Ihren Prozessen verwenden möchten. Sie können einen Parameter mit dem Datentyp `aws:ec2:image` erstellen und für seinen Wert die ID eines AMI eingeben. Dies ist das AMI, von dem Sie neue Instances erstellen möchten. Anschließend verweisen Sie in Ihren Vorlagen, Befehlen und Skripts auf diesen Parameter.

Sie können beispielsweise den Parameter angeben, der Ihre bevorzugte AMI ID enthält, wenn Sie den `run-instances` Befehl Amazon Elastic Compute Cloud (Amazon EC2) ausführen.

> **Note**
>
> Der Benutzer, der diesen Befehl ausführt, muss über AWS Identity and Access Management (IAM-) Berechtigungen verfügen, die den `ssm:GetParameters` API-Vorgang einschließen,

damit der Parameterwert validiert werden kann. Andernfalls schlägt die Parametererstellung fehl.

Linux & macOS

```
aws ec2 run-instances \
    --image-id resolve:ssm:/golden-ami \
    --count 1 \
    --instance-type t2.micro \
    --key-name my-key-pair \
    --security-groups my-security-group
```

Windows

```
aws ec2 run-instances ^
    --image-id resolve:ssm:/golden-ami ^
    --count 1 ^
    --instance-type t2.micro ^
    --key-name my-key-pair ^
    --security-groups my-security-group
```

Sie können auch Ihre bevorzugte Option wählenAMI, wenn Sie eine Instance mit der EC2 Amazon-Konsole erstellen. Weitere Informationen finden Sie unter Verwenden eines Systems Manager Manager-Parameters AMI im EC2 Amazon-Benutzerhandbuch.

Wenn Sie ein anderes AMI in Ihrem Workflow zur Instance-Erstellung verwenden möchten, müssen Sie nur den Parameter mit dem neuen AMI-Wert aktualisieren und Parameter Store überprüft erneut, ob die ID im richtigen Format eingegeben wurde.

Erteilen von Berechtigungen zum Erstellen eines Parameters aus dem Datentyp **`aws:ec2:image`**

Mithilfe von AWS Identity and Access Management (IAM-) Richtlinien können Sie Benutzern den Zugriff auf Parameter Store API-Operationen und -Inhalte gewähren oder einschränken.

Um einen `aws:ec2:image`-Datentypparameter zu erstellen, muss der Benutzer `ssm:PutParameter`- sowohl als auch über `ec2:DescribeImages`-Berechtigungen verfügen.

Die folgende Beispielrichtlinie erteilt Benutzern die Berechtigung zum Aufrufen der API-Operation `PutParameter` für `aws:ec2:image`. Dies bedeutet, dass der Benutzer einen Parameter des Datentyps `aws:ec2:image`zum System hinzufügen kann.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ssm:PutParameter",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeImages",
            "Resource": "*"
        }
    ]
}
```

Funktionsweise der AMI-Formatvalidierung

Wenn Sie `aws:ec2:image` als Datentyp für einen Parameter angeben, erstellt Systems Manager den Parameter nicht sofort. Stattdessen wird eine asynchrone Validierungsoperation ausgeführt, um sicherzustellen, dass der Parameterwert die Formatierungsanforderungen für eine AMI-ID erfüllt und das angegebene AMI in Ihrem AWS-Konto verfügbar ist.

Eine Parameterversionsnummer wird möglicherweise generiert, bevor die Validierungsoperation abgeschlossen ist. Die Operation wird möglicherweise nicht abgeschlossen, auch wenn eine Parameterversionsnummer geniert wird.

Um zu überprüfen, ob Ihre Parameter erfolgreich erstellt wurden, empfehlen wir, Amazon EventBridge zu verwenden, um Ihnen Benachrichtigungen über Ihre `create` und die `update` Parameteroperationen zu senden. Diese Benachrichtigungen melden, ob eine Parameteroperation erfolgreich war oder nicht. Wenn eine Operation fehlschlägt, enthält die Benachrichtigung eine Fehlermeldung, die den Grund für den Fehler angibt.

```
{
```

```
    "version": "0",
    "id": "eed4a719-0fa4-6a49-80d8-8ac65EXAMPLE",
    "detail-type": "Parameter Store Change",
    "source": "aws.ssm",
    "account": "111122223333",
    "time": "2020-05-26T22:04:42Z",
    "region": "us-east-2",
    "resources": [
        "arn:aws:ssm:us-east-2:111122223333:parameter/golden-ami"
    ],
    "detail": {
        "exception": "Unable to Describe Resource",
        "dataType": "aws:ec2:image",
        "name": "golden-ami",
        "type": "String",
        "operation": "Create"
    }
}
```


Informationen zum Abonnieren von Parameter Store Veranstaltungen in finden Sie EventBridge unterEinrichten von Benachrichtigungen oder Auslösen von Aktionen basierend auf Parameter Store-Ereignissen.

Löschen von Parametern in Parameter Store

In diesem Thema wird beschrieben, wie Sie Parameter löschen, die Sie in Parameter Store, einem Tool in AWS Systems Manager, erstellt haben.

> **Warning**
>
> Durch das Löschen eines Parameters werden alle Versionen davon entfernt. Nach dem Löschen können der Parameter und seine Versionen nicht wiederhergestellt werden.

So löschen Sie einen Parameter mithilfe der Konsole

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Aktivieren Sie auf der Registerkarte My parameters (Meine Parameter) das Kontrollkästchen neben jedem zu löschenden Parameter.

4. Wählen Sie Löschen aus.
5. Wählen Sie im Bestätigungs-Dialogfeld die Option Delete parameters (Parameter löschen).

So löschen Sie einen Parameter mithilfe der AWS CLI

- Führen Sie den folgenden Befehl aus:

```
aws ssm delete-parameter --name "my-parameter"
```

  Ersetzen Sie *my-parameter* durch den Namen Ihres Parameters, der gelöscht werden soll.

  Informationen zu allen verfügbaren Optionen, die Sie mit dem Befehl `delete-parameter` verwenden können, finden Sie unter delete-parameter im Abschnitt AWS Systems Manager der AWS CLI-Befehlsreferenz.

Arbeiten mit öffentlichen Parametern in Parameter Store

Einige AWS-Services veröffentlichen Informationen über allgemeine Artefakte als AWS Systems Manager öffentliche Parameter. Der Amazon Elastic Compute Cloud (Amazon EC2)-Service beispielsweise veröffentlicht Informationen zu Amazon Machine Images (AMIs) als öffentliche Parameter.

Themen in diesem Leitfaden

- Entdecken von öffentlichen Parametern in Parameter Store
- Aufrufen von öffentlichen AMI-Parametern in Parameter Store
- Aufrufen der ECS-optimierten öffentlichen AMI-Parameter in Parameter Store
- Aufrufen der EKS-optimierten öffentlichen AMI-Parameter in Parameter Store
- Aufrufen öffentlicher Parameter für AWS-Services, Regionen, Endpunkte, Availability Zones, lokale Zonen und Wavelength Zones in Parameter Store

Verwandte AWS-Blog-Beiträge

- Abfragen nach AWS-Regionen, Endpunkten und mehr mithilfe von AWS Systems ManagerParameter Store
- Abfragen nach den aktuellen Amazon Linux AMI-IDs mit AWS Systems ManagerParameter Store

- Abfragen nach dem aktuellen Windows-AMI mit AWS Systems ManagerParameter Store

Entdecken von öffentlichen Parametern in Parameter Store

Sie können mithilfe der Parameter Store-Konsole oder der AWS Command Line Interface nach öffentlichen Parametern suchen.

Ein öffentlicher Parametername beginnt mit `aws/service/list`. Der nächste Teil des Namens entspricht dem Service, dem dieser Parameter gehört.

Im Folgenden finden Sie eine unvollständige Liste von Ressourcen AWS-Services und anderen Ressourcen, die öffentliche Parameter bereitstellen:

- `ami-amazon-linux-latest`
- `ami-windows-latest`
- `ec2-macos`
- `appmesh`
- `aws-for-fluent-bit`
- `aws-sdk-pandas`
- `bottlerocket`
- `canonical`
- `cloud9`
- `datasync`
- `deeplearning`
- `ecs`
- `eks`
- `fis`
- `freebsd`
- `global-infrastructure`
- `marketplace`
- `neuron`
- `powertools`
- `sagemaker-distribution`
- `storagegateway`

Nicht alle öffentlichen Parameter werden für alle veröffentlicht AWS-Region.

Auffinden von öffentlichen Parametern mithilfe der Parameter Store-Konsole

Sie müssen mindestens einen Parameter in Ihrem AWS-Konto und haben, AWS-Region bevor Sie mit der Konsole nach öffentlichen Parametern suchen können.

Auffinden von öffentlichen Parametern mithilfe der Konsole

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store aus.
3. Wählen Sie die Registerkarte Öffentliche Parameter aus.
4. Wählen Sie das Dropdown-Menü Einen Service auswählen aus. Wählen Sie den Service aus, dessen Parameter Sie verwenden möchten.
5. (Optional) Filtern Sie die Parameter des ausgewählten Services, indem Sie weitere Informationen in die Suchleiste eingeben.
6. Wählen Sie den zu verwendenden öffentlichen Parameter aus.

Suchen nach öffentlichen Parametern mit dem AWS CLI

Verwenden Sie `describe-parameters`, um öffentliche Parameter zu entdecken.

Verwenden Sie `get-parameters-by-path`, um den tatsächlichen Pfad für einen Service zu erhalten, der unter `/aws/service/list` gelistet ist. Um den Pfad des Services abzurufen, entfernen Sie `/list` aus dem Pfad. Beispielsweise wird `/aws/service/list/ecs` zu `/aws/service/ecs`.

Um eine Liste von öffentlichen Parametern abzurufen, die verschiedenen Services in Parameter Store gehören, führen Sie den folgenden Befehl aus.

```
aws ssm get-parameters-by-path --path /aws/service/list
```

Der Befehl gibt Informationen wie die folgenden zurück. Dieses Beispiel wurde aus Platzgründen gekürzt.

```
{
    "Parameters": [
        {
```

```
            "Name": "/aws/service/list/ami-al-latest",
            "Type": "String",
            "Value": "/aws/service/ami-al-latest/",
            "Version": 1,
            "LastModifiedDate": "2021-01-29T10:25:10.902000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/list/ami-al-latest",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/list/ami-windows-latest",
            "Type": "String",
            "Value": "/aws/service/ami-windows-latest/",
            "Version": 1,
            "LastModifiedDate": "2021-01-29T10:25:12.567000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/list/ami-windows-
latest",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/list/aws-storage-gateway-latest",
            "Type": "String",
            "Value": "/aws/service/aws-storage-gateway-latest/",
            "Version": 1,
            "LastModifiedDate": "2021-01-29T10:25:09.903000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/list/aws-storage-
gateway-latest",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/list/global-infrastructure",
            "Type": "String",
            "Value": "/aws/service/global-infrastructure/",
            "Version": 1,
            "LastModifiedDate": "2021-01-29T10:25:11.901000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/list/global-
infrastructure",
            "DataType": "text"
        }
    ]
}
```


Wenn Sie Parameter anzeigen möchten, die einem bestimmten Service gehören, wählen Sie den Service aus der Liste aus, die nach dem Ausführen des vorherigen Befehls erstellt wurde. Dann

machen Sie einen `get-parameters-by-path`-Aufruf nach dem Namen Ihres gewünschten Services.

Beispiel, `/aws/service/global-infrastructure`. Der Pfad kann einstufig sein (ruft nur Parameter auf, die genau den angegebenen Werten entsprechen) oder rekursiv (enthält Elemente im Pfad über das hinaus, was Sie angegeben haben).

> **Note**
>
> Der `/aws/service/global-infrastructure`-Pfad wird nicht für Abfragen in allen Regionen unterstützt. Weitere Informationen finden Sie unter Aufrufen öffentlicher Parameter für AWS-Services, Regionen, Endpunkte, Availability Zones, lokale Zonen und Wavelength Zones in Parameter Store.

Wenn für den von Ihnen angegebenen Dienst keine Ergebnisse zurückgegeben werden, fügen Sie das `--recursive`-Flag hinzu, und führen Sie den Befehl erneut aus.

```
aws ssm get-parameters-by-path --path /aws/service/global-infrastructure
```

Dadurch werden alle Parameter im Besitz von `global-infrastructure` ausgegeben. Im Folgenden wird ein -Beispiel gezeigt.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/global-infrastructure/current-region",
            "Type": "String",
            "LastModifiedDate": "2019-06-21T05:15:34.252000-07:00",
            "Version": 1,
            "Tier": "Standard",
            "Policies": [],
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/version",
            "Type": "String",
            "LastModifiedDate": "2019-02-04T06:59:32.875000-08:00",
            "Version": 1,
            "Tier": "Standard",
            "Policies": [],
```

```
            "DataType": "text"
        }
    ]
}
```

Sie können auch Parameter, die einem bestimmten Service gehören, anzeigen, indem Sie den `Option:BeginsWith`-Filter verwenden.

```
aws ssm describe-parameters --parameter-filters "Key=Name, Option=BeginsWith, Values=/
aws/service/ami-amazon-linux-latest"
```

Der Befehl gibt Informationen wie die folgenden zurück. In diesem Beispiel wurde die Ausgabe aus Platzgründen gekürzt.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-ebs",
            "Type": "String",
            "LastModifiedDate": "2021-01-26T13:39:40.686000-08:00",
            "Version": 25,
            "Tier": "Standard",
            "Policies": [],
            "DataType": "text"
        },
        {
            "Name": "/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-gp2",
            "Type": "String",
            "LastModifiedDate": "2021-01-26T13:39:40.807000-08:00",
            "Version": 25,
            "Tier": "Standard",
            "Policies": [],
            "DataType": "text"
        },
        {
            "Name": "/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-s3",
            "Type": "String",
            "LastModifiedDate": "2021-01-26T13:39:40.920000-08:00",
            "Version": 25,
            "Tier": "Standard",
            "Policies": [],
            "DataType": "text"
        }
```

```
    ]
}
```

> **Note**
>
> Die zurückgegebenen Parameter können unterschiedlich sein, wenn Sie `Option=BeginsWith` verwenden, da es ein anderes Suchmuster verwendet.

Aufrufen von öffentlichen AMI-Parametern in Parameter Store

Öffentliche Parameter von Amazon Elastic Compute Cloud Amazon Machine Image (Amazon EC2AMI) () sind für Amazon Linux 2, Amazon Linux 2023 (AL2023) und Windows Server über die folgenden Pfade verfügbar: macOS

- Amazon Linux 2 und Amazon Linux 2023: `/aws/service/ami-amazon-linux-latest`
- macOS: `/aws/service/ec2-macos`
- Windows Server: `/aws/service/ami-windows-latest`

Aufrufen öffentlicher AMI-Parameter für Amazon Linux 2 und Amazon Linux 2023

Sie können eine Liste aller aktuellen Versionen von Amazon Linux 2 und Amazon Linux 2023 (AL2023) AMIs anzeigen, AWS-Region indem Sie den folgenden Befehl in der AWS Command Line Interface (AWS CLI) verwenden.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/ami-amazon-linux-latest \
    --query 'Parameters[].Name'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/ami-amazon-linux-latest ^
    --query Parameters[].Name
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
[
    "/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-arm64",
    "/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64",
    "/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-kernel-6.1-arm64",
    "/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-kernel-6.1-x86_64",
    "/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-kernel-default-arm64",
    "/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-gp2",
    "/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-s3",
    "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-ebs",
    "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2",
    "/aws/service/ami-amazon-linux-latest/amzn2-ami-kernel-5.10-hvm-x86_64-ebs",
    "/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-arm64",
    "/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64",
    "/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-kernel-default-x86_64",
    "/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86_64-ebs",
    "/aws/service/ami-amazon-linux-latest/amzn-ami-minimal-hvm-x86_64-ebs",
    "/aws/service/ami-amazon-linux-latest/amzn-ami-minimal-hvm-x86_64-s3",
    "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-arm64-gp2",
    "/aws/service/ami-amazon-linux-latest/amzn2-ami-kernel-5.10-hvm-arm64-gp2",
    "/aws/service/ami-amazon-linux-latest/amzn2-ami-kernel-5.10-hvm-x86_64-gp2",
    "/aws/service/ami-amazon-linux-latest/amzn2-ami-minimal-hvm-arm64-ebs",
    "/aws/service/ami-amazon-linux-latest/amzn2-ami-minimal-hvm-x86_64-ebs"
]
```

Mit dem folgenden Befehl können Sie sich Details zu diesen Ressourcen anzeigen lassenAMIs, einschließlich der Amazon Resource Names (ARNs). AMI IDs

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path "/aws/service/ami-amazon-linux-latest" \
    --region region
```

Windows

```
aws ssm get-parameters-by-path ^
    --path "/aws/service/ami-amazon-linux-latest" ^
    --region region
```

*region*stellt den Bezeichner für eine Region dar AWS Systems Manager, die von AWS-Region unterstützt wird, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten

region Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

Der Befehl gibt Informationen wie die folgenden zurück. In diesem Beispiel wurde die Ausgabe aus Platzgründen gekürzt.

```
{
    "Parameters": [
         {
            "Name": "/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-arm64",
            "Type": "String",
            "Value": "ami-0b1b8b24a6c8e5d8b",
            "Version": 69,
            "LastModifiedDate": "2024-03-13T14:05:09.583000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ami-amazon-linux-
latest/al2023-ami-kernel-6.1-arm64",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-
x86_64",
            "Type": "String",
            "Value": "ami-0e0bf53f6def86294",
            "Version": 69,
            "LastModifiedDate": "2024-03-13T14:05:09.890000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ami-amazon-linux-
latest/al2023-ami-kernel-6.1-x86_64",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-
kernel-6.1-arm64",
            "Type": "String",
            "Value": "ami-09951bb66f9e5b5a5",
            "Version": 69,
            "LastModifiedDate": "2024-03-13T14:05:10.197000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ami-amazon-linux-
latest/al2023-ami-minimal-kernel-6.1-arm64",
            "DataType": "text"
        }
    ]
}
```

Sie können Details zu einem bestimmten Objekt anzeigen, AMI indem Sie den GetParametersAPI-Vorgang mit dem vollständigen AMI Namen, einschließlich des Pfads, verwenden. Hier sehen Sie ein Beispiel für einen Befehl.

Linux & macOS

```
aws ssm get-parameters \
    --names /aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-arm64 \
    --region us-east-2
```

Windows

```
aws ssm get-parameters ^
    --names /aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-arm64 ^
    --region us-east-2
```

Der Befehl gibt die folgenden Informationen zurück.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-arm64",
            "Type": "String",
            "Value": "ami-0b1b8b24a6c8e5d8b",
            "Version": 69,
            "LastModifiedDate": "2024-03-13T14:05:09.583000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ami-amazon-linux-
latest/al2023-ami-kernel-6.1-arm64",
            "DataType": "text"
        }
    ],
    "InvalidParameters": []
}
```

Aufrufen von öffentlichen AMI-Parametern für macOS

Sie können eine Liste aller macOS AMIs aktuellen Einträge anzeigen, AWS-Region indem Sie den folgenden Befehl in der verwenden AWS CLI.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/ec2-macos\
    --query 'Parameters[].Name'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/ec2-macos ^
    --query Parameters[].Name
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
[
    "/aws/service/ec2-macos/sonoma/x86_64_mac/latest/image_id",
    "/aws/service/ec2-macos/ventura/x86_64_mac/latest/image_id",
    "/aws/service/ec2-macos/sonoma/arm64_mac/latest/image_id",
    "/aws/service/ec2-macos/ventura/arm64_mac/latest/image_id"
]
```

Mit dem folgenden Befehl können Sie sich Details zu diesen Ressourcen anzeigen lassenAMIs, einschließlich der Amazon Resource Names (ARNs). AMI IDs

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path "/aws/service/ec2-macos" \
    --region region
```

Windows

```
aws ssm get-parameters-by-path ^
    --path "/aws/service/ec2-macos" ^
    --region region
```

*region*stellt den Bezeichner für eine Region dar AWS Systems Manager, die von AWS-Region unterstützt wird, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten

region Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

Der Befehl gibt Informationen wie die folgenden zurück. In diesem Beispiel wurde die Ausgabe aus Platzgründen gekürzt.

```
{
    "Parameters": [
        ...sample results pending...
    ]
}
```

Sie können Details zu einem bestimmten Objekt anzeigen, AMI indem Sie den GetParametersAPI-Vorgang mit dem vollständigen AMI Namen, einschließlich des Pfads, verwenden. Hier sehen Sie ein Beispiel für einen Befehl.

Linux & macOS

```
aws ssm get-parameters \
    --names /aws/service/ec2-macos/...pending... \
    --region us-east-2
```

Windows

```
aws ssm get-parameters ^
     --names /aws/service/ec2-macos/...pending... ^
    --region us-east-2
```

Der Befehl gibt die folgenden Informationen zurück.

```
{
    "Parameters": [
         ...sample results pending...
    ],
    "InvalidParameters": []
}
```

Aufrufen von öffentlichen AMI-Parametern für Windows Server

Sie können eine Liste aller Windows Server AMIs aktuellen Einträge anzeigen, AWS-Region indem Sie den folgenden Befehl in der verwenden AWS CLI.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/ami-windows-latest \
    --query 'Parameters[].Name'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/ami-windows-latest ^
    --query Parameters[].Name
```

Der Befehl gibt Informationen wie die folgenden zurück. In diesem Beispiel wurde die Ausgabe aus Platzgründen gekürzt.

```
[
    "/aws/service/ami-windows-latest/EC2LaunchV2-Windows_Server-2016-English-Full-
Base",
    "/aws/service/ami-windows-latest/Windows_Server-2016-English-Full-
SQL_2014_SP3_Enterprise",
    "/aws/service/ami-windows-latest/Windows_Server-2016-German-Full-Base",
    "/aws/service/ami-windows-latest/Windows_Server-2016-Japanese-Full-
SQL_2016_SP3_Standard",
    "/aws/service/ami-windows-latest/Windows_Server-2016-Japanese-Full-SQL_2017_Web",
    "/aws/service/ami-windows-latest/Windows_Server-2019-English-Core-
EKS_Optimized-1.25",
    "/aws/service/ami-windows-latest/Windows_Server-2019-Italian-Full-Base",
    "/aws/service/ami-windows-latest/Windows_Server-2022-Japanese-Full-
SQL_2019_Enterprise",
    "/aws/service/ami-windows-latest/Windows_Server-2022-Portuguese_Brazil-Full-Base",
    "/aws/service/ami-windows-latest/amzn2-ami-hvm-2.0.20191217.0-x86_64-gp2-mono",
    "/aws/service/ami-windows-latest/Windows_Server-2016-English-Deep-Learning",
    "/aws/service/ami-windows-latest/Windows_Server-2016-Japanese-Full-
SQL_2016_SP3_Web",
    "/aws/service/ami-windows-latest/Windows_Server-2016-Korean-Full-Base",
    "/aws/service/ami-windows-latest/Windows_Server-2019-English-STIG-Core",
    "/aws/service/ami-windows-latest/Windows_Server-2019-French-Full-Base",
```

```
    "/aws/service/ami-windows-latest/Windows_Server-2019-Japanese-Full-
SQL_2017_Enterprise",
    "/aws/service/ami-windows-latest/Windows_Server-2019-Korean-Full-Base",
    "/aws/service/ami-windows-latest/Windows_Server-2022-English-Full-SQL_2022_Web",
    "/aws/service/ami-windows-latest/Windows_Server-2022-Italian-Full-Base",
    "/aws/service/ami-windows-latest/amzn2-x86_64-SQL_2019_Express",
    "/aws/service/ami-windows-latest/EC2LaunchV2-Windows_Server-2016-English-Core-
Base",
    "/aws/service/ami-windows-latest/Windows_Server-2016-English-Full-
SQL_2019_Enterprise",
    "/aws/service/ami-windows-latest/Windows_Server-2016-English-Full-
SQL_2019_Standard",
    "/aws/service/ami-windows-latest/Windows_Server-2016-Portuguese_Portugal-Full-
Base",
    "/aws/service/ami-windows-latest/Windows_Server-2019-English-Core-
EKS_Optimized-1.24",
    "/aws/service/ami-windows-latest/Windows_Server-2019-English-Deep-Learning",
    "/aws/service/ami-windows-latest/Windows_Server-2019-English-Full-SQL_2017_Web",
    "/aws/service/ami-windows-latest/Windows_Server-2019-Hungarian-Full-Base
]
```

Mit dem folgenden Befehl können Sie sich Details zu diesen Ressourcen anzeigen lassenAMIs, einschließlich der Amazon Resource Names (ARNs). AMI IDs

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path "/aws/service/ami-windows-latest" \
    --region region
```

Windows

```
aws ssm get-parameters-by-path ^
    --path "/aws/service/ami-windows-latest" ^
    --region region
```

*region*stellt den Bezeichner für eine Region dar AWS Systems Manager, die von AWS-Region unterstützt wird, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

Der Befehl gibt Informationen wie die folgenden zurück. In diesem Beispiel wurde die Ausgabe aus Platzgründen gekürzt.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/ami-windows-latest/EC2LaunchV2-Windows_Server-2016-
English-Full-Base",
            "Type": "String",
            "Value": "ami-0a30b2e65863e2d16",
            "Version": 36,
            "LastModifiedDate": "2024-03-15T15:58:37.976000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ami-windows-latest/
EC2LaunchV2-Windows_Server-2016-English-Full-Base",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/ami-windows-latest/Windows_Server-2016-English-Full-
SQL_2014_SP3_Enterprise",
            "Type": "String",
            "Value": "ami-001f20c053dd120ce",
            "Version": 69,
            "LastModifiedDate": "2024-03-15T15:53:58.905000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ami-windows-latest/
Windows_Server-2016-English-Full-SQL_2014_SP3_Enterprise",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/ami-windows-latest/Windows_Server-2016-German-Full-
Base",
            "Type": "String",
            "Value": "ami-063be4935453e94e9",
            "Version": 102,
            "LastModifiedDate": "2024-03-15T15:51:12.003000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ami-windows-latest/
Windows_Server-2016-German-Full-Base",
            "DataType": "text"
        }
    ]
}
```

Sie können Details zu einem bestimmten Objekt anzeigen, AMI indem Sie den GetParametersAPI-Vorgang mit dem vollständigen AMI Namen, einschließlich des Pfads, verwenden. Hier sehen Sie ein Beispiel für einen Befehl.

Linux & macOS

```
aws ssm get-parameters \
    --names /aws/service/ami-windows-latest/EC2LaunchV2-Windows_Server-2016-English-
Full-Base \
    --region us-east-2
```

Windows

```
aws ssm get-parameters ^
    --names /aws/service/ami-windows-latest/EC2LaunchV2-Windows_Server-2016-English-
Full-Base ^
    --region us-east-2
```

Der Befehl gibt die folgenden Informationen zurück.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/ami-windows-latest/EC2LaunchV2-Windows_Server-2016-
English-Full-Base",
            "Type": "String",
            "Value": "ami-0a30b2e65863e2d16",
            "Version": 36,
            "LastModifiedDate": "2024-03-15T15:58:37.976000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ami-windows-latest/
EC2LaunchV2-Windows_Server-2016-English-Full-Base",
            "DataType": "text"
        }
    ],
    "InvalidParameters": []
}
```

Aufrufen der ECS-optimierten öffentlichen AMI-Parameter in Parameter Store

Der Amazon Elastic Container Service (Amazon ECS)-Service veröffentlicht den Namen des neuesten von Amazon ECS optimierten Amazon Machine Images (AMIs) als öffentlichen Parameter.

Benutzer sollten dieses AMI beim Erstellen eines neuen Amazon Elastic Compute Cloud (Amazon EC2)-Clusters für Amazon ECS verwenden, da das optimierte AMIs Fehlerbehebungen und Funktionsaktualisierungen enthält.

Führen Sie den folgenden Befehl aus, um den Namen des neuesten von Amazon ECS optimierten AMI für Amazon Linux 2 anzuzeigen. Befehle für andere Betriebssysteme finden Sie unter Amazon ECS-optimierte AMI-Metadaten abrufen im Entwicklerhandbuch für Amazon Elastic Container Service.

Linux & macOS

```
aws ssm get-parameters \
    --names /aws/service/ecs/optimized-ami/amazon-linux-2/recommended
```

Windows

```
aws ssm get-parameters ^
    --names /aws/service/ecs/optimized-ami/amazon-linux-2/recommended
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/recommended",
            "Type": "String",
            "Value": "{\"schema_version\":1,\"image_name\":\"amzn2-ami-ecs-
hvm-2.0.20210929-x86_64-ebs\",\"image_id\":\"ami-0c38a2329ed4dae9a\",\"os\":\"Amazon
 Linux 2\",\"ecs_runtime_version\":\"Docker version 20.10.7\",\"ecs_agent_version\":
\"1.55.4\"}",
            "Version": 73,
            "LastModifiedDate": "2021-10-06T16:35:10.004000-07:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/recommended",
            "DataType": "text"
        }
    ],
    "InvalidParameters": []
}
```

Aufrufen der EKS-optimierten öffentlichen AMI-Parameter in Parameter Store

Der Amazon Elastic Kubernetes Service (Amazon EKS)-Service veröffentlicht den Namen des neuesten von Amazon EKS optimiertenAmazon Machine Image (AMI) als öffentlichen Parameter. Benutzer sollten beim Hinzufügen von Knoten zu einem Amazon EKS-Cluster dieses AMI verwenden, da neue Versionen Kubernetes-Patches und -Sicherheitsupdates enthalten. Um sicherzustellen, dass Sie das neueste AMI verwenden, mussten Sie bisher die Amazon EKS-Dokumentation überprüfen und Bereitstellungsvorlagen oder Ressourcen manuell mit der neuen AMI-ID aktualisieren.

Führen Sie den folgenden Befehl aus, um den Namen des neuesten von Amazon EKS optimierten AMI für Amazon Linux 2 anzuzeigen.

Linux & macOS

```
aws ssm get-parameters \
    --names /aws/service/eks/optimized-ami/1.14/amazon-linux-2/recommended
```

Windows

```
aws ssm get-parameters ^
    --names /aws/service/eks/optimized-ami/1.14/amazon-linux-2/recommended
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/eks/optimized-ami/1.14/amazon-linux-2/recommended",
            "Type": "String",
            "Value": "{\"schema_version\":\"2\",\"image_id\":\"ami-08984d8491de17ca0\",
\"image_name\":\"amazon-eks-node-1.14-v20201007\",\"release_version\":
\"1.14.9-20201007\"}",
            "Version": 24,
            "LastModifiedDate": "2020-11-17T10:16:09.971000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/eks/optimized-
ami/1.14/amazon-linux-2/recommended",
            "DataType": "text"
        }
    ],
    "InvalidParameters": []
```

```
}
```

Aufrufen öffentlicher Parameter für AWS-Services, Regionen, Endpunkte, Availability Zones, lokale Zonen und Wavelength Zones in Parameter Store

Sie können öffentliche Parameter für AWS-Region, Service, Endpunkt, Availability und Wavelength Zone über den folgenden Pfad aufrufen.

`/aws/service/global-infrastructure`

> **Note**
>
> Derzeit wird der Pfad `/aws/service/global-infrastructure` nur für Abfragen in den folgenden AWS-Regionen unterstützt:
>
> - USA Ost (Nord-Virginia): (us-east-1)
> - USA Ost (Ohio): (us-east-2)
> - USA West (Nordkalifornien) (us-west-1)
> - USA West (Oregon): (us-west-2)
> - Asien-Pazifik (Hongkong) (ap-east-1)
> - Asien-Pazifik (Mumbai): (ap-south-1)
> - Asien-Pazifik (Seoul): (ap-northeast-2)
> - Asien-Pazifik (Singapur): (ap-southeast-1)
> - Asien-Pazifik (Sydney): (ap-southeast-2)
> - Asien-Pazifik (Tokyo) (ap-northeast-1)
> - Kanada (Zentral): (ca-central-1)
> - Europa (Frankfurt) (eu-central-1)
> - Europa (Irland) (eu-west-1)
> - Europa (London) (eu-west-2)
> - Europa (Paris) (eu-west-3)
> - Europa (Stockholm) (eu-north-1)
> - Südamerika (São Paulo) (sa-east-1)
>
> Wenn Sie in einer anderen Handelsregion arbeiten, können Sie in Ihrer Abfrage eine unterstützte Region angeben, um die Ergebnisse anzuzeigen. Wenn Sie beispielsweise in

der Region Kanada West (Calgary) (ca-west-1) arbeiten, könnten Sie in Ihrer Abfrage Kanada (Zentral) (ca-central-1) angeben:

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/regions \
    --region ca-central-1
```

Anzeigen aktiver AWS-Regionen

Sie können eine Liste aller aktiven AWS-Regionen anzeigen, indem Sie den folgenden Befehl in der AWS Command Line Interface (AWS CLI) verwenden.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/regions \
    --query 'Parameters[].Name'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/regions ^
    --query Parameters[].Name
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
[
    "/aws/service/global-infrastructure/regions/af-south-1",
    "/aws/service/global-infrastructure/regions/ap-east-1",
    "/aws/service/global-infrastructure/regions/ap-northeast-3",
    "/aws/service/global-infrastructure/regions/ap-south-2",
    "/aws/service/global-infrastructure/regions/ca-central-1",
    "/aws/service/global-infrastructure/regions/eu-central-2",
    "/aws/service/global-infrastructure/regions/eu-west-2",
    "/aws/service/global-infrastructure/regions/eu-west-3",
    "/aws/service/global-infrastructure/regions/us-east-1",
    "/aws/service/global-infrastructure/regions/us-gov-west-1",
    "/aws/service/global-infrastructure/regions/ap-northeast-2",
    "/aws/service/global-infrastructure/regions/ap-southeast-1",
```

```
    "/aws/service/global-infrastructure/regions/ap-southeast-2",
    "/aws/service/global-infrastructure/regions/ap-southeast-3",
    "/aws/service/global-infrastructure/regions/cn-north-1",
    "/aws/service/global-infrastructure/regions/cn-northwest-1",
    "/aws/service/global-infrastructure/regions/eu-south-1",
    "/aws/service/global-infrastructure/regions/eu-south-2",
    "/aws/service/global-infrastructure/regions/us-east-2",
    "/aws/service/global-infrastructure/regions/us-west-1",
    "/aws/service/global-infrastructure/regions/ap-northeast-1",
    "/aws/service/global-infrastructure/regions/ap-south-1",
    "/aws/service/global-infrastructure/regions/ap-southeast-4",
    "/aws/service/global-infrastructure/regions/ca-west-1",
    "/aws/service/global-infrastructure/regions/eu-central-1",
    "/aws/service/global-infrastructure/regions/il-central-1",
    "/aws/service/global-infrastructure/regions/me-central-1",
    "/aws/service/global-infrastructure/regions/me-south-1",
    "/aws/service/global-infrastructure/regions/sa-east-1",
    "/aws/service/global-infrastructure/regions/us-gov-east-1",
    "/aws/service/global-infrastructure/regions/eu-north-1",
    "/aws/service/global-infrastructure/regions/eu-west-1",
    "/aws/service/global-infrastructure/regions/us-west-2"
]
```

Anzeigen von verfügbaren AWS-Services

Sie können eine vollständige Liste aller verfügbaren AWS-Services anzeigen und sie in alphabetischer Reihenfolge sortieren, indem Sie den folgenden Befehl ausführen. In diesem Beispiel wurde die Ausgabe aus Platzgründen gekürzt.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/services \
    --query 'Parameters[].Name | sort(@)'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/services ^
    --query "Parameters[].Name | sort(@)"
```

Der Befehl gibt Informationen wie die folgenden zurück. Dieses Beispiel wurde aus Platzgründen gekürzt.

```
[
     "/aws/service/global-infrastructure/services/accessanalyzer",
    "/aws/service/global-infrastructure/services/account",
    "/aws/service/global-infrastructure/services/acm",
    "/aws/service/global-infrastructure/services/acm-pca",
    "/aws/service/global-infrastructure/services/ahl",
    "/aws/service/global-infrastructure/services/aiq",
    "/aws/service/global-infrastructure/services/amazonlocationservice",
    "/aws/service/global-infrastructure/services/amplify",
    "/aws/service/global-infrastructure/services/amplifybackend",
    "/aws/service/global-infrastructure/services/apigateway",
    "/aws/service/global-infrastructure/services/apigatewaymanagementapi",
    "/aws/service/global-infrastructure/services/apigatewayv2",
    "/aws/service/global-infrastructure/services/appconfig",
    "/aws/service/global-infrastructure/services/appconfigdata",
    "/aws/service/global-infrastructure/services/appflow",
    "/aws/service/global-infrastructure/services/appintegrations",
    "/aws/service/global-infrastructure/services/application-autoscaling",
    "/aws/service/global-infrastructure/services/application-insights",
    "/aws/service/global-infrastructure/services/applicationcostprofiler",
    "/aws/service/global-infrastructure/services/appmesh",
    "/aws/service/global-infrastructure/services/apprunner",
    "/aws/service/global-infrastructure/services/appstream",
    "/aws/service/global-infrastructure/services/appsync",
    "/aws/service/global-infrastructure/services/aps",
    "/aws/service/global-infrastructure/services/arc-zonal-shift",
    "/aws/service/global-infrastructure/services/artifact",
    "/aws/service/global-infrastructure/services/athena",
    "/aws/service/global-infrastructure/services/auditmanager",
    "/aws/service/global-infrastructure/services/augmentedairuntime",
    "/aws/service/global-infrastructure/services/aurora",
    "/aws/service/global-infrastructure/services/autoscaling",
    "/aws/service/global-infrastructure/services/aws-appfabric",
    "/aws/service/global-infrastructure/services/awshealthdashboard",
```

Anzeigen unterstützter Regionen für einen AWS-Service

Hier finden Sie eine Liste von AWS-Regionen, in denen ein Service verfügbar ist. Dieses Beispiel verwendet AWS Systems Manager (`ssm`).

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/services/ssm/regions \
    --query 'Parameters[].Value'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/services/ssm/regions ^
    --query Parameters[].Value
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
[
    "ap-south-1",
    "eu-central-1",
    "eu-central-2",
    "eu-west-1",
    "eu-west-2",
    "eu-west-3",
    "il-central-1",
    "me-south-1",
    "us-east-2",
    "us-gov-west-1",
    "af-south-1",
    "ap-northeast-3",
    "ap-southeast-1",
    "ap-southeast-4",
    "ca-central-1",
    "ca-west-1",
    "cn-north-1",
    "eu-north-1",
    "eu-south-2",
    "us-west-1",
    "ap-east-1",
    "ap-northeast-1",
    "ap-northeast-2",
    "ap-southeast-2",
    "ap-southeast-3",
    "cn-northwest-1",
    "eu-south-1",
```

```
    "me-central-1",
    "us-gov-east-1",
    "us-west-2",
    "ap-south-2",
    "sa-east-1",
    "us-east-1"
]
```

Anzeigen des regionalen Endpunkts für einen Service

Sie können einen regionalen Endpunkt für einen Service anzeigen, indem Sie den folgenden Befehl ausführen. Mit diesem Befehl wird die Region USA Ost (Ohio) (us-east-2) abgefragt.

Linux & macOS

```
aws ssm get-parameter \
    --name /aws/service/global-infrastructure/regions/us-east-2/services/ssm/
endpoint \
    --query 'Parameter.Value'
```

Windows

```
aws ssm get-parameter ^
    --name /aws/service/global-infrastructure/regions/us-east-2/services/ssm/
endpoint ^
    --query Parameter.Value
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
"ssm.us-east-2.amazonaws.com"
```

Anzeigen der vollständigen Details zu Availability Zones

Sie können den folgenden Befehl verwenden, um Availability Zones anzuzeigen

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/availability-zones/
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/availability-zones/
```

Der Befehl gibt Informationen wie die folgenden zurück. Dieses Beispiel wurde aus Platzgründen gekürzt.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/global-infrastructure/availability-zones/afs1-az3",
            "Type": "String",
            "Value": "afs1-az3",
            "Version": 1,
            "LastModifiedDate": "2020-04-21T12:05:35.375000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
availability-zones/afs1-az3",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/availability-zones/aps1-az2",
            "Type": "String",
            "Value": "aps1-az2",
            "Version": 1,
            "LastModifiedDate": "2020-04-03T16:13:57.351000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
availability-zones/aps1-az2",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/availability-zones/apse3-az1",
            "Type": "String",
            "Value": "apse3-az1",
            "Version": 1,
            "LastModifiedDate": "2021-12-13T08:51:38.983000-05:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
availability-zones/apse3-az1",
            "DataType": "text"
        }
    ]
```

```
}
```

Anzeigen ausschließlich der Namen von Availability Zones

Sie können den folgenden Befehl verwenden, um ausschließlich die Namen von Availability Zones anzuzeigen.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/availability-zones \
    --query 'Parameters[].Name | sort(@)'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/availability-zones ^
    --query "Parameters[].Name | sort(@)"
```

Der Befehl gibt Informationen wie die folgenden zurück. Dieses Beispiel wurde aus Platzgründen gekürzt.

```
[
    "/aws/service/global-infrastructure/availability-zones/afs1-az1",
    "/aws/service/global-infrastructure/availability-zones/afs1-az2",
    "/aws/service/global-infrastructure/availability-zones/afs1-az3",
    "/aws/service/global-infrastructure/availability-zones/ape1-az1",
    "/aws/service/global-infrastructure/availability-zones/ape1-az2",
    "/aws/service/global-infrastructure/availability-zones/ape1-az3",
    "/aws/service/global-infrastructure/availability-zones/apne1-az1",
    "/aws/service/global-infrastructure/availability-zones/apne1-az2",
    "/aws/service/global-infrastructure/availability-zones/apne1-az3",
    "/aws/service/global-infrastructure/availability-zones/apne1-az4"
```

Anzeigen der Namen von Availability Zones in einer einzelnen Region

Sie können den folgenden Befehl verwenden, um die Namen der Availability Zones in einer Region (in diesem Beispiel `us-east-2`) anzuzeigen.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/regions/us-east-2/availability-zones \
    --query 'Parameters[].Name | sort(@)'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/regions/us-east-2/availability-zones ^
    --query "Parameters[].Name | sort(@)"
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
[
    "/aws/service/global-infrastructure/regions/us-east-2/availability-zones/use2-az1",
    "/aws/service/global-infrastructure/regions/us-east-2/availability-zones/use2-az2",
    "/aws/service/global-infrastructure/regions/us-east-2/availability-zones/use2-az3"
```

Anzeigen ausschließlich der ARNs von Availability Zones

Sie können den folgenden Befehl verwenden, um ausschließlich die Amazon-Ressourcennamen (ARN) von Availability Zones anzuzeigen.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/availability-zones \
    --query 'Parameters[].ARN | sort(@)'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/availability-zones ^
    --query "Parameters[].ARN | sort(@)"
```

Der Befehl gibt Informationen wie die folgenden zurück. Dieses Beispiel wurde aus Platzgründen gekürzt.

```
[
    "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/availability-
zones/afs1-az1",
    "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/availability-
zones/afs1-az2",
    "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/availability-
zones/afs1-az3",
    "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/availability-
zones/ape1-az1",
    "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/availability-
zones/ape1-az2",
    "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/availability-
zones/ape1-az3",
    "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/availability-
zones/apne1-az1",
```

Anzeigen der Details zu lokalen Zonen

Sie können den folgenden Befehl verwenden, um lokale Zonen anzuzeigen.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/local-zones
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/local-zones
```

Der Befehl gibt Informationen wie die folgenden zurück. Dieses Beispiel wurde aus Platzgründen gekürzt.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/global-infrastructure/local-zones/afs1-los1-az1",
            "Type": "String",
            "Value": "afs1-los1-az1",
            "Version": 1,
```

```
            "LastModifiedDate": "2023-01-25T11:53:11.690000-05:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/afs1-los1-az1",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/local-zones/apne1-tpe1-az1",
            "Type": "String",
            "Value": "apne1-tpe1-az1",
            "Version": 1,
            "LastModifiedDate": "2024-03-15T12:35:41.076000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/apne1-tpe1-az1",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/local-zones/aps1-ccu1-az1",
            "Type": "String",
            "Value": "aps1-ccu1-az1",
            "Version": 1,
            "LastModifiedDate": "2022-12-19T11:34:43.351000-05:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/aps1-ccu1-az1",
            "DataType": "text"
        }
    ]
}
```

Anzeigen von Details zu Wavelength Zones

Sie können den folgenden Befehl verwenden, um Wavelength Zones anzuzeigen.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/wavelength-zones
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/wavelength-zones
```

Der Befehl gibt Informationen wie die folgenden zurück. Dieses Beispiel wurde aus Platzgründen gekürzt.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/global-infrastructure/wavelength-zones/apne1-wl1-nrt-
wlz1",
            "Type": "String",
            "Value": "apne1-wl1-nrt-wlz1",
            "Version": 3,
            "LastModifiedDate": "2020-12-15T17:16:04.715000-05:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
wavelength-zones/apne1-wl1-nrt-wlz1",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/wavelength-zones/apne2-wl1-sel-
wlz1",
            "Type": "String",
            "Value": "apne2-wl1-sel-wlz1",
            "Version": 1,
            "LastModifiedDate": "2022-05-25T12:29:13.862000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
wavelength-zones/apne2-wl1-sel-wlz1",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/wavelength-zones/cac1-wl1-yto-
wlz1",
            "Type": "String",
            "Value": "cac1-wl1-yto-wlz1",
            "Version": 1,
            "LastModifiedDate": "2022-04-26T09:57:44.495000-04:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
wavelength-zones/cac1-wl1-yto-wlz1",
            "DataType": "text"
        }
    ]
}
```

Anzeigen aller Parameter und Werte unter einer lokalen Zone

Sie können den folgenden Befehl verwenden, um alle Parameterdaten für eine lokale Zone anzuzeigen.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path "/aws/service/global-infrastructure/local-zones/usw2-lax1-az1/"
```

Windows

```
aws ssm get-parameters-by-path ^
    --path "/aws/service/global-infrastructure/local-zones/use1-bos1-az1"
```

Der Befehl gibt Informationen wie die folgenden zurück. Dieses Beispiel wurde aus Platzgründen gekürzt.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/
geolocationCountry",
            "Type": "String",
            "Value": "US",
            "Version": 3,
            "LastModifiedDate": "2020-12-15T14:16:17.641000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/use1-bos1-az1/geolocationCountry",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/
geolocationRegion",
            "Type": "String",
            "Value": "US-MA",
            "Version": 3,
            "LastModifiedDate": "2020-12-15T14:16:17.794000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/use1-bos1-az1/geolocationRegion",
            "DataType": "text"
        },
        {
```

```
            "Name": "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/
location",
            "Type": "String",
            "Value": "US East (Boston)",
            "Version": 1,
            "LastModifiedDate": "2021-01-11T10:53:24.634000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/use1-bos1-az1/location",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/
network-border-group",
            "Type": "String",
            "Value": "us-east-1-bos-1",
            "Version": 3,
            "LastModifiedDate": "2020-12-15T14:16:20.641000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/use1-bos1-az1/network-border-group",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/
parent-availability-zone",
            "Type": "String",
            "Value": "use1-az4",
            "Version": 3,
            "LastModifiedDate": "2020-12-15T14:16:20.834000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/use1-bos1-az1/parent-availability-zone",
            "DataType": "text"
        },
        {
            "Name": "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/
parent-region",
            "Type": "String",
            "Value": "us-east-1",
            "Version": 3,
            "LastModifiedDate": "2020-12-15T14:16:20.721000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/use1-bos1-az1/parent-region",
            "DataType": "text"
        },
        {
```

```
            "Name": "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/zone-
group",
            "Type": "String",
            "Value": "us-east-1-bos-1",
            "Version": 3,
            "LastModifiedDate": "2020-12-15T14:16:17.983000-08:00",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/global-infrastructure/
local-zones/use1-bos1-az1/zone-group",
            "DataType": "text"
        }
    ]
}
```

Anzeigen ausschließlich der Namen von Parametern für lokale Zonen

Sie können den folgenden Befehl verwenden, um ausschließlich die Namen der Parameter für lokale Zonen anzuzeigen.

Linux & macOS

```
aws ssm get-parameters-by-path \
    --path /aws/service/global-infrastructure/local-zones/usw2-lax1-az1 \
    --query 'Parameters[].Name | sort(@)'
```

Windows

```
aws ssm get-parameters-by-path ^
    --path /aws/service/global-infrastructure/local-zones/use1-bos1-az1 ^
    --query "Parameters[].Name | sort(@)"
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
[
    "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/geolocationCountry",
    "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/geolocationRegion",
    "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/location",
    "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/network-border-
group",
    "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/parent-availability-
zone",
    "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/parent-region",
```

```
    "/aws/service/global-infrastructure/local-zones/use1-bos1-az1/zone-group"
]
```

Walkthroughs für Parameter Store

Der Walkthrough in diesem Abschnitt zeigt, wie Sie Parameter mit Parameter Store, einem Tool in AWS Systems Manager, in einer Testumgebung erstellen, speichern und ausführen können. Diese Anleitungen zeigen Ihnen die Verwendung von Parameter Store mit anderen Systems-Manager-Tools. Sie können Parameter Store auch mit anderen AWS-Services verwenden. Weitere Informationen finden Sie unter Was ist ein Parameter?.

Inhalt

- Einen SecureString Parameter in einer Domain erstellen Parameter Store und einen Knoten mit einer Domain verbinden (PowerShell)

Einen SecureString Parameter in einer Domain erstellen Parameter Store und einen Knoten mit einer Domain verbinden (PowerShell)

Diese exemplarische Vorgehensweise zeigt, wie Sie mithilfe der AWS Systems Manager `SecureString` Parameter und Run Command einen Windows Server Knoten mit einer Domäne verbinden. In der Anleitung werden typische Domain-Parameter verwendet, z. B. der Domain-Name und ein Benutzername für die Domain. Diese Werte werden als unverschlüsselte Zeichenfolgen weitergegeben. Das Passwort für die Domain wird unter Verwendung eines Von AWS verwalteter Schlüssel verschlüsselt und als verschlüsselte Zeichenfolge übergeben.

Voraussetzungen

In dieser Anleitung wird davon ausgegangen, dass Sie Ihren Domain-Namen und die DNS-Server-IP-Adresse in der DHCP-Optionsliste, die Ihrer Amazon VPC zugeordnet ist, bereits angegeben haben. Informationen finden Sie unter Arbeiten mit DHCP-Optionslisten im Amazon VPC-Benutzerhandbuch.

So erstellen Sie einen **`SecureString`**-Parameter und verknüpfen einen Knoten mit einer Domain

1. Geben Sie Parameter in das System ein mit AWS Tools for Windows PowerShell.

   Ersetzen Sie in den folgenden Befehlen jeden Befehl *`user input placeholder`* durch Ihre eigenen Informationen.

   ```
   Write-SSMParameter -Name "domainName" -Value "DOMAIN-NAME" -Type String
   Write-SSMParameter -Name "domainJoinUserName" -Value "DOMAIN\USERNAME" -Type String
   ```

```
Write-SSMParameter -Name "domainJoinPassword" -Value "PASSWORD" -Type SecureString
```

> **Important**
>
> Nur der Wert eines `SecureString`-Parameters wird verschlüsselt. Der Name des Parameters, die Beschreibung und andere Eigenschaften sind nicht verschlüsselt.

2. Fügen Sie den IAM-Rollenberechtigungen für Ihren Knoten die folgenden AWS Identity and Access Management (IAM-) Richtlinien hinzu:

   - Amazon SSMManaged InstanceCore — Erforderlich. Diese AWS verwaltete Richtlinie ermöglicht es einem verwalteten Knoten, die Kernfunktionen des Systems Manager Manager-Dienstes zu verwenden.
   - Amazon SSMDirectory ServiceAccess — Erforderlich. Diese AWS verwaltete Richtlinie ermöglicht den SSM Agent Zugriff in Ihrem Namen AWS Directory Service auf Anfragen zum Beitritt zur Domain durch den verwalteten Knoten.
   - Eine benutzerdefinierte Richtlinie für S3-Bucket-Zugriff – Erforderlich. SSM Agent, der sich auf Ihrem Knoten befindet und Systems-Manager-Aufgaben ausführt, erfordert Zugriff auf bestimmte Amazon Simple Storage Service (Amazon S3)-Buckets im Besitz von Amazon. In der benutzerdefinierten S3-Bucket-Richtlinie, die Sie erstellen, können Sie auch Zugriff auf Ihre eigenen S3-Buckets gewähren, die für Systems Manager-Operationen benötigt werden.

     Beispiele: Sie können Ausgabe für Run Command-Befehle oder Session Manager-Sitzungen in einen S3-Bucket schreiben und diese Ausgabe dann zu einem späteren Zeitpunkt für Auditing-Zwecke oder zur Fehlerbehebung nutzen. Sie speichern Zugriffsskripts oder benutzerdefinierte Patch-Baseline-Listen in einem S3-Bucket und verweisen dann auf das Skript oder die Liste, wenn Sie einen Befehl ausführen oder wenn eine Patch-Baseline angewendet wird.

     Weitere Informationen zum Erstellen einer benutzerdefinierten Richtlinie für den Zugriff auf einen Amazon S3-Bucket finden Sie unter Erstellen einer benutzerdefinierten S3-Bucket-Richtlinie für ein Instance-Profil

     > **Note**
     >
     > Die Speicherung von Ausgabeprotokolldaten in einem S3-Bucket ist optional. Wenn Sie sich jedoch hierzu entschlossen haben, sollte die Funktion zu Beginn des Systems Manager-Konfigurationsprozesses eingerichtet werden. Weitere Informationen finden

Sie unter Erstellen eines Buckets im Benutzerhandbuch zu Amazon Simple Storage Service.

- CloudWatchAgentServerPolicy – Optional. Diese AWS verwaltete Richtlinie ermöglicht es Ihnen, den CloudWatch Agenten auf verwalteten Knoten auszuführen. Diese Richtlinie ermöglicht es, Informationen auf einem Knoten zu lesen und an Amazon zu schreiben CloudWatch. Ihr Instance-Profil benötigt diese Richtlinie nur, wenn Sie Dienste wie Amazon EventBridge oder CloudWatch Logs verwenden.

> **Note**
>
> Die Verwendung CloudWatch der EventBridge Funktionen ist optional, wir empfehlen jedoch, sie zu Beginn des Systems Manager Manager-Konfigurationsprozesses einzurichten, wenn Sie sich für deren Verwendung entschieden haben. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch und im Amazon CloudWatch Logs-Benutzerhandbuch.

3. Bearbeiten Sie die IAM-Rolle, die dem Knoten zugeordnet ist, und fügen Sie die folgende Richtlinie hinzu. Diese Richtlinie erteilt dem Knoten Berechtigungen, um die `kms:Decrypt`- und `ssm:CreateDocument`-API aufrufen zu können.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "ssm:CreateDocument"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/kms-key-id"
            ]
        }
    ]
}
```

4. Kopieren Sie den folgenden JSON-Text in einen Texteditor und speichern Sie die Datei unter dem Namen `JoinInstanceToDomain.json` am folgenden Speicherort: `c:\temp\JoinInstanceToDomain.json`.

```
{
    "schemaVersion": "2.2",
    "description": "Run a PowerShell script to securely join a Windows Server
 instance to a domain",
    "mainSteps": [
        {
            "action": "aws:runPowerShellScript",
            "name": "runPowerShellWithSecureString",
            "precondition": {
                "StringEquals": [
                    "platformType",
                    "Windows"
                ]
            },
            "inputs": {
                "runCommand": [
                    "$domain = (Get-SSMParameterValue -Name
 domainName).Parameters[0].Value",
                    "if ((gwmi Win32_ComputerSystem).domain -eq $domain){write-host
 \"Computer is part of $domain, exiting\"; exit 0}",
                    "$username = (Get-SSMParameterValue -Name
 domainJoinUserName).Parameters[0].Value",
                    "$password = (Get-SSMParameterValue -Name domainJoinPassword -
WithDecryption $True).Parameters[0].Value | ConvertTo-SecureString -asPlainText -
Force",
                    "$credential = New-Object
 System.Management.Automation.PSCredential($username,$password)",
                    "Add-Computer -DomainName $domain -Credential $credential -
ErrorAction SilentlyContinue -ErrorVariable domainjoinerror",
                    "if($?){Write-Host \"Instance joined to domain successfully.
 Restarting\"; exit 3010}else{Write-Host \"Instance failed to join domain with
 error:\" $domainjoinerror; exit 1 }"
                ]
            }
        }
    ]
}
```

5. Führen Sie den folgenden Befehl in Tools für Windows aus PowerShell , um ein neues SSM-Dokument zu erstellen.

```
$json = Get-Content C:\temp\JoinInstanceToDomain | Out-String
New-SSMDocument -Name JoinInstanceToDomain -Content $json -DocumentType Command
```

6. Führen Sie den folgenden Befehl in Tools für Windows aus PowerShell , um den Knoten mit der Domäne zu verbinden.

```
Send-SSMCommand -InstanceId instance-id -DocumentName JoinInstanceToDomain
```

Wenn der Befehl erfolgreich ausgeführt wurde, sieht das Ergebnis im System in etwa wie folgt aus:

```
WARNING: The changes will take effect after you restart the computer EC2ABCD-
EXAMPLE.
Domain join succeeded, restarting
Computer is part of example.local, exiting
```

Wenn der Befehl nicht erfolgreich ausgeführt wurde, sieht das Ergebnis im System in etwa wie folgt aus:

```
Failed to join domain with error:
Computer 'EC2ABCD-EXAMPLE' failed to join domain 'example.local'
from its current workgroup 'WORKGROUP' with following error message:
The specified domain either does not exist or could not be contacted.
```

Prüfen und Protokollieren von Parameter Store-Aktivitäten

AWS CloudTrail erfasst über die AWS Systems Manager-Konsole, die AWS Command Line Interface (AWS CLI) und das Systems Manager SDK ausgeführte API-Aufrufe. Sie können die Informationen in der CloudTrail-Konsole oder in einem Amazon Simple Storage Service (Amazon S3)-Bucket anzeigen. Alle CloudTrail-Protokolle in Ihrem Konto verwenden nur ein Bucket. Weitere Informationen zum Anzeigen und Verwenden von CloudTrail-Protokollen von Systems Manager-Aktivitäten finden Sie unter Protokollierung von AWS Systems Manager-API-Aufrufen mit AWS CloudTrail. Weitere Informationen zu den Prüfungs- und Protokollierungsoptionen für Systems Manager finden Sie unter Einloggen und Überwachen AWS Systems Manager.

Fehlerbehebung für Parameter Store

Im Folgenden finden Sie Informationen zur Behandlung von Problemen mit Parameter Store, einem Tool in AWS Systems Manager.

Fehlerbehebung bei der Erstellung von **aws:ec2:image**-Parametern

Verwenden Sie die folgenden Informationen, um Probleme bei der Erstellung von `aws:ec2:image`-Datentypparametern zu beheben.

Keine Berechtigung zum Erstellen einer Instance

Problem: Sie versuchen, eine Instance mithilfe eines `aws:ec2:image` Parameters zu erstellen, erhalten jedoch eine Fehlermeldung wie „Sie sind nicht berechtigt, diesen Vorgang auszuführen“.

- Lösung: Sie verfügen nicht über alle erforderlichen Berechtigungen zum Erstellen einer EC2-Instance mit einem Parameterwert, beispielsweise die Berechtigungen für `ec2:RunInstances`, `ec2:DescribeImages` und `ssm:GetParameter`, Wenden Sie sich an einen Benutzer mit Administratorberechtigungen in Ihrer Organisation, um die erforderlichen Berechtigungen anzufordern.

EventBridge meldet den Fehler „Unable to Describe Resource“ (Ressource kann nicht beschrieben werden)

Problem: Sie haben einen Befehl ausgeführt, um einen `aws:ec2:image`-Parameter zu erstellen, die Parametererstellung ist jedoch fehlgeschlagen. Sie erhalten von Amazon EventBridge die Benachrichtigung über die Ausnahme „Unable to Describe Resource“ (Ressource kann nicht beschrieben werden).

Lösung: Diese Meldung kann auf Folgendes hinweisen:

- Sie verfügen nicht über alle für die `ec2:DescribeImages`-API-Operation erforderlichen Berechtigungen oder Sie haben keine Berechtigung für den Zugriff auf das spezifische Image, auf das im Parameter verwiesen wird. Wenden Sie sich an einen Benutzer mit Administratorberechtigungen in Ihrer Organisation, um die erforderlichen Berechtigungen anzufordern.
- Die Amazon Machine Image (AMI)-ID, die Sie als Parameterwert eingegeben haben, ist ungültig. Stellen Sie sicher, dass Sie die ID eines AMI eingeben, das in der AWS-Region und in dem Konto verfügbar ist, in der bzw. dem Sie aktuell arbeiten.

Es ist kein neuer **`aws:ec2:image`**-Parameter verfügbar

Problem: Sie haben gerade einen Befehl ausgeführt, um einen `aws:ec2:image`-Parameter zu erstellen, und eine Versionsnummer wurde gemeldet, der Parameter ist jedoch nicht verfügbar.

- Lösung: Wenn Sie den Befehl zum Erstellen eines Parameters ausführen, der den Datentyp `aws:ec2:image` verwendet, wird sofort eine Versionsnummer für den Parameter generiert. Das Parameterformat muss jedoch validiert werden, bevor der Parameter verfügbar ist. Dieser Vorgang kann einige Minuten dauern. Wenn Sie die Parametererstellung- und -validierung überwachen möchten, können Sie Folgendes tun:
  - Lassen Sie sich über EventBridge Benachrichtigungen über die `create`- und `update`-Parameter-Operationen senden. Diese Benachrichtigungen melden, ob eine Parameteroperation erfolgreich war oder nicht. Weitere Informationen zum Abonnieren von Parameter Store-Ereignissen in EventBridge finden Sie unter Einrichten von Benachrichtigungen oder Auslösen von Aktionen basierend auf Parameter Store-Ereignissen.
  - Aktualisieren Sie im Abschnitt Parameter Store der Systems Manager-Konsole regelmäßig die Parameterliste, um nach den neuen oder aktualisierten Parameterdetails zu suchen.
  - Verwenden Sie den Befehl GetParameter, um nach dem neuen oder aktualisierten Parameter zu suchen. Beispielsweise mithilfe der AWS Command Line Interface (AWS CLI):

    ```
    aws ssm get-parameter name MyParameter
    ```

    Für einen neuen Parameter wird die Meldung `ParameterNotFound` zurückgegeben, bis der Parameter validiert wurde. Für einen vorhandenen Parameter, den Sie aktualisieren, werden Informationen zur neuen Version erst erfasst, wenn der Parameter validiert wurde.

  Wenn Sie versuchen, den Parameter erneut zu erstellen oder zu aktualisieren, bevor der Validierungsprozess abgeschlossen ist, meldet das System, dass die Validierung noch läuft. Wenn der Parameter nicht erstellt oder aktualisiert wurde, können Sie es fünf Minuten nach dem ersten Versuch erneut versuchen.

AWS Systems Manager-Tools für den Betrieb

Operations-Tools besteht aus einer Reihe von Funktionen, die Sie bei der Verwaltung Ihrer AWS-Ressourcen unterstützen.

- Von Systems Manager gehostete Amazon-CloudWatch-Dashboards sind anpassbare Homepages in der CloudWatch-Konsole, mit denen Sie Ihre Ressourcen über AWS-Regionen in einer einzigen Ansicht überwachen können.
- Explorer ist ein anpassbares Betriebs-Dashboard, in dem Informationen zu Ihren AWS-Ressourcen aufgeführt werden. Explorer zeigt eine Gesamtansicht der Betriebsdaten (OpsData) für Ihre AWS-Konten und für alle AWS-Regionen an.
- Incident Manager hilft Ihnen, Ihre von AWS gehosteten Anwendungen entlasten und wiederherstellen zu können.
- OpsCenter bietet einen zentralen Speicherort, an dem Techniker und IT-Experten operative Arbeitselemente anzeigen, untersuchen und lösen können (OpsItems), die im Zusammenhang mit AWS-Ressourcen stehen.

Themen

- AWS Systems Manager Incident Manager
- AWS Systems Manager Explorer
- AWS Systems Manager OpsCenter
- Amazon-CloudWatch-Dashboards, die von Systems Manager gehostet werden

AWS Systems Manager Incident Manager

> **Änderung der Verfügbarkeit von Incident Manager**
>
> AWS Systems Manager Incident Manager steht ab dem 7. November 2025 nicht mehr für Neukunden zur Verfügung. Wenn Sie Incident Manager nutzen möchten, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems Manager Incident Manager Verfügbarkeit.

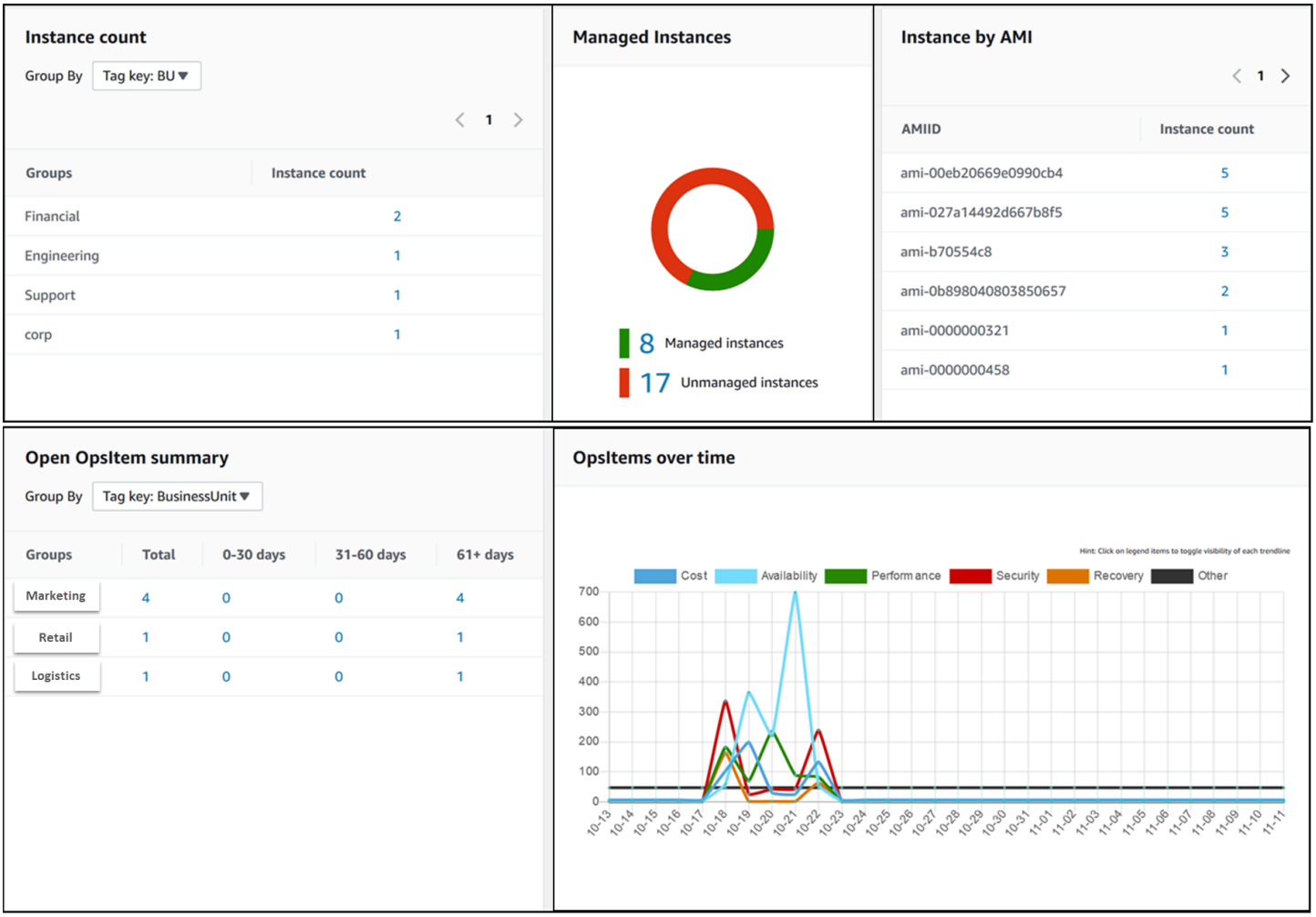
Verwenden Sie Incident Manager, ein Tool in AWS Systems Manager, um Vorfälle zu verwalten, die in Ihren AWS gehosteten Anwendungen auftreten. Incident Manager kombiniert Benutzereingriffe, Eskalation, Runbooks, Reaktionspläne, Chat-Kanäle und Analysen nach Vorfällen, damit Ihr Team Vorfälle schneller einordnen und Ihre Anwendungen wieder in den Normalzustand versetzen kann. Weitere Informationen zu Incident Manager finden Sie unter Incident Manager-Benutzerhandbuch.

AWS Systems Manager Explorer

AWS Systems Manager Explorer ist ein anpassbares Operations-Dashboard, das Informationen über Ihre AWS Ressourcen enthält. Explorerzeigt eine aggregierte Ansicht der Betriebsdaten (OpsData) für Sie AWS-Konten und Across AWS-Regionen an. OpsData Enthält Metadaten zu den verwalteten Knoten in Ihrer Hybrid- und Multi-Cloud-Umgebung. Explorer OpsData umfasst auch Informationen, die von anderen Systems Manager Manager-Tools bereitgestellt werden, einschließlich Informationen zur Patch Manager Patch-Compliance und State Manager zur Einhaltung von Zuordnungen. Um den Zugriff weiter zu vereinfachen OpsData, Explorer werden Informationen von unterstützenden AWS Diensten wie AWS Config, AWS Trusted Advisor AWS Compute Optimizer, und AWS -Support (Supportfälle) angezeigt.

Um die betriebliche Sensibilisierung zu erhöhen zeigt Explorer auch operative Arbeitselemente an (OpsItems). Explorer bietet Kontext darüber, wie OpsItems über Ihre Geschäftsbereiche oder Anwendungen verteilt sind, welche Trends im Zeitverlauf bestehen und wie sie je nach Kategorie variieren. Sie können Informationen in Explorer gruppieren und filtern, um sich auf die Elemente zu konzentrieren, die für Sie relevant sind und eine Aktion erfordern. Wenn Sie Probleme mit hoher Priorität erkennen, können Sie mit Systems Manager OpsCenter Automation-Runbooks ausführen und die Probleme schnell beheben. Um mit Explorer zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich Explorer aus.

Die folgende Abbildung zeigt einige der einzelnen Berichtsfelder, Widgets, genannt, die in Explorer verfügbar sind.



Über welche Features verfügt Explorer?

Explorer umfasst die folgenden Features:

- Anpassbare Anzeige verwertbarer Informationen: Explorer beinhaltet drag-and-drop Widgets, die automatisch umsetzbare Informationen zu Ihren AWS Ressourcen anzeigen. Explorerzeigt Informationen in zwei Arten von Widgets an.
  - Informations-Widgets: Diese Widgets fassen Daten von Amazon EC2, Patch ManagerState Manager, und unterstützenden Programmen AWS-Services wie AWS Trusted Advisor AWS Compute Optimizer und zusammen. Support Diese Widgets bieten wichtigen Kontext, der Ihnen hilft, den Zustand und die betrieblichen Risiken Ihrer Ressourcen zu verstehen. AWS Beispiele für Informations-Widgets sind: Instance count (Instance-Anzahl), Instance by AMI (Instance nach AMI), Total noncompliant nodes (Gesamtzahl nicht konformer Knoten) (Patching), Noncompliant associations (Nicht konforme Zuordnungen) und Support Center cases (Support-Center-Fälle).

  - OpsItemWidgets: Ein Systems Manager OpsItemist ein operatives Arbeitselement, das sich auf eine oder mehrere AWS Ressourcen bezieht. OpsItemssind ein Feature von Systems ManagerOpsCenter. OpsItemserfordert möglicherweise DevOps Techniker, um ein Problem zu untersuchen und möglicherweise zu beheben. Mögliche OpsItems Beispiele hierfür sind eine hohe EC2 Instance-CPU-Auslastung, getrennte Amazon Elastic Block Store (Amazon EBS) - Volumes, ein AWS CodeDeploy Bereitstellungsfehler oder ein Fehler bei der Ausführung von Systems Manager Automation. Zu den Beispielen für OpsItem-Widgets gehören Open OpsItem summary (OpsItem-Zusammenfassung öffnen), OpsItem by status (nach Status) und OpsItems over time (im Laufe der Zeit).
- Filter: Jedes Widget bietet die Möglichkeit, Informationen anhand von AWS-Konto AWS-Region, und Tags zu filtern. Mithilfe von Filtern können Sie die in Explorer angezeigten Informationen schnell verfeinern.
- Direkte Links zu Servicebildschirmen: Um Ihnen bei der Untersuchung von Problemen mit AWS Ressourcen zu helfen, enthalten Explorer Widgets direkte Links zu verwandten Servicebildschirmen. Filter, die auf ein Widget angewendet werden, bleiben wirksam, wenn Sie zu einem zugehörigen Service-Bildschirm navigieren.
- Gruppen: Mit einigen Widgets können Sie Daten basierend auf Konto, Region und Tag gruppieren, um die Arten von betrieblichen Problemen in Ihrer Organisation zu verstehen.
- Berichts-Tag-Schlüssel: Wenn Sie Explorer einrichten, können Sie bis zu fünf Tag-Schlüssel angeben. Diese Schlüssel helfen Ihnen, Daten in Explorer zu gruppieren und zu filtern. Wenn ein angegebener Schlüssel mit einem Schlüssel auf einer Ressource übereinstimmt, die ein OpsItem generiert, sind der Schlüssel und der Wert in OpsItems enthalten.
- Drei Modi von AWS-Konto und AWS-Region Anzeige: Explorer umfasst die folgenden Anzeigemodi für OpsData und OpsItems in AWS-Konten und AWS-Regionen:
  - Einzelkonto/Einzelregion: Dies ist die Standardansicht. In diesem Modus können Benutzer Daten und OpsItems von ihrem eigenen Konto und der aktuellen Region anzeigen.
  - Einzelkonto/Mehrfachregion: In diesem Modus müssen Sie mithilfe der Seite Explorer-Settings (Einstellungen) mindestens eine Ressourcen-Datensynchronisierung erstellen. Bei einer Ressourcendatensynchronisierung werden Aggregate OpsData aus einer oder mehreren Regionen gesammelt. Nachdem Sie eine Ressourcen-Datensynchronisierung erstellt haben, können Sie umschalten, welche Synchronisierung im Explorer-Dashboard verwendet werden soll. Anschließend können Sie Daten basierend auf der Region filtern und gruppieren.
  - Mehrfachkonto/Mehrfachregion: Dieser Modus erfordert, dass Ihre Organisation oder Ihr Unternehmen AWS Organizations mit Alle Features eingeschaltet verwendet. Nach der Konfiguration AWS Organizations in Ihrer Computerumgebung können Sie alle Kontodaten

in einem Verwaltungskonto zusammenfassen. Anschließend können Sie die Ressourcen-Datensynchronisierungen erstellen, damit Sie die Daten basierend auf der Region filtern und gruppieren können. Weitere Informationen über den Modus Alle Features in Organisationen finden Sie unter Aktivieren aller Features in Ihrer Organisation.

- Berichterstellung: Sie können Explorer-Berichte als CSV-Dateien in einen Amazon Simple Storage Service (Amazon S3)-Bucket exportieren. Wenn ein Export abgeschlossen ist, erhalten Sie eine Benachrichtigung von Amazon Simple Notification Service (Amazon SNS).

In welcher Verbindung steht Explorer mit OpsCenter?

Systems Manager OpsCenter bietet einen zentralen Ort, an dem Betriebsingenieure und IT-Experten AWS Ressourcen einsehen, untersuchen und OpsItems Probleme lösen können. Explorerist ein Berichtszentrum, in dem DevOps Manager zusammengefasste Zusammenfassungen ihrer Betriebsdaten einsehen könnenOpsItems, einschließlich Across AWS-Regionen - und Accounts. Explorerhilft Benutzern, Trends und Muster zu erkennen und, falls erforderlich, Probleme mithilfe von Systems Manager Automation-Runbooks schnell zu lösen.

Das OpsCenter-Setup ist jetzt in das Explorer-Setup integriert. Wenn Sie OpsCenter bereits eingerichtet haben,zeigt Explorer automatisch Betriebsdaten an, einschließlich aggregierter Informationen über OpsItems. Wenn Sie OpsCenter nicht eingerichtet haben, können Sie das Explorer-Setup verwenden, um mit beiden Tools zu beginnen. Weitere Informationen finden Sie unter Erste Schritte mit Systems Manager Explorer und OpsCenter.

Was ist OpsData?

OpsData sind alle Betriebsdaten, die im Systems Manager Explorer-Dashboard angezeigt werden. Explorerruft OpsData aus den folgenden Quellen ab:

- Amazon Elastic Compute Cloud (Amazon EC2)

  Die in Explorer angezeigten Daten umfassen die Gesamtanzahl der Knoten, die Gesamtanzahl der verwalteten und nicht verwalteten Knoten und die Anzahl von Knoten, die ein spezifisches Amazon Machine Image (AMI) verwenden.
- Systems Manager OpsCenter

  Die in Explorer angezeigten Daten umfassen die Anzahl von OpsItems nach Status, die Anzahl von OpsItems nach Schweregrad, die Anzahl der offenen OpsItems in allen Gruppen und für Zeiträume von 30 Tagen sowie historische Daten zu OpsItems im Zeitverlauf.

- Systems Manager Patch Manager

  Die in Explorer angezeigten Daten enthalten die Anzahl der nicht konformen und kritischen nicht konformen Knoten.

- AWS Trusted Advisor

  Zu den angezeigten Daten Explorer gehören: Status der Best-Practice-Prüfungen für EC2 Reserved Instances in den Bereichen Kostenoptimierung, Sicherheit, Fehlertoleranz, Leistung und Servicebeschränkungen.

- AWS Compute Optimizer

  Zu den angezeigten Daten Explorer gehören: die Anzahl der zu wenig bereitgestellten und zu viel bereitgestellten EC2 Instances, Optimierungsergebnisse, On-Demand-Preisdetails und Empfehlungen zu Instance-Typ und Preis.

- Support Fälle im Mittelpunkt

  Angezeigte Daten in Explorer umfassen: Fall-ID, Schweregrad, Status, Erstellungszeit, Betreff, Service und Kategorie.

- AWS Config

  Zu den Daten, die unter angezeigt werden, Explorer gehören: eine allgemeine Zusammenfassung der konformen und nicht konformen AWS Config Regeln, die Anzahl der konformen und nicht konformen Ressourcen sowie spezifische Details zu den einzelnen Regeln (wenn Sie eine nicht konforme Regel oder Ressource genauer untersuchen).

- AWS Security Hub CSPM

  Zu den angezeigten Daten Explorer gehören: eine allgemeine Zusammenfassung der CSPM-Ergebnisse von Security Hub, die Anzahl der einzelnen Ergebnisse, gruppiert nach Schweregrad, und spezifische Details zu den Ergebnissen.

> **Note**
>
> Um Fälle anzeigen AWS Trusted Advisor und Support bearbeiten zu könnenExplorer, müssen Sie entweder ein Enterprise- oder ein Business-Konto mit eingerichtet haben. AWS - Support

Sie können OpsData Quellen auf der Seite Explorer „Einstellungen“ anzeigen und verwalten. Informationen zum Einrichten und Konfigurieren von Diensten, mit denen Explorer Widgets gefüllt werden OpsData, finden Sie unterEinrichten von zugehörigen Services für Explorer.

Entstehen Kosten für die Verwendung von Explorer?

Ja. Wenn Sie die Standardregeln für die Erstellung von OpsItems während des integrierten Setups aktivieren, starten Sie einen Prozess, der OpsItems automatisch erstellt. Ihr Konto wird basierend auf der Anzahl der pro Monat erstellten OpsItems belastet. Ihr Konto wird außerdem basierend auf der Anzahl der pro Monat getätigten `GetOpsItem`-, `DescribeOpsItem`-, `UpdateOpsItem`- und `GetOpsSummary`-API-Aufrufe belastet. Darüber hinaus können öffentliche API-Aufrufe an andere Services, die relevante Diagnoseinformationen bereitstellen, in Rechnung gestellt werden. Weitere Informationen finden Sie unter AWS Systems Manager - Preise.

Themen

- Erste Schritte mit Systems Manager Explorer und OpsCenter
- Verwenden von Explorer
- OpsData Aus Systems Manager exportieren Explorer
- Fehlerbehebung von Systems Manager Explorer

Erste Schritte mit Systems Manager Explorer und OpsCenter

AWS Systems Manager verwendet eine integrierte Einrichtungsumgebung, um Ihnen den Einstieg in Systems Manager Explorer und Systems Manager OpsCenter zu erleichtern. In dieser Dokumentation wird das Setup von Explorer und OpsCenter als Integriertes Setup bezeichnet. Wenn Sie OpsCenter bereits eingerichtet haben, müssen Sie das integrierte Setup trotzdem ausführen, um die Einstellungen und Optionen zu überprüfen. Wenn Sie OpsCenter nicht eingerichtet haben, können Sie mit dem integrierten Setup beide Tools einrichten.

> **Note**
>
> Integriertes Setup ist nur in der Systems Manager-Konsole verfügbar. Sie können Explorer oder OpsCenter nicht programmatisch einrichten.

Das integrierte Setup führt die folgenden Aufgaben aus:

- Konfiguriert Rollen und Berechtigungen: Das integrierte Setup erstellt eine AWS Identity and Access Management (IAM)-Rolle, die es Amazon EventBridge ermöglicht, OpsItems automatisch basierend auf Standardregeln zu erstellen. Nach dem Einrichten müssen Sie Benutzer-, Gruppen- oder Rollenberechtigungen für OpsCenter konfigurieren, wie in diesem Abschnitt beschrieben.
- Aktiviert Standardregeln für die OpsItem-Erstellung: Das integrierte Setup erstellt Standardregeln in EventBridge. Diese Regeln erstellen OpsItems automatisch als Reaktion auf Ereignisse. Beispiele für diese Ereignisse sind: Statusänderung für eine AWS-Ressource, Änderung der Sicherheitseinstellungen oder ein Service, der nicht mehr verfügbar ist.
- Aktiviert OpsData-Quellen: Das integrierte Setup aktiviert die folgenden Datenquellen, die Explorer-Widgets ausfüllen.
  - Support Center (Sie müssen entweder über einen Business- oder Enterprise-Support-Plan verfügen, um diese Quelle zu aktivieren.)
  - AWS Compute Optimizer (Sie müssen entweder über einen Business- oder Enterprise-Support-Plan verfügen, um diese Quelle zu aktivieren.)
  - Systems Manager State Manager Association Compliance
  - AWS Config-Compliance
  - Systems Manager OpsCenter
  - Systems Manager Patch Manager Patch Compliance
  - Amazon Elastic Compute Cloud (Amazon EC2)
  - Systems Manager Inventory
  - AWS Trusted Advisor (Sie müssen entweder über einen Business- oder Enterprise-Support-Plan verfügen, um diese Quelle zu aktivieren.)
  - AWS Security Hub CSPM

> **Note**
>
> Sie können die Setup-Konfigurationen jederzeit auf der Seite Settings (Einstellungen) ändern.

Nach dem Abschluss des integrierten Setups empfehlen wir, dass Sie Set up Explorer to display data from multiple Regions and accounts (einrichten, um Daten aus mehreren Regionen und Konten anzuzeigen). Explorer und OpsCenter synchronisieren OpsData und OpsItems automatisch für das AWS-Konto und die AWS-Region, die Sie verwendet haben, als Sie das integrierte Setup

abgeschlossen haben. Sie können OpsData und OpsItems aus anderen Konten und Regionen aggregieren, indem Sie eine Ressourcendatensynchronisierung erstellen.

Einrichten von zugehörigen Services für Explorer

AWS Systems Manager Erkunden und AWS Systems Manager OpsCenter sammeln Sie Informationen aus anderen Tools AWS-Services und Systems Manager Manager-Tools oder interagieren Sie mit diesen. Wir empfehlen, diese anderen Services oder Tools einzurichten und zu konfigurieren, bevor Sie das integrierte Setup verwenden.

Die folgende Tabelle enthält Aufgaben, die es ermöglichen, Informationen aus anderen Tools Explorer AWS-Services und Systems Manager Manager-Tools OpsCenter zu sammeln oder mit diesen zu interagieren.

Aufgabe	Informationen
Überprüfen der Berechtigungen in Systems Manager Automation	Explorer und OpsCenter ermöglichen es Ihnen, Probleme mit AWS -Ressourcen mithilfe von Systems Manager Automation-Runbooks zu beheben. Zur Verwendung dieses Tools benötigen Sie die Berechtigung zum Ausführen von Systems-Manager-Automation-Runbooks. Weitere Informationen finden Sie unter Einrichten der Automatisierung.
Einrichten und Konfigurieren von Systems Manager Patch Manager	Explorer enthält ein Widget, das Informationen zu Patch Compliance bereitstellt. Um diese Daten in Explorer anzuzeigen, müssen Sie das Patching konfigurieren. Weitere Informationen finden Sie unter AWS Systems Manager Patch Manager.
Einrichten und Konfigurieren von Systems Manager State Manager	Explorer enthält ein Widget, das Informationen zu Systems Manager State Manager-Association-Compliance bereitstellt. Um diese Daten in Explorer anzuzeigen, müssen Sie State Manager konfigurieren. Weitere Informationen

Aufgabe	Informationen
	finden Sie unter AWS Systems Manager State Manager.
Schalten Sie den AWS Config Configuration Recorder ein	Explorerverwendet die vom AWS Config Configuration Recorder bereitgestellten Daten, um Widgets mit Informationen über Ihre EC2 Instanzen zu füllen. Um diese Daten anzuzeige nExplorer, schalten Sie den AWS Config Konfigurationsrekorder ein. Weitere Informati onen finden Sie unter Verwalten von Configura tion Recorder. **Note** Nachdem Sie den Konfigurationsreko rder zugelassen haben, kann es bis zu sechs Stunden dauern, bis Systems Manager Daten in Explorer Widgets anzeigt, die Informationen zu Ihren EC2 Instanzen anzeigen.
Einschalten AWS Trusted Advisor	Explorerverwendet die von bereitgestellten Daten Trusted Advisor , um den Status von Best-Practice-Prüfungen für Amazon EC2 Reserved Instances in den Bereichen Kostenoptimierung, Sicherheit, Fehlertol eranz, Leistung und Servicebeschränkun gen anzuzeigen. Um diese Daten in Explorer anzuzeigen, benötigen Sie einen Business- oder Enterprise-Supportplan. Weitere Informati onen finden Sie unter Support.

Aufgabe	Informationen
Einschalten AWS Compute Optimizer	Explorerverwendet von Compute Optimizer bereitgestellte Daten, um Details wie die Anzahl der zu wenig bereitgestellten und übermäßig bereitgestellten EC2 Instanzen, Optimieru ngsergebnisse, On-Demand-Preisdetails und Empfehlungen zu Instanztyp und Preis anzuzeigen. Aktivieren Sie Compute Optimizer zum Anzeigen von Daten in Explorer. Weitere Informationen finden Sie unter Erste Schritte mit. AWS Compute Optimizer
Einschalten AWS Security Hub CSPM	Explorerverwendet von Security Hub CSPM bereitgestellte Daten, um Widgets mit Informati onen zu Ihren Sicherheitsergebnissen zu füllen. Um diese Daten anzuzeigenExplorer, aktiviere n Sie die Security Hub CSPM-Integration. Weitere Informationen finden Sie unter Was ist. AWS Security Hub CSPM

Konfigurieren von Rollen und Berechtigungen für Systems Manager Explorer

Integrated Setup erstellt und konfiguriert automatisch AWS Identity and Access Management (IAM) -Rollen für AWS Systems Manager Explorer und. AWS Systems Manager OpsCenter Wenn Sie das integrierte Setup abgeschlossen haben, müssen Sie keine zusätzlichen Aufgaben ausführen, um Rollen und Berechtigungen für Explorer zu konfigurieren. Sie müssen jedoch die Berechtigung für OpsCenter konfigurieren, wie weiter unten in diesem Thema beschrieben.

Das integrierte Setup erstellt und konfiguriert die folgenden Rollen für die Arbeit mit Explorer und OpsCenter.

- `AWSServiceRoleForAmazonSSM`: Bietet Zugriff auf AWS -Ressourcen, die von Systems Manager verwaltet oder verwendet werden.
- `OpsItem-CWE-Role`: Ermöglicht CloudWatch Ereignisse und das Erstellen EventBridge von Inhalten als Reaktion OpsItems auf häufig auftretende Ereignisse.

- `AWSServiceRoleForAmazonSSM_AccountDiscovery`: Ermöglicht Systems Manager, beim Synchronisieren von Daten andere aufzurufen, AWS-Services um AWS-Konto Informationen zu ermitteln. Weitere Informationen über diese Rolle finden Sie unter Verwenden von Rollen zum Sammeln von AWS-Konto-Informationen für OpsCenter und Explorer.
- `AmazonSSMExplorerExport`: Ermöglicht Explorer den Export in eine OpsData Datei mit kommagetrennten Werten (CSV).

Wenn Sie so konfigurierenExplorer, dass Daten aus mehreren Konten und Regionen mithilfe AWS Organizations einer Ressourcendatensynchronisierung angezeigt werden, erstellt Systems Manager die `AWSServiceRoleForAmazonSSM_AccountDiscovery` dienstverknüpfte Rolle. Systems Manager verwendet diese Rolle, um Informationen über Ihr Konto AWS-Konten abzurufen AWS Organizations. Die Rolle verwendet die folgende Berechtigungsrichtlinie.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "organizations:DescribeAccount",
            "organizations:DescribeOrganization",
            "organizations:ListAccounts",
            "organizations:ListAWSServiceAccessForOrganization",
            "organizations:ListChildren",
            "organizations:ListParents"
         ],
         "Resource":"*"
      }
   ]
}
```

Weitere Informationen zur `AWSServiceRoleForAmazonSSM_AccountDiscovery`-Rolle finden Sie unter Verwenden von Rollen zum Sammeln von AWS-Konto-Informationen für OpsCenter und Explorer.

Konfigurieren von Berechtigungen für Systems Manager OpsCenter

Nachdem Sie die integrierte Einrichtung abgeschlossen haben, müssen Sie Benutzer-, Gruppen- oder Rollenberechtigungen konfigurieren, damit Benutzer Aktionen in OpsCenter ausführen können.

Bevor Sie beginnen

Sie können OpsCenter so konfigurieren, dass OpsItems für ein einzelnes Konto oder über mehrere Konten hinweg erstellt und verwaltet wird. Wenn Sie so konfigurieren, OpsCenter dass mehrere Konten erstellt OpsItems und verwaltet werden, können Sie entweder das delegierte Systems Manager Manager-Administratorkonto oder das AWS Organizations Verwaltungskonto verwenden, um andere Konten manuell zu erstellen, anzuzeigen oder zu bearbeitenOpsItems. Weitere Informationen zu dem delegierten Administratorkonto von Systems Manager finden Sie unter Konfigurierung eines delegierten Administrators für Explorer.

Wenn Sie OpsCenter für ein einzelnes Konto konfigurieren, können Sie OpsItems nur in dem Konto anzeigen oder bearbeiten, in dem OpsItems erstellt wurden. Sie können es nicht teilen oder übertragenOpsItems. AWS-Konten Aus diesem Grund empfehlen wir Ihnen, Berechtigungen für OpsCenter das zu konfigurieren AWS-Konto , das zur Ausführung Ihrer AWS Workloads verwendet wird. Anschließend können Sie in diesem Konto -Benutzer oder -Gruppen erstellen. Auf diese Weise können mehrere Betriebstechniker oder IT-Experten OpsItems in demselben AWS-Konto erstellen, einsehen und bearbeiten.

Explorer und OpsCenter verwenden die folgenden API-Operationen. Sie können alle Funktionen von Explorer und OpsCenter verwenden, wenn Ihre Benutzer, Gruppe oder Rolle über Zugriff für diese Aktionen verfügen. Sie können auch strengere Zugriffsberechtigungen erstellen, wie weiter unten in diesem Abschnitt beschrieben.

- CreateOpsItem
- CreateResourceDataSync
- DescribeOpsItems
- DeleteResourceDataSync
- GetOpsItem
- GetOpsSummary
- ListResourceDataSync
- UpdateOpsItem

- UpdateResourceDataSync

Wenn Sie möchten, können Sie eine schreibgeschützte Berechtigung angeben, indem Sie Ihrem Konto, Ihrer Gruppe oder Rolle die folgende Inline-Richtlinie hinzufügen.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetOpsItem",
        "ssm:GetOpsSummary",
        "ssm:DescribeOpsItems",
        "ssm:GetServiceSetting",
        "ssm:ListResourceDataSync"
      ],
      "Resource": "*"
    }
  ]
}
```

Weitere Informationen zum Erstellen von IAM-Benutzer-Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch. Weitere Informationen zum Zuweisen dieser Richtlinie zu einer IAM-Gruppe finden Sie unter Zuordnen einer Richtlinie zu einer IAM-Gruppe.

Erstellen Sie wie folgt eine Berechtigung und fügen Sie sie Ihren Benutzern, Gruppen oder Rollen hinzu:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "ssm:GetOpsItem",
        "ssm:UpdateOpsItem",
        "ssm:DescribeOpsItems",
        "ssm:CreateOpsItem",
        "ssm:CreateResourceDataSync",
        "ssm:DeleteResourceDataSync",
        "ssm:ListResourceDataSync",
        "ssm:UpdateResourceDataSync"

      ],
      "Resource": "*"
    }
  ]
}
```

Abhängig von der Identitätsanwendung, die Sie in Ihrer Organisation verwenden, können Sie eine der folgenden Optionen auswählen, um den Benutzerzugriff zu konfigurieren.

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

  Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center -Benutzerhandbuch.
- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

  Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anleitung unter Eine Rolle für einen externen Identitätsanbieter (Verbund) erstellen im IAM-Benutzerhandbuch.
- IAM-Benutzer:
  - Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Befolgen Sie die Anleitung unter Eine Rolle für einen IAM-Benutzer erstellen im IAM-Benutzerhandbuch.
  - (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter Hinzufügen von Berechtigungen zu einem Benutzer (Konsole) im IAM-Benutzerhandbuch.

Einschränken des Zugriffs auf OpsItems mithilfe von Tags

Sie können auch den Zugriff auf OpsItems einschränken, indem Sie eine IAM-Inlinerichtlinie verwenden, die Tags festlegt. Hier sehen Sie ein Beispiel, das einen Tag-Schlüssel für Abteilung und

einen Tag-Wert für Finanzen angibt. Mit dieser Richtlinie kann der Benutzer den GetOpsItemAPI-Vorgang nur aufrufen, um Daten anzuzeigenOpsItems, die zuvor mit Key=Department und Value=Finance gekennzeichnet waren. Benutzer haben keinen Zugriff auf andere OpsItems.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetOpsItem"
             ],
      "Resource": "*"
      ,
      "Condition": { "StringEquals": { "ssm:resourceTag/Department":
 "Finance" } }
    }
  ]
}
```

Hier sehen Sie ein Beispiel, das API-Vorgänge zum Einsehen und Aktualisieren von OpsItems angibt. Diese Richtlinie gibt auch zwei Sätze von Tag-Schlüssel-Wert-Paaren an: Abteilung-Finanzen und Projekt-Unity.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ssm:GetOpsItem",
            "ssm:UpdateOpsItem"
         ],
         "Resource":"*",
         "Condition":{
            "StringEquals":{
```

```
                "ssm:resourceTag/Department":"Finance",
                "ssm:resourceTag/Project":"Unity"
            }
        }
    }
  ]
}
```

Weitere Informationen zum Hinzufügen von Tags zu einem OpsItem finden Sie unter Manuelles Erstellen der OpsItems.

Grundlegendes zu den von der integrierten Einrichtung erstellten EventBridge-Standardregeln

Während des integrierten Einrichtungsprozesses für Explorer und OpsCenter können Sie eine Reihe von Standardregeln aktivieren, die auf Ereignissen basieren, die von Amazon EventBridge erkannt wurden. Wenn diese Ereignisse erkannt werden, erstellt das System automatisch OpsItems in AWS Systems Manager OpsCenter.

Beispielsweise führt die Regel `SSMOpsItems-Autoscaling-instance-termination-failure` dazu, dass ein OpsItem erstellt wird, wenn die Beendigung einer EC2-Auto-Scaling-Instance fehlschlägt.

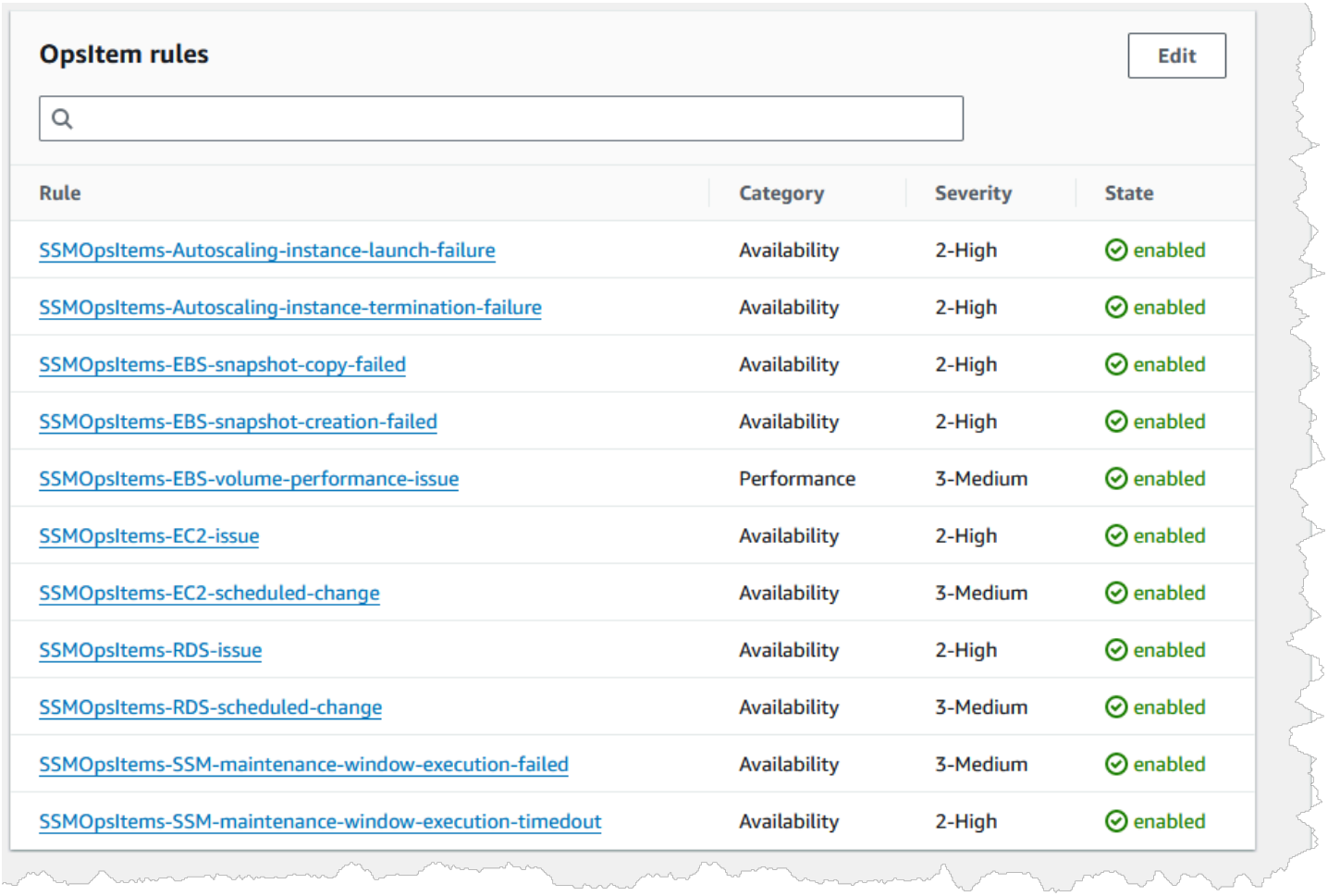
Die Regel `SSMOpsItems-SSM-maintenance-window-execution-failed` führt dazu, dass OpsItem erstellt wird, wenn ein Systems-Manager-Wartungsfenster nicht erfolgreich abgeschlossen werden kann.

Anweisungen zur Einrichtung und Beschreibungen aller EventBridge-Regeln, die Sie während des Einrichtungsvorgangs aktivieren können, finden Sie unter Einrichten von OpsCenter.

Wenn Sie nicht möchten, dass EventBridge OpsItems für diese Ereignisse erstellt, können Sie diese Option im integrierten Setup nicht aktivieren. Wenn Sie möchten, können Sie OpsCenter als Ziel bestimmter EventBridge-Ereignisse angeben. Weitere Informationen finden Sie unter Zu erstellende EventBridge Regeln konfigurieren OpsItems.

Sie können eine Standardregel deaktivieren oder ihre Kategorie und Schweregrad auf der Seite OpsCenter Einstellungen ändern, indem Sie OpsCenter-Einstellungen und dann im OpsItem-Regelbereich Bearbeiten wählen.

Sie können in der Systems-Manager-Konsole auch die Kategorie oder den Schweregrad bearbeiten, die einem einzelnen OpsItem zugewiesen sind, das aus diesen Regeln erstellt wurde. Weitere Informationen finden Sie unter Bearbeitung eines OpsItem.



Konfigurierung eines delegierten Administrators für Explorer

Wenn Sie AWS Systems Manager-Explorer-Daten von mehreren AWS-Regionen und Konten zusammenfassen, indem Sie die Ressourcendaten-Synchronisierung mit AWS Organizations verwenden, empfehlen wir, dass Sie einen delegierten Administrator für Explorer konfigurieren. Ein delegierter Administrator verbessert die Explorer-Sicherheit auf aus folgenden Gründen.

- Sie schränken die Anzahl der Explorer-Administratoren, die Ressourcendatensynchronisierungen für mehrere Konten und Regionen erstellen oder löschen können, auf ein einzelnes AWS-Konto.
- Sie müssen nicht mehr im AWS Organizations-Verwaltungskonto angemeldet sein, um die Ressourcendatensynchronisierung in Explorer zu verwalten.

Ein delegierter Administrator kann die folgenden APIs zur Synchronisierung von Explorer-Ressourcendaten über die Konsole, das SDK, AWS Command Line Interface (AWS CLI) oder AWS Tools for Windows PowerShell verwenden:

- CreateResourceDataSync
- DeleteResourceDataSync
- ListResourceDataSync
- UpdateResourceDataSync

Ein delegierter Administrator kann Explorer-Daten in der Konsole oder mithilfe von programmgesteuerten Tools wie dem SDK, der AWS CLI oder AWS Tools for Windows PowerShell durchsuchen, filtern und aggregieren. Suchen, Filtern und die Aggregation von Daten verwenden den API-Vorgang GetOpsSummary.

Ein delegierter Administrator kann maximal fünf Ressourcendaten-Synchronisierungen für eine gesamte Organisation oder eine Untergruppe von Organisationseinheiten erstellen. Ressourcendatensynchronisierungen, die von einem delegierten Administrator erstellt wurden, sind nur in dem delegierten Administratorkonto verfügbar. Sie können die Synchronisierung und die aggregierten Daten auch nicht im AWS Organizations-Verwaltungskonto anzeigen.

> **Note**
>
> Sie können ein delegiertes Administratorkonto nicht verwenden, um in Opt-in-AWS-Regionen eine Ressourcendaten-Synchronisierung zu erstellen. Sie müssen ein AWS Organizations-Verwaltungskonto verwenden.

Informationen zur Ressourcendatensynchronisierung finden Sie unter Einrichten von Systems Manager Explorer, um Daten aus mehreren Konten und Regionen anzuzeigen. Weitere Informationen zu AWS Organizations, finden Sie unter Was ist AWS Organizations? im AWS Organizations-Benutzerhandbuch.

Themen

- Bevor Sie beginnen
- Konfigurieren eines delegierten Explorer-Administrators
- Registrieren eines delegierten Explorer-Administrators

Bevor Sie beginnen

Die folgende Liste enthält wichtige Informationen zur Delegierung der Administration von Explorer.

- Sie können nur ein Konto für die Administration von Explorer delegieren.
- Die Konto-ID, die Sie als delegierter Explorer-Administrator angeben, muss als Mitgliedskonto in AWS Organizations aufgeführt sein. Weitere Informationen finden Sie unter Erstellen eines AWS-Konto in Ihrer Organisation im AWS Organizations-Benutzerhandbuch.
- Ein delegierter Administrator kann alle Explorer-API-Vorgänge zur Ressourcendatensynchronisierung in der Konsole oder mithilfe von programmgesteuerten Tools wie dem SDK, der AWS Command Line Interface (AWS CLI), oder AWS Tools for Windows PowerShell verwenden. Zu den API-Vorgängen für die Ressourcendatensynchronisierung gehören u. a: CreateResourceDataSync, DeleteResourceDataSync, ListResourceDataSync und UpdateResourceDataSync.
- Ein delegierter Administrator kann Explorer-Daten in der Konsole oder mithilfe von programmgesteuerten Tools wie dem SDK, der AWS CLI oder AWS Tools for Windows PowerShell durchsuchen, filtern und aggregieren. Suchen, Filtern und die Aggregation von Daten verwenden den API-Vorgang GetOpsSummary.
- Ressourcendatensynchronisierungen, die von einem delegierten Administrator erstellt wurden, sind nur in dem delegierten Administratorkonto verfügbar. Sie können die Synchronisierung und die aggregierten Daten auch nicht im AWS Organizations-Verwaltungskonto anzeigen.
- Ein delegierter Administrator kann maximal fünf Ressourcendatensynchronisierungen erstellen.
- Ein delegierter Administrator kann in AWS Organizations eine Ressourcendatensynchronisierung für eine ganze Organisation oder für eine Auswahl von Organisationseinheiten erstellen.

Konfigurieren eines delegierten Explorer-Administrators

Gehen Sie wie folgt vor, um einen delegierten Explorer-Administrator zu registrieren.

So registrieren Sie einen delegierten Explorer-Administrator

1. Loggen Sie sich in Ihr AWS Organizations Verwaltungskonto ein.
2. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
3. Wählen Sie im Navigationsbereich Explorer aus.
4. Wählen Sie Einstellungen aus.
5. Überprüfen Sie im Abschnott Delegierter Administrator für Explorer, ob Sie die erforderlichen dienstverknüpften Rollen- und Servicezugriffsoptionen konfiguriert haben. Wählen Sie bei Bedarf

die Schaltflächen Create role (Rolle erstellen) und Enable access (Zugriff gewähren) aus, um diese Optionen zu konfigurieren.

6. Geben Sie als Konto-ID die AWS-Konto ID ein. Bei diesem Konto muss es sich um ein Mitgliedskonto in handeln AWS Organizations.
7. Wählen Sie Register delegated administrator (Delegierten Administrator registrieren).

Der delegierte Administrator hat jetzt Zugriff auf die AWS Organizations Optionen Alle Konten aus meiner AWS Organizations Konfiguration einbeziehen und Organisationseinheiten auswählen auf der Seite Ressourcendatensynchronisierung erstellen.

Registrieren eines delegierten Explorer-Administrators

Gehen Sie wie folgt vor, um die Registrierung eines delegierten Explorer-Administrators aufzuheben. Die Registrierung eines delegierten Administratorkonto kann nur vom AWS Organizations -Verwaltungskonto aufgehoben werden. Wenn die Registrierung eines delegierten Administratorkontos aufgehoben wird, löscht das System alle vom delegierten Administrator erstellten AWS Organizations Ressourcendatensynchronisationen.

So heben Sie die Registrierung eines delegierten Explorer-Administrators auf

1. Melden Sie sich bei Ihrem Verwaltungskonto an. AWS Organizations
2. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
3. Wählen Sie im Navigationsbereich Explorer aus.
4. Wählen Sie Einstellungen aus.
5. Wählen Sie Abmelden im Abschnitt Delegierter Administrator für Explorer. Das System zeigt eine Warnung an.
6. Geben Sie die Konto-ID ein und wählen Sie Remove (Entfernen).

Das Konto hat keinen Zugriff mehr auf die API-Operationen zur AWS Organizations Ressourcendatensynchronisierung. Das System löscht alle vom Konto erstellten AWS Organizations Ressourcendatensynchronisierungen.

Einrichten von Systems Manager Explorer, um Daten aus mehreren Konten und Regionen anzuzeigen

AWS Systems Manager verwendet eine integrierte Einrichtungsumgebung, um Ihnen den Einstieg in AWS Systems Manager-Explorer und AWS Systems Manager OpsCenter zu erleichtern. Nach Abschluss der integrierten Einrichtung synchronisieren Explorer und OpsCenter automatisch die Daten. Genauer gesagt synchronisieren diese Tools OpsData und OpsItems für das AWS-Konto und die AWS-Region, die Sie beim Abschluss der integrierten Einrichtung verwendet haben. Wenn Sie OpsData und OpsItems von anderen Konten und Regionen aggregieren möchten, müssen Sie eine Ressourcendatensynchronisierung erstellen, wie in diesem Thema beschrieben.

> **Note**
>
> Weitere Hinweise zum integrierten Setup finden Sie unter Erste Schritte mit Systems Manager Explorer und OpsCenter.

Themen

- Ressourcendatensynchronisierungen verwalten für Explorer
- Verständnis der Synchronisierung von Daten mehrerer Konto- und Regions-Ressourcendaten
- Erstellen einer Ressourcendatensynchronisierung

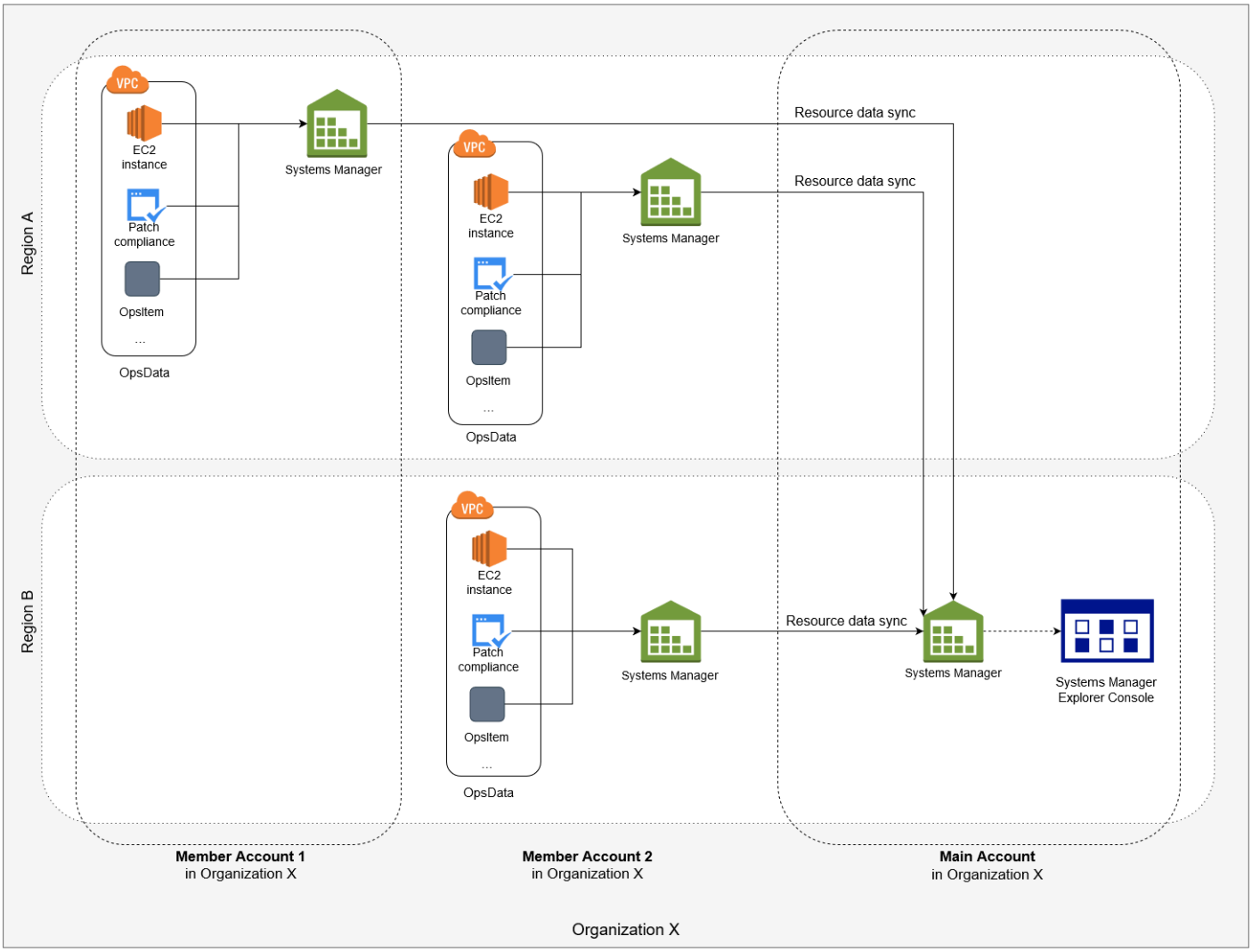
Ressourcendatensynchronisierungen verwalten für Explorer

Die Ressourcendatensynchronisierung für Explorer bietet zwei Aggregierungsoptionen:

- Einzelkonto/mehrere Regionen: Sie können so konfigurierenExplorer, dass OpsData Daten aus mehreren OpsItems aggregiert werden AWS-Regionen, aber der Datensatz ist auf den aktuellen Datensatz beschränkt. AWS-Konto
- Multiple-accounts/Multiple-regions (Mehrere Konten/Mehrere Regionen): Sie können Explorer so konfigurieren, dass Daten aus mehreren AWS-Regionen und -Konten aggregiert werden. Diese Option erfordert die Einrichtung und Konfiguration von AWS Organizations. Nach der Einrichtung und Konfiguration AWS Organizations können Sie Daten nach Organisationseinheit (OU) oder für eine gesamte Organisation zusammenfassen. Explorer Systems Manager aggregiert die Daten in das AWS Organizations -Verwaltungskonto vor der Anzeige in Explorer. Weitere Informationen finden Sie unter Was ist AWS Organizations? im AWS Organizations Benutzerhandbuch.

> **Warning**
>
> Wenn Sie so konfigurierenExplorer, dass Daten aus einer Organisation in aggregiert werden AWS Organizations, aktiviert OpsData das System alle Mitgliedskonten in der Organisation. Durch die Aktivierung von OpsData Quellen in allen Mitgliedskonten erhöht sich die Anzahl der Aufrufe von OpsCenter APIs Like CreateOpsItemund GetOpsSummary. Aufrufe dieser API-Aktionen werden Ihnen in Rechnung gestellt.

Das folgende Diagramm zeigt eine Ressourcendatensynchronisierung, die für die Arbeit mit AWS Organizations konfiguriert ist. In diesem Szenario hat der Benutzer zwei Konten in AWS Organizations definiert. Bei Explorer der Synchronisierung von Ressourcendaten werden Daten aus beiden Konten und mehreren AWS-Regionen Konten im AWS Organizations Verwaltungskonto zusammengefasst, wo sie dann angezeigt werden.



Verständnis der Synchronisierung von Daten mehrerer Konto- und Regions-Ressourcendaten

In diesem Abschnitt werden wichtige Details zur Synchronisierung von mehreren Konto- und mehreren Regions-Ressourcendaten beschrieben, die AWS Organizations verwenden. Die Informationen in diesem Abschnitt gelten insbesondere, wenn Sie auf der Seite Erstellen von Ressourcendaten-Synchronisierung eine der folgenden Optionen wählen:

- Schließt alle Konten aus meiner AWS Organizations Konfiguration ein
- Wählen Sie Organisationseinheiten in AWS Organizations

Wenn Sie keine dieser Optionen verwenden möchten, können Sie diesen Abschnitt überspringen.

Wenn Sie eine Ressourcendatensynchronisierung in der SSM-Konsole erstellen und eine der AWS Organizations Optionen wählen, lässt Systems Manager automatisch alle OpsData Quellen in den ausgewählten Regionen für alle AWS-Konten in Ihrer Organisation (oder in den ausgewählten Organisationseinheiten) zu. Wenn Sie beispielsweise in einer Region nicht Explorer aktiviert haben, wenn Sie eine AWS Organizations Option für Ihre Ressourcendatensynchronisierung auswählen, sammelt Systems Manager automatisch Daten OpsData aus dieser Region. Um eine Ressourcendatensynchronisierung zu erstellen, ohne OpsData Quellen zuzulassen, geben Sie EnableAllOpsDataSourcesbei der Erstellung der Datensynchronisierung den Wert False an. Weitere Informationen finden Sie in den `EnableAllOpsDataSources` Parameterdetails für den ResourceDataSyncSourceDatentyp in der Amazon EC2 Systems Manager API-Referenz.

Wenn Sie keine der AWS Organizations Optionen für eine Ressourcendatensynchronisierung wählen, müssen Sie das integrierte Setup für jedes Konto und jede Region abschließen, in der Sie auf Daten zugreifen Explorer möchten. Wenn Sie dies nicht tun, Explorer werden auch nicht OpsItems für die Konten und Regionen angezeigt OpsData , in denen Sie das Integrierte Setup nicht abgeschlossen haben.

Wenn Sie Ihrer Organisation ein Kinderkonto hinzufügen, Explorer werden automatisch alle OpsData Quellen für das Konto zugelassen. Wenn Sie das Kinderkonto zu einem späteren Zeitpunkt aus Ihrer Organisation entfernen, werden Explorer weiterhin Daten über das Konto gesammelt OpsData .

Wenn Sie eine bestehende Ressourcendatensynchronisierung aktualisieren, die eine der AWS Organizations Optionen verwendet, werden Sie vom System aufgefordert, die Erfassung aller OpsData Quellen für alle Konten und Regionen zu genehmigen, die von der Änderung betroffen sind.

Wenn Sie Ihrem AWS-Konto Service einen neuen Service hinzufügen und wenn OpsData für diesen Service Explorer sammelt, konfiguriert Explorer Systems Manager automatisch, dass dieser OpsData

Service erfasst wird. Wenn Ihre Organisation beispielsweise AWS Trusted Advisor bei der Erstellung noch keine Ressourcendatensynchronisierung verwendet hat, Ihre Organisation sich aber für diesen Dienst angemeldet hat, werden Ihre Explorer Ressourcendatensynchronisierungen automatisch aktualisiert, um diese Daten zu erfassen. OpsData

Important

Beachten Sie die folgenden wichtigen Informationen über mehrere Konto- und Regions-Ressourcendatensynchronisierungen:

- Durch das Löschen einer Ressourcendatensynchronisierung wird eine OpsData Quelle in Explorer nicht deaktiviert.
- Um mehrere Konten anzeigen OpsData zu können, muss der Modus AWS Organizations Alle Funktionen aktiviert sein und Sie müssen mit dem AWS Organizations Verwaltungskonto angemeldet sein. OpsItems
- Die meisten AWS-Regionen sind standardmäßig für Sie aktiv AWS-Konto, aber bestimmte Regionen werden nur aktiviert, wenn Sie sie manuell auswählen. Diese Regionen werden als Opt-in-Regionen bezeichnet. Standardmäßig unterstützen konten-/regionsübergreifende Ressourcendaten-Synchronisierungen in Explorer keine Datenaggregation in Opt-in-Regionen. Am 30. Juni 2025 wurde Support für die folgenden Opt-In-Regionen hinzugefügt.
  - Europa (Milan)
  - Afrika (Kapstadt)
  - Middle East (Bahrain)
  - Asien-Pazifik (Hongkong)

  Beachten Sie, dass Sie kein delegiertes Administratorkonto verwenden können, um eine Ressourcendatensynchronisierung in Opt-in-Regionen zu erstellen. Sie müssen ein AWS Organizations -Verwaltungskonto verwenden.

Erstellen einer Ressourcendatensynchronisierung

Beachten Sie die folgenden Details, bevor Sie eine Ressourcendatensynchronisierung für Explorer konfigurieren.

- Explorer unterstützt maximal fünf Ressourcendatensynchronisierungen.

- Nachdem Sie eine Ressourcendatensynchronisierung für eine Region erstellt haben, können Sie die Kontooptionen für diese Synchronisierung nicht ändern. Wenn Sie beispielsweise eine Synchronisierung in der Region US-East-2 (Ohio) erstellen und die Option Nur das aktuelle Konto einbeziehen auswählen, können Sie diese Synchronisierung später nicht bearbeiten und die Option Alle Konten aus meiner AWS Organizations Konfiguration einbeziehen wählen. Stattdessen müssen Sie die erste Ressourcendatensynchronisierung löschen und eine neue erstellen.
- OpsData angezeigt in Explorer ist schreibgeschützt.

Gehen Sie folgendermaßen vor, um eine Ressourcendatensynchronisierung für Explorer zu erstellen.

Erstellen einer Resource Data Sync

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Wählen Sie Einstellungen aus.
4. Wählen Sie im Abschnitt Configure resource data sync (Ressourcendatensynchronisierung konfigurieren) die Option Create resource data sync (Ressourcendatensynchronisierung erstellen) aus.
5. Geben Sie unter Resource data sync name (Name der Ressourcen-Datensynchronisierung) einen Namen ein.
6. Wählen Sie im Abschnitt Add accounts (Konten hinzufügen) eine Option aus.

   > **Note**
   >
   > Um eine der AWS Organizations Optionen verwenden zu können, müssen Sie mit dem AWS Organizations Verwaltungskonto oder mit einem Explorer delegierten Administratorkonto angemeldet sein. Weitere Informationen zu dem delegierten Administratorkonto finden Sie unter Konfigurierung eines delegierten Administrators für Explorer.

7. Wählen Sie im Abschnitt Regions to include (Einzubeziehende Regionen) eine der folgenden Optionen aus.

- Wählen Sie Alle aktuellen und zukünftigen Regionen, um Daten aus allen aktuellen AWS-Regionen und allen neuen Regionen, die in future online gehen, automatisch zu synchronisieren.
- Wählen Sie Alle Regionen, um Daten aus allen aktuellen Regionen automatisch zu synchronisieren AWS-Regionen.
- Wählen Sie Regionen, die Sie einbeziehen möchten, einzeln aus.

8. Wählen Sie Ressourcen-Datensynchronisierung erstellen.

Nach dem Erstellen einer Ressourcendatensynchronisierung kann das System einige Minuten benötigen, bis Explorer mit Daten gefüllt wird. Sie können die Synchronisierung anzeigen, indem Sie sie aus der Liste Select a resource data sync (Ressourcen-Datensynchronisierung auswählen) in Explorer auswählen.

Verwenden von Explorer

Dieser Abschnitt enthält Informationen zum Anpassen von AWS Systems Manager-Explorer durch Ändern des Widget-Layouts und durch Ändern der im Dashboard angezeigten Daten.

Inhalt

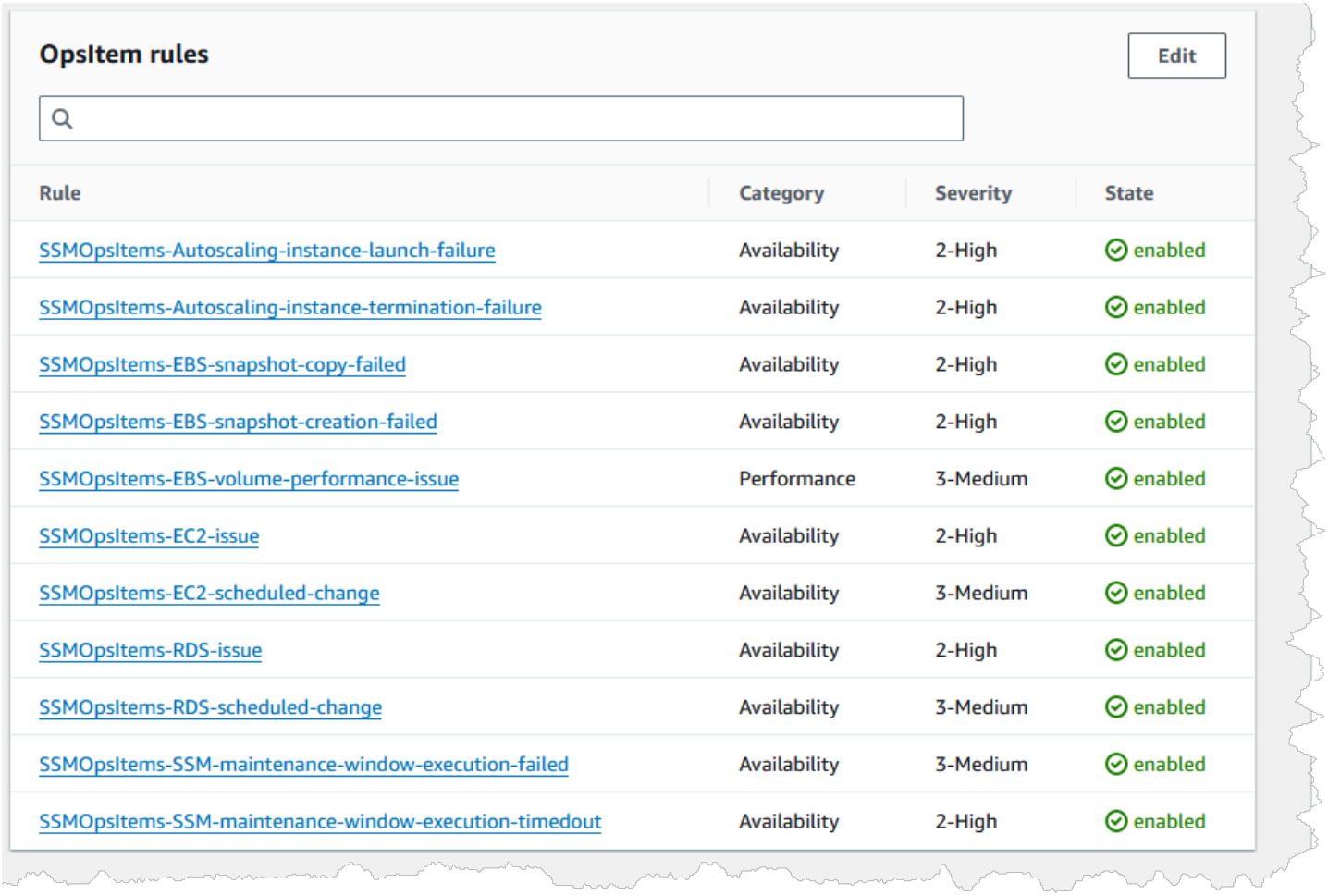
- Bearbeiten von EventBridge-Regeln, die erstellt wurden für Explorer
- Bearbeiten von Systems-Manager-Explorer-Datenquellen
- Anpassen der Anzeige Explorer
- Empfangen von Ergebnissen von AWS Security Hub CSPM in Explorer
- Löschen einer Systems-Manager-Explorer-Ressourcendatensynchronisierung

Bearbeiten von EventBridge-Regeln, die erstellt wurden für Explorer

Wenn Sie das integrierte Setup abgeschlossen haben, werden mehr als ein Dutzend Regeln in Amazon EventBridge aktiviert. Diese Regeln erstellen automatisch OpsItems in AWS Systems Manager-OpsCenter. AWS Systems Manager-Explorer zeigt anschließend aggregierte Informationen über OpsItems an.

Jede Regel enthält einen voreingestellten Wert für Category (Kategorie) und Severity (Schweregrad). Wenn das System OpsItems aus einem Ereignis erstellt, weist es automatisch die voreingestellten Werte für Category (Kategorie) und Severity (Schweregrad) zu.

> **Important**
>
> Sie können die Werte für Category (Kategorie) und Severity (Schweregrad) für Standardregeln nicht bearbeiten, Sie können diese Werte jedoch an OpsItems bearbeiten, die aus den Standardregeln erstellt wurden.



Bearbeiten von Standardregeln zum Erstellen von OpsItems

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Wählen Sie Einstellungen aus.
4. Wählen Sie im Abschnitt OpsItems rules (OIS-Regeln) Edit (Bearbeiten).
5. Erweitern Sie CWE rules (CWE-Regeln).

6. Deaktivieren Sie das Kontrollkästchen neben den Regeln, die Sie nicht verwenden möchten.
7. Verwenden Sie die Listen Category (Kategorie) und Severity (Schweregrad), um diese Informationen für eine Regel zu ändern.
8. Wählen Sie Speichern.

Ihre Änderungen werden wirksam, wenn das System das nächste Mal einen OpsItem erstellt.

Bearbeiten von Systems-Manager-Explorer-Datenquellen

Da AWS-Regionen diese standardmäßig verfügbar sind, zeigt der AWS Systems Manager Explorer Daten aus den folgenden Quellen an. Sie können Explorer-Einstellungen zum Hinzufügen oder Entfernen von Datenquellen bearbeiten:

- Amazon Elastic Compute Cloud (Amazon EC2)
- AWS Systems Manager -Bestand
- AWS Systems Manager OpsCenter OpsItems
- AWS Systems Manager Patch Manager Patch Compliance
- AWS Systems Manager State ManagerEinhaltung der Vorschriften durch die
- AWS Trusted Advisor
- AWS Compute Optimizer
- AWS -Support Fälle im Mittelpunkt
- AWS Config Einhaltung von Regeln und Ressourcen
- AWS Security Hub CSPM Feststellungen

> **Note**
>
> Für die Region Asien-Pazifik (Osaka) werden Explorer keine Daten AWS Compute Optimizer und AWS Security Hub CSPM Ergebnisse angezeigt.

Für die Opt-in-AWS -Regionen zeigt Explorer Daten aus den folgenden Quellen an:

- Amazon Elastic Compute Cloud (Amazon EC2)
- AWS Systems Manager -Bestand

- AWS Systems Manager OpsCenter OpsItems
- AWS Systems Manager Patch Manager Patch Compliance
- AWS Systems Manager State ManagerEinhaltung der Vorschriften durch die
- AWS Trusted Advisor
- AWS -Support Fälle im Mittelpunkt
- AWS Config Einhaltung von Regeln und Ressourcen

> **Note**
>
> - Um Anfragen im Support Center einsehen zu könnenExplorer, müssen Sie entweder ein Enterprise- oder ein Business-Konto bei eingerichtet haben Support.
> - Sie können Explorer nicht so konfigurieren, dass OpsCenter-OpsItem-Daten nicht mehr angezeigt werden.

Bevor Sie beginnen

Stellen Sie sicher, dass Sie Services eingerichtet und konfiguriert haben, die Explorer-Widgets mit Daten füllen. Weitere Informationen finden Sie unter Einrichten von zugehörigen Services für Explorer.

So bearbeiten Sie Datenquellen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Wählen Sie Einstellungen und anschließend die Registerkarte Dashboard konfigurieren.
4. Aktivieren oder deaktivieren Sie im Bereich OpsData Quellen in der Spalte Status die Quellen entsprechend den Daten, die Sie anzeigen möchten.

Anpassen der Anzeige Explorer

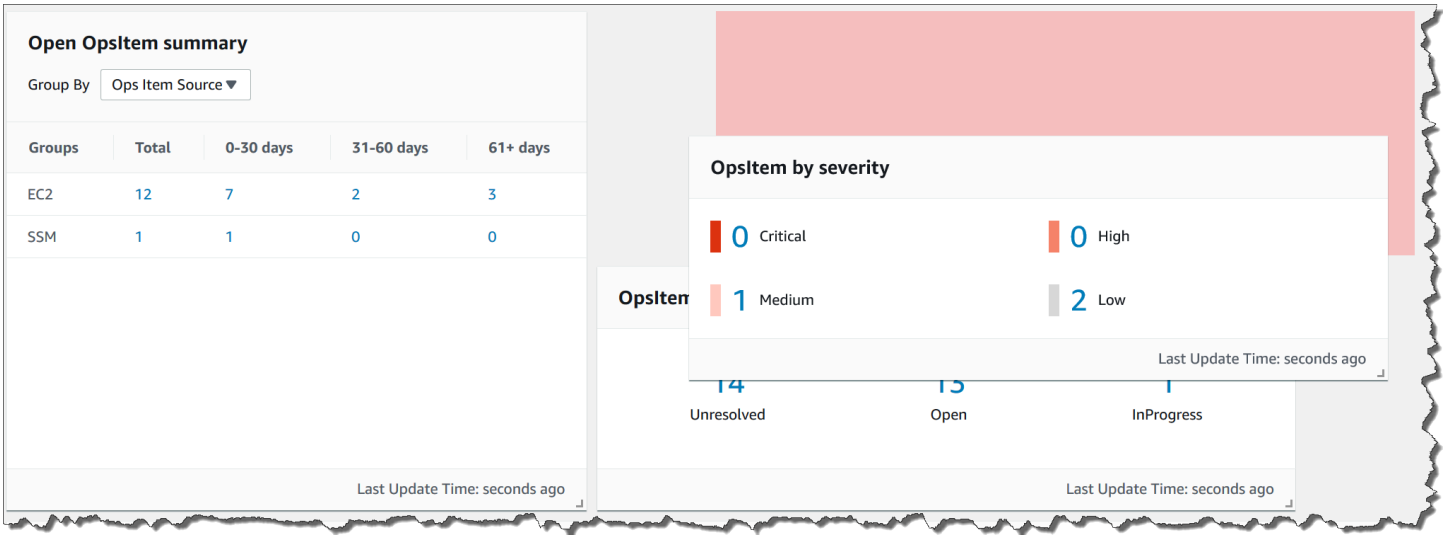
Sie können das Widget-Layout in AWS Systems Manager-Explorer mithilfe einer Drag-&-Drop-Funktion anpassen. Sie können OpsData und OpsItems, die in Explorer angezeigt werden, auch mithilfe von Filtern anpassen, wie in diesem Thema beschrieben.

Vergewissern Sie sich vor dem Anpassen des Widget-Layouts, dass die Widgets, die Sie anzeigen möchten, derzeit in Explorer angezeigt werden. Um einige Widgets in Explorer (z. B. das AWS Config-Compliance-Widget) anzuzeigen, müssen Sie sie auf der Seite Configure dashboard (Dashboard konfigurieren) aktivieren.

So aktivieren Sie die Anzeige von Widgets in Explorer

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Wählen Sie Dashboard actions (Dashboard-Aktionen), Configure dashboard (Dashboard konfigurieren).
4. Wählen Sie die Registerkarte Configure Dashboard (Dashboard konfigurieren).
5. Wählen Sie entweder Enable all (Alle aktivieren) oder aktivieren Sie ein einzelnes Widget oder eine einzelne Datenquelle.
6. Wählen Sie Explorer, um Ihre Änderungen anzuzeigen.

Um das Widget-Layout in Explorer anzupassen, wählen Sie ein Widget aus, das Sie verschieben möchten. Klicken Sie auf den Namen des Widgets, halten Sie es und ziehen Sie es dann an seine neue Position.



Wiederholen Sie diesen Vorgang für jedes Widget, das Sie neu positionieren möchten.

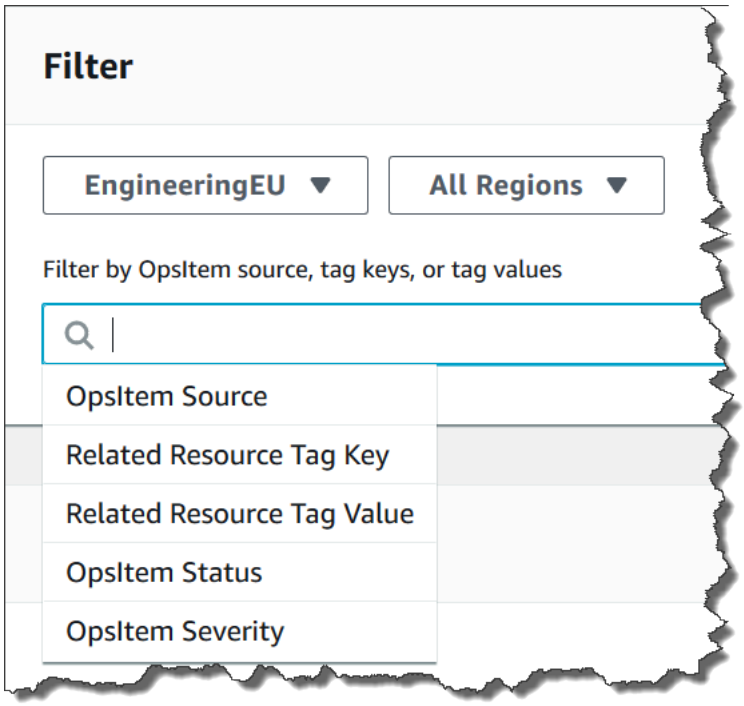
Wenn Sie sich entscheiden, dass Ihnen das neue Layout nicht gefällt, wählen Sie Reset layout (Layout zurücksetzen), um alle Widgets wieder an ihren ursprünglichen Speicherort zu verschieben.

Verwenden von Filtern zum Ändern der angezeigten Daten in Explorer

Standardmäßig zeigt Explorer Daten für das aktuelle AWS-Konto und die aktuelle Region an. Wenn Sie mindestens eine Ressourcen-Datensynchronisierung erstellen, können Sie mithilfe von Filtern ändern, welche Synchronisierung aktiv ist. Sie können dann wählen, ob Daten für eine bestimmte Region oder für alle Regionen angezeigt werden sollen. Sie können die Suchleiste auch verwenden, um nach verschiedenen OpsItem- und Schlüssel-Tag-Kriterien zu filtern.

Ändern der in Explorer angezeigten Daten mithilfe von Filtern

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Verwenden Sie im Abschnitt Filter die Liste Select a resource data sync (Ressourcen-Datensynchronisierung auswählen), um eine Synchronisierung auszuwählen.
4. Verwenden Sie die Liste Regions (Regionen), um entweder eine bestimmte AWS-Region oder All Regions (Alle Regionen) auszuwählen.
5. Wählen Sie die Suchleiste und dann die Kriterien aus, nach denen die Daten gefiltert werden sollen.



6. Drücken Sie die Eingabetaste.

Explorer behält die ausgewählten Filteroptionen bei, wenn Sie die Seite schließen und erneut öffnen.

Empfangen von Ergebnissen von AWS Security Hub CSPM in Explorer

AWS Security Hub CSPMbietet einen umfassenden Überblick über Ihren Sicherheitsstatus in. AWS Der Dienst sammelt Sicherheitsdaten, so genannte Erkenntnisse, aus allen AWS-Konten, Diensten und unterstützten Produkten von Drittanbietern. Die CSPM-Ergebnisse von Security Hub können Ihnen helfen, Ihre Umgebung anhand von Industriestandards und Best Practices zu überprüfen, Ihre Sicherheitstrends zu analysieren und Sicherheitsprobleme mit der höchsten Priorität zu identifizieren.

Security Hub CSPM sendet Ergebnisse an Amazon EventBridge, das die Ergebnisse mithilfe einer Ereignisregel an Amazon sendet. Explorer Nachdem Sie die Integration wie hier beschrieben aktiviert haben, können Sie die Ergebnisse des Security Hub CSPM in einem Explorer Widget und die Details zu den Ergebnissen in anzeigen. OpsCenter OpsItems Das Widget bietet eine Zusammenfassung aller Security Hub CSPM-Ergebnisse basierend auf ihrem Schweregrad. Neue Erkenntnisse in Security Hub CSPM sind in der Regel Explorer innerhalb von Sekunden nach ihrer Erstellung sichtbar.

⚠ Warning

Beachten Sie die folgenden wichtigen Informationen:

- Explorer ist in OpsCenter integriert, ein Tool in Systems Manager. Nachdem Sie die Explorer Integration mit Security Hub CSPM aktiviert haben, OpsCenter werden automatisch CSPM-Ergebnisse OpsItems für Security Hub erstellt. Abhängig von Ihrer AWS Umgebung kann die Aktivierung der Integration zu einer großen Anzahl von Daten führenOpsItems, was mit Kosten verbunden ist.

  Bevor Sie fortfahren, lesen Sie mehr über die OpsCenter Integration mit Security Hub CSPM. Das Thema enthält spezifische Details darüber, wie Änderungen und Aktualisierungen von Erkenntnissen und OpsItems Ihrem Konto berechnet werden. Weitere Informationen finden Sie unter Verständnis der OpsCenter Integration mit AWS Security Hub CSPM. OpsCenter-Preisinformationen finden Sie unter AWS Systems Manager - Preise.
- Wenn Sie eine Ressourcendatensynchronisierung erstellen, Explorer während Sie mit dem Administratorkonto angemeldet sind, wird die Security Hub CSPM-Integration automatisch für den Administrator und alle Mitgliedskonten in der Synchronisierung aktiviert. Nach der Aktivierung OpsCenter werden automatisch kostenpflichtige CSPM-Ergebnisse OpsItems für Security Hub erstellt. Weitere Informationen zum Erstellen einer Ressourcendaten-

Synchronisierung finden Sie unter Einrichten von Systems Manager Explorer, um Daten aus mehreren Konten und Regionen anzuzeigen.

Ergebnisarten, die Explorer empfängt

Explorererhält alle Ergebnisse von Security Hub CSPM. Sie können alle Ergebnisse basierend auf dem Schweregrad im Explorer Widget sehen, wenn Sie die Security Hub CSPM-Standardeinstellungen aktivieren. Standardmäßig erstellt Explorer OpsItems für kritische und hochgradig schwerwiegende Befunde. Sie können Explorer manuell konfigurieren, um OpsItems für Erkenntnisse mit mittlerem und niedrigem Schweregrad zu erstellen.

ExplorerEs wird zwar nicht OpsItems für informative Ergebnisse erstellt, Sie können jedoch informative Betriebsdaten (OpsData) im Security Hub CSPM-Ergebniszusammenfassungs-Widget anzeigen. Explorererstellt OpsData für alle Ergebnisse unabhängig vom Schweregrad. Weitere Informationen zu den Schweregraden von Security Hub CSPM finden Sie unter Schweregrad in der AWS Security Hub API-Referenz.

Aktivieren der Integration

In diesem Abschnitt wird beschrieben, wie Sie den Empfang von Security Hub CSPM-Ergebnissen aktivieren und konfigurierenExplorer.

Bevor Sie beginnen

Führen Sie die folgenden Aufgaben aus, bevor Sie Explorer den Empfang von Security Hub CSPM-Ergebnissen konfigurieren.

- Aktivieren und konfigurieren Sie Security Hub CSPM. Weitere Informationen finden Sie unter Security Hub CSPM einrichten im AWS Security Hub Benutzerhandbuch.
- Melden Sie sich beim AWS Organizations Verwaltungskonto an. Systems Manager benötigt Zugriff auf, AWS Organizations um OpsItems CSPM-Ergebnisse aus dem Security Hub erstellen zu können. Nachdem Sie sich beim Verwaltungskonto angemeldet haben, werden Sie aufgeforderter die Schaltfläche Enable access (Zugriff erlauben) auf der Registerkarte Explorer Configure Dashboard zu wählen, wie nachfolgend beschrieben: Wenn Sie sich nicht beim AWS Organizations Verwaltungskonto anmelden, können Sie keinen Zugriff gewähren und Explorer keine OpsItems CSPM-Ergebnisse aus Security Hub erstellen.

Um mit dem Empfang von Security Hub CSPM-Ergebnissen zu beginnen

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Klicken Sie auf Einstellungen.
4. Wählen Sie die Registerkarte Dashboard konfigurieren.
5. Wählen Sie AWS Security Hub CSPM.
6. Wählen Sie den Schieberegler Disabled, um AWS Security Hub CSPM zu aktivieren.

   Kritische und schwerwiegende Erkenntnisse werden standardmäßig angezeigt. Um Erkenntnisse mit mittlerem und niedrigem Schweregrad anzuzeigen, wählen Sie den Schieberegler Deaktiviert neben Mittel, Niedrig.
7. Wählen Sie im Abschnitt OpsItemsCreated by Security Hub CSPM Findings die Option Enable access aus. Wenn Sie diese Schaltfläche nicht sehen, melden Sie sich beim AWS Organizations Verwaltungskonto an und kehren Sie zu dieser Seite zurück, um die Schaltfläche auszuwählen.

So zeigen Sie die Ergebnisse von Security Hub CSPM an

Das folgende Verfahren beschreibt, wie Sie die CSPM-Ergebnisse von Security Hub einsehen können.

So zeigen Sie die CSPM-Ergebnisse von Security Hub an

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Suchen Sie das Widget AWS Security Hub CSPM findings summary (Ergebniszusammenfassung). Dadurch werden Ihre Security Hub CSPM-Ergebnisse angezeigt. Sie können einen Schweregrad auswählen, um eine detaillierte Beschreibung der entsprechenden OpsItem auszuwählen.

Empfangen von Ergebnisse stoppen

Das folgende Verfahren beschreibt, wie Sie den Empfang von Security Hub CSPM-Ergebnissen beenden können.

Um den Empfang von Security Hub CSPM-Ergebnissen zu beenden

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Klicken Sie auf Einstellungen.
4. Wählen Sie die Registerkarte Configure dashboard.
5. Wählen Sie den Schieberegler Enabled (aktiviert), um AWS Security Hub CSPM zu deaktivieren..

> **Important**
>
> Wenn die Option zum Deaktivieren der Security Hub CSPM-Ergebnisse in der Konsole abgeblendet ist, können Sie diese Einstellung deaktivieren, indem Sie den folgenden Befehl in der ausführen. AWS CLI Sie müssen den Befehl ausführen, während Sie entweder mit dem AWS Organizations Verwaltungskonto oder dem delegierten Administratorkonto von Systems Manager angemeldet sind. Geben Sie für den `region` Parameter an, AWS-Region wo Sie den Empfang von Security Hub CSPM-Ergebnissen beenden möchten. Explorer
>
> ```
> aws ssm update-service-setting --setting-id /ssm/opsdata/SecurityHub --setting-value Disabled --region AWS-Region
> ```
>
> Ein Beispiel:
>
> ```
> aws ssm update-service-setting --setting-id /ssm/opsdata/SecurityHub --setting-value Disabled --region us-east-1
> ```

Löschen einer Systems-Manager-Explorer-Ressourcendatensynchronisierung

In AWS Systems Manager-Explorer können Sie OpsData und OpsItems aus anderen Konten und Regionen zusammenfassen, indem Sie eine Ressourcendaten-Synchronisierung erstellen.

Sie können die Kontooptionen für eine Ressourcen-Datensynchronisierung nicht ändern. Wenn Sie beispielsweise eine Synchronisierung in der Region us-east-2 (Ohio) erstellt haben und die Option

Include only the current account (Nur das aktuelle Konto einschließen) ausgewählt haben, können Sie diese Synchronisierung später nicht bearbeiten und die Option Include all accounts from my AWS Organizations configuration (Alle Konten aus meiner -Konfiguration einschließen) auswählen. Stattdessen müssen Sie die Ressourcen-Datensynchronisierung löschen und eine neue erstellen, wie im folgenden Verfahren beschrieben.

Löschen einer Resource Data Sync

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Wählen Sie Einstellungen aus.
4. Wählen Sie im Abschnitt Configure resource data sync (Ressourcen-Datensynchronisierung konfigurieren) die Ressourcen-Datensynchronisierung aus, die Sie löschen möchten.
5. Wählen Sie Löschen aus.

OpsData Aus Systems Manager exportieren Explorer

Sie können 5.000 OpsData Artikel als Datei mit kommagetrennten Werten (.csv) aus AWS Systems Manager dem Explorer in einen Amazon Simple Storage Service (Amazon S3) -Bucket exportieren. Der Explorer verwendet das AWS-ExportOpsDataToS3Automatisierungs-Runbook für den Export. OpsData Beim Exportieren zeigt das System die Automations-Runbook-Seite an OpsData, auf der Sie Details wie AssumeRole, den Amazon S3 S3-Bucket-Namen, den SNS-Thema-ARN und die zu exportierenden Felder angeben können.

Um zu exportieren: OpsData

- Schritt 1: Festlegen eines SNS-Themas
- Schritt 2: (Optional) Datenexport konfigurieren
- Schritt 3: Exportieren OpsData

Schritt 1: Festlegen eines SNS-Themas

Wenn Sie den Datenexport konfigurieren, müssen Sie ein Amazon Simple Notification Service (Amazon SNS) -Thema angeben, das in derselben AWS-Region Datei vorhanden ist, in die Sie die Daten exportieren möchten. Wenn ein Export abgeschlossen ist, sendet Systems Manager eine

Benachrichtigung an das Amazon SNS-Thema. Informationen zum Erstellen eines Amazon-SNS-Themas finden Sie unter Erstellen eines Amazon-SNS-Themas.

Schritt 2: (Optional) Datenexport konfigurieren

Sie können die Einstellungen für den Datenexport auf der Seite Einstellungen oder Ops-Daten in S3-Bucket exportieren konfigurieren.

So konfigurieren Sie den Datenexport von Explorer

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Wählen Sie Einstellungen aus.
4. Wählen Sie im Abschnitt Configure data export (Datenexport konfigurieren) die Option Edit (Bearbeiten).
5. Um die Datenexport-Datei in einen vorhandenen Amazon-S3-Bucket hochzuladen, Wählen Sie einen vorhandenen S3-Bucket aus und wählen Sie den Bucket aus der Liste aus.

   Um die Datenexportdatei in einen neuen Amazon-S3-Bucket hochzuladen, wählen Sie Neuen S3-Bucket erstellen und geben den Namen ein, den Sie für den neuen Bucket verwenden möchten.

   > **Note**
   >
   > Sie können den Namen des Amazon-S3-Buckets und den Amazon-SNS-Themen-ARN nur auf der Seite bearbeiten, auf der Sie diese Einstellungen zum ersten Mal in Explorer konfiguriert haben. Wenn Sie den Amazon-S3-Bucket und den Amazon-SNS-Themen-ARN auf der Seite Einstellungen eingerichtet haben, können Sie diese Einstellungen nur auf der Seite Einstellungen ändern.

6. Wählen Sie unter Amazon-SNS-Themen-ARN auswählen das Thema aus, das Sie benachrichtigen möchten, wenn der Export abgeschlossen ist.
7. Wählen Sie Erstellen aus.

Schritt 3: Exportieren OpsData

Wenn Sie Explorer Daten exportieren, erstellt Systems Manager eine AWS Identity and Access Management (IAM-) Rolle mit dem Namen`AmazonSSMExplorerExportRole`. Diese Rolle verwendet die folgende IAM-Richtlinie.

JSON

```
{
      "Version":"2012-10-17",
      "Statement": [
          {
              "Sid": "OpsSummaryExportAutomationServiceRoleStatement1",
              "Effect": "Allow",
              "Action": [
                  "s3:PutObject"
              ],
              "Resource": [
                 "arn:aws:s3:::amzn-s3-demo-bucket/*"
              ]
          },
          {
              "Sid": "OpsSummaryExportAutomationServiceRoleStatement2",
              "Effect": "Allow",
              "Action": [
                  "s3:GetBucketAcl",
                  "s3:GetBucketLocation"
              ],
              "Resource": [
                 "arn:aws:s3:::amzn-s3-demo-bucket"
              ]
          },
          {
              "Sid": "OpsSummaryExportAutomationServiceRoleStatement3",
              "Effect": "Allow",
              "Action": [
                  "sns:Publish"
              ],
              "Resource": [
                  "arn:aws:sns:us-east-1:111122223333:SNSTopicName"
              ]
          },
          {
```

```
            "Sid": "OpsSummaryExportAutomationServiceRoleStatement4",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:MyLogGroup"
            ]
        },
        {
            "Sid": "OpsSummaryExportAutomationServiceRoleStatement5",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:PutLogEvents",
                "logs:CreateLogStream"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "OpsSummaryExportAutomationServiceRoleStatement6",
            "Effect": "Allow",
            "Action": [
                "ssm:GetOpsSummary"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Die Rolle umfasst die folgende Vertrauenseinheit.

JSON

```
{
      "Version":"2012-10-17",
      "Statement": [
```

```
        {
          "Sid": "OpsSummaryExportAutomationServiceRoleTrustPolicy",
          "Effect": "Allow",
          "Principal": {
            "Service": "ssm.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }
```

Um zu exportieren OpsData aus Explorer

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Explorer aus.
3. Wählen Sie einen Datenlink in einem Explorer-Widget aus. Wählen Sie im Widget OpsItems nach Status“ beispielsweise Ungelöste oder Offene Probleme aus. Oder wählen Sie im Widget „OpsItemNach Schweregrad“ den Link „Kritisch“ oder „Hoch“. Exploreröffnet die OpsDataSeite für den ausgewählten Datensatz.
4. Wählen Sie Tabelle exportieren.

   > **Note**
   >
   > Wenn Sie OpsData zum ersten Mal exportieren, erstellt das System eine Rolle für den Export. Sie können die standardmäßig angenommene Rolle nicht ändern.

5. Wählen Sie für S3-Bucket-Name einen bestehenden Bucket. Sie können Erstellen wählen, um einen Amazon-S3-Bucket zu erstellen, falls erforderlich.

   Wenn Sie den Namen des S3-Buckets nicht ändern können, bedeutet dies, dass Sie den Bucket-Namen auf der Seite Einstellungen konfiguriert haben. Sie können den Bucket-Namen nur auf der Seite Einstellungen ändern.

> **Note**
>
> Sie können den Namen des Amazon-S3-Buckets und den Amazon-SNS-Themen-ARN nur auf der Seite bearbeiten, auf der Sie diese Einstellungen zum ersten Mal in Explorer konfiguriert haben.

6. Wählen Sie für SNS-Themen-ARN einen bestehenden Amazon-SNS-Themen-ARN, der benachrichtigt werden soll, wenn der Download abgeschlossen ist.

   Wenn Sie den ARN des Amazon SNS-Themas nicht ändern können, bedeutet dies, dass Sie den ARN des Amazon SNS-Themas auf der Seite Einstellungen konfiguriert haben. Sie können den Themen-ARN nur auf der Seite Einstellungen ändern.
7. (Optional) Geben Sie für SNS-Erfolgsmeldung eine Erfolgsmeldung an, die angezeigt werden soll, wenn der Export erfolgreich abgeschlossen wurde.
8. Wählen Sie Absenden aus. Das System navigiert zur vorherigen Seite und zeigt die Meldung Hier klicken, um den Status des Exportvorgangs anzuzeigen. Details anzeigen.

   Sie können Details anzeigen wählen, um den Status des Runbooks und den Fortschritt in Systems Manager Automation anzuzeigen.

Sie können jetzt OpsData aus dem Explorer angegebenen Amazon S3 S3-Bucket exportieren.

Wenn Sie mit diesem Verfahren keine Daten exportieren können, stellen Sie sicher, dass Ihr Benutzer, Ihre Gruppe oder Rolle die `iam:CreatePolicyVersion`- und `iam:DeletePolicyVersion`-Aktionen einschließt. Weitere Informationen zum Hinzufügen dieser Aktionen zu Ihrem Benutzer, Ihrer Gruppe oder Ihrer Rolle finden Sie unter Bearbeiten von IAM-Richtlinien im IAM-Benutzerhandbuch.

Fehlerbehebung von Systems Manager Explorer

Dieses Thema enthält Informationen zum Beheben gängiger Probleme mit AWS Systems Manager-Explorer.

Die Ressourcendaten-Synchronisierung funktioniert nicht mehr

Wenn Ihre Ressourcendaten-Synchronisierung für eine Opt-in-AWS-Region nicht mehr funktioniert, überprüfen Sie, ob die Region für das AWS Organizations-Verwaltungskonto sowie für das Mitgliedskonto, für das die Synchronisierung konfiguriert ist, ausgewählt wurde.

Die meisten AWS-Regionen sind standardmäßig für Ihr AWS-Konto aktiv, aber bestimmte Regionen werden nur aktiviert, wenn Sie sie manuell auswählen. Diese Regionen werden als Opt-in-Regionen bezeichnet. Standardmäßig unterstützen konten-/regionsübergreifende Ressourcendaten-Synchronisierungen in Explorer keine Datenaggregation in Opt-in-Regionen. Am 30. Juni 2025 wurde Support für die folgenden Opt-In-Regionen hinzugefügt.

- Europa (Milan)
- Afrika (Kapstadt)
- Naher Osten (Bahrain)
- Asien-Pazifik (Hongkong)

Beachten Sie, dass Sie kein delegiertes Administratorkonto verwenden können, um eine Ressourcendatensynchronisierung in Opt-in-Regionen zu erstellen. Sie müssen ein AWS Organizations-Verwaltungskonto verwenden.

Kann AWS-Ressourcen in Explorer nach dem Aktualisieren von Tags auf der Seite Settings (Einstellungen) nicht filtern

Wenn Sie Tag-Schlüssel oder andere Dateneinstellungen im Explorer aktualisieren, kann das System bis zu sechs Stunden benötigen, um Daten basierend auf Ihren Änderungen zu synchronisieren.

Die AWS Organizations-Optionen auf der Seite Create resource data sync (Ressourcendatensynchronisierung erstellen) sind ausgegraut

Die Optionen Include all accounts from my AWS Organizations configuration (Alle Konten von meiner AOlong-Konfiguration einbeziehen) und Select organization units in AWS Organizations(Organisationseinheiten in AOlong auswählen auf der Seite Create resource data sync (Ressourcendatensynchronisierung erstellen) sind nur verfügbar, wenn Sie AWS Organizations eingerichtet und konfiguriert haben. Wenn Sie AWS Organizations eingerichtet und konfiguriert haben, kann das AWS Organizations-Verwaltungskonto oder ein delegierter Explorer-Administrator Ressourcendatensynchronisierungen erstellen, die diese Optionen verwenden.

Weitere Informationen erhalten Sie unter Einrichten von Systems Manager Explorer, um Daten aus mehreren Konten und Regionen anzuzeigen und Konfigurierung eines delegierten Administrators für Explorer.

Explorer zeigt überhaupt keine Daten an

- Stellen Sie sicher, dass Sie das integrierte Setup in jedem Konto und jeder Region abgeschlossen haben, in der Explorer Daten aufrufen und anzeigen soll. Wenn dies nicht der Fall ist, zeigt Explorer keine OpsData und OpsItems für die Konten und Regionen an, in denen Sie das integrierte Setup nicht abgeschlossen haben. Weitere Informationen finden Sie unter Erste Schritte mit Systems Manager Explorer und OpsCenter.
- Wenn Sie Explorer verwenden, um Daten aus mehreren Konten und Regionen anzuzeigen, stellen Sie sicher, dass Sie beim AWS Organizations-Verwaltungskonto oder dem delegierten Explorer-Administratorkonto angemeldet sind. Zum Anzeigen von OpsData und OpsItems von mehreren Konten und Regionen müssen Sie bei diesem Konto angemeldet sein.

Widgets zu Amazon-EC2-Instances zeigen keine Daten an

Wenn Widgets zu Amazon Elastic Compute Cloud (Amazon EC2)-Instances, z. B. Instance count (Anzahl der Instances), Managed Instances (Verwaltete Instances) und Instance by AMI (Instance nach AMI) keine Daten anzeigen, überprüfen Sie Folgendes:

- Stellen Sie sicher, dass Sie mehrere Minuten gewartet haben. OpsData kann einige Minuten benötigen, bis es in Explorer angezeigt wird, nachdem Sie das integrierte Setup abgeschlossen haben.
- Stellen Sie sicher, dass Sie AWS Config Configuration Recorder konfiguriert haben. Explorer verwendet Daten, die von AWS Config Configuration Recorder bereitgestellt werden, um Widgets mit Informationen zu Ihren EC2-Instances zu füllen. Weitere Informationen finden Sie unter Verwalten von Configuration Recorder.
- Stellen Sie sicher, dass die Amazon EC2-OpsData-Quelle auf der Seite Settings (Einstellungen) aktiviert ist. Stellen Sie außerdem sicher, dass mehr als 6 Stunden vergangen sind, seit Sie den Konfigurationsrekorder aktiviert oder seit Sie Änderungen an Ihren Instances vorgenommen haben. Systems Manager kann bis zu sechs Stunden benötigen, um Daten von AWS Config in Explorer EC2-Widgets anzuzeigen, nachdem Sie den Konfigurationsrekorder aktiviert oder Änderungen an Ihren Instance vorgenommen haben.
- Wenn eine Instance angehalten oder beendet wird, beachten Sie, dass Explorer diese Instances nach 24 Stunden nicht mehr anzeigt.
- Stellen Sie sicher, dass Sie sich in der richtigen AWS-Region befinden, in der Sie Ihre Amazon–EC2-Instances konfiguriert haben. Explorer zeigt keine Daten über On-Premises-Instances an.
- Wenn Sie eine Ressourcendatensynchronisierung für mehrere Konten und Regionen konfiguriert haben, stellen Sie sicher, dass Sie beim Organisationsverwaltungskonto oder dem delegierten Explorer-Administratorkonto angemeldet sind.

Das Patch-Widget zeigt keine Daten an

Das Widget Non-compliant instances for patching (Nicht konforme Instances für das Patchen) zeigt nur Daten über Patch-Instances an, die nicht kompatibel sind. Dieses Widget zeigt keine Daten an, wenn Ihre Instances konform sind. Wenn Sie vermuten, dass Sie nicht konforme Instances haben, stellen Sie sicher, dass Sie Systems Manager-Patches eingerichtet und konfiguriert haben, und verwenden Sie AWS Systems Manager Patch Manager, um die Patch-Compliance zu überprüfen. Weitere Informationen finden Sie unter AWS Systems Manager Patch Manager.

Sonstige Probleme

Explorer lässt Sie OpsItems nicht bearbeiten und keine Fehler beheben: Über Konten oder Regionen hinweg angezeigte OpsItems sind schreibgeschützt. Sie können nur über ihr Heimatkonto oder ihre Region aktualisiert und korrigiert werden.

AWS Systems Manager OpsCenter

OpsCenter, ein Tool in AWS Systems Manager, bietet einen zentralen Ort, an dem Betriebsingenieure und IT-Experten betriebliche Arbeitsaufgaben (OpsItems) im Zusammenhang mit AWS Ressourcen einsehen, untersuchen und lösen können. OpsCenterwurde entwickelt, um die durchschnittliche Zeit bis zur Lösung von Problemen zu reduzieren, die sich auf AWS Ressourcen auswirken. OpsCenteraggregiert und standardisiert OpsItems alle Dienste und stellt gleichzeitig kontextbezogene Untersuchungsdaten zu den einzelnen OpsItem verwandten und verwandten OpsItems Ressourcen bereit. OpsCenterstellt außerdem Systems Manager Automation-Runbooks bereit, mit denen Sie Probleme schnell lösen können. Sie können durchsuchbare, benutzerdefinierte Daten für jedes OpsItem angeben. Sie können auch automatisch generierte Zusammenfassungsberichte über OpsItems nach Status und Quelle anzeigen. Um mit OpsCenter zu beginnen, öffnen Sie die Systems-Manager-Konsole. Wählen Sie im Navigationsbereich OpsCenter aus.

OpsCenterist in Amazon EventBridge und Amazon integriert CloudWatch. Das bedeutet, dass Sie diese Dienste so konfigurieren können, dass sie automatisch eine OpsItem Anmeldung erstellen, OpsCenter wenn ein CloudWatch Alarm in den `ALARM` Status wechselt oder wenn ein Ereignis von einem AWS Dienst EventBridge verarbeitet wird, der Ereignisse veröffentlicht. Wenn Sie CloudWatch Alarme und EventBridge Ereignisse so konfigurieren, dass sie automatisch erstellt werden, OpsItems können Sie Probleme mit AWS Ressourcen von einer einzigen Konsole aus schnell diagnostizieren und beheben.

Um Ihnen bei der Diagnose von Problemen zu helfen, OpsItem enthalten sie jeweils kontextrelevante Informationen wie den Namen und die ID der AWS Ressource, die die Probleme generiert hatOpsItem, Alarm- oder Ereignisdetails, den Alarmverlauf und ein Diagramm mit der Alarmzeitleiste.

Für die AWS Ressource werden Informationen OpsCenter von AWS Config, AWS CloudTrail Protokollen und CloudWatch Amazon-Ereignissen zusammengefasst, sodass Sie während Ihrer Untersuchung nicht über mehrere Konsolenseiten navigieren müssen.

Die folgende Liste enthält Ressourcentypen AWS und Metriken, für die Kunden CloudWatch Alarme konfigurieren, die Folgendes auslösenOpsItems:

- Amazon DynamoDB: Lese- und Schreibaktionen in der Datenbank erreichen einen Schwellenwert
- Amazon EC2: Die CPU-Auslastung erreicht einen Schwellenwert
- AWS Abrechnung: Die geschätzten Gebühren erreichen einen Schwellenwert
- Amazon EC2: Eine Instance besteht eine Statusprüfung nicht
- Amazon Elastic Block Store (EBS): Die Festplattenspeichernutzung erreicht einen Schwellenwert

Die folgende Liste enthält EventBridge Regeltypen, für deren Erstellung der Kunde konfiguriert hatOpsItems.

- AWS Security Hub CSPM: Sicherheitswarnung ausgegeben
- DynamoDB: ein Drosselungsereignis
- Amazon EC2 Auto Scaling: Fehler beim Starten einer Instance
- Systems Manager: Fehler beim Ausführen einer Automatisierung
- AWS Health: eine Warnung für geplante Wartungsarbeiten
- EC2: Änderung des Instance-Status von `Running` zu `Stopped`

OpsCenterist auch in Amazon CloudWatch Application Insights für .NET und SQL Server integriert. Das bedeutet, dass Sie automatisch OpsItems für Probleme erstellen können, die in Ihren Anwendungen erkannt wurden. Sie können auch in Systems Manager OpsCenter integrieren AWS Security Hub CSPM , um Ihre Sicherheits-, Leistungs- und Betriebsprobleme zu aggregieren und Maßnahmen zu ergreifen.

Betriebsingenieure und IT-Experten können OpsItems mithilfe der OpsCenter Seite in der AWS Systems Manager Konsole, öffentliche API-Operationen, die AWS Command Line Interface (AWS CLI), oder die Daten erstellen AWS Tools for Windows PowerShell, anzeigen und bearbeiten AWS

SDKs. OpsCenterÖffentliche API-Operationen ermöglichen Ihnen auch die OpsCenter Integration in Ihre Fallmanagementsysteme und Gesundheits-Dashboards.

Welche Vorteile bietet OpsCenter meiner Organisation?

OpsCenterbietet eine standardisierte und einheitliche Oberfläche für die Anzeige, Bearbeitung und Behebung von Problemen im Zusammenhang AWS mit Ressourcen. Eine standardisierte und einheitliche Erfahrung optimiert die Zeit, die benötigt wird, um Probleme zu beheben, verwandte Probleme zu untersuchen und neue Betriebstechniker und IT-Experten zu schulen. Eine standardisierte und einheitliche Erfahrung reduziert ebenfalls die Anzahl der manuellen Fehler, die in das System zur Verwaltung und Behebung von Problemen eingegeben werden.

Genauer gesagt bietet OpsCenter Betriebstechnikern und Unternehmen folgende Vorteile:

- Sie müssen nicht mehr über mehrere Konsolenseiten navigieren, um AWS Ressourcen anzusehen, zu untersuchen und OpsItems zu lösen. OpsItemswerden dienstübergreifend an einem zentralen Ort zusammengefasst.
- Sie können dienstspezifische und kontextrelevante Daten anzeigenOpsItems, die automatisch durch CloudWatch Alarme, EventBridge Ereignisse und CloudWatch Application Insights für .NET und SQL Server generiert werden.
- Sie können den Amazon-Ressourcennamen (ARN) einer Ressource im Zusammenhang mit einer OpsItem festlegen. Durch die Angabe zugehöriger Ressourcen nutztOpsCenter eine integrierte Logik, damit die Erstellung doppelter OpsItems vermieden wird.
- Sie können Detail- und Auflösungsinformationen zu ähnlichen OpsItems einsehen.
- Sie können schnell Informationen zu Systems Manager Automation-Runbooks anzeigen und ausführen, um Probleme zu beheben.

Über welche Features verfügt OpsCenter?

- Automatisierte und manuelle OpsItem-Erstellung

  OpsCenterist in Amazon integriert CloudWatch. Das bedeutet, dass Sie so konfigurieren können CloudWatch , dass automatisch ein OpsItem IN erstellt wird, OpsCenter wenn ein Alarm den `ALARM` Status erreicht oder wenn Amazon ein Ereignis von einem AWS Service EventBridge verarbeitet, der Ereignisse veröffentlicht. Sie können auch manuell OpsItems erstellen.

OpsCenterist auch in Amazon CloudWatch Application Insights für .NET und SQL Server integriert. Das bedeutet, dass Sie automatisch OpsItems für Probleme erstellen können, die in Ihren Anwendungen erkannt wurden.

- Detaillierte und durchsuchbare OpsItems

  Jedes OpsItem enthält mehrere Informationsfelder, darunter einen Titel, eine ID, eine Priorität, eine BeschreibungOpsItem, die Quelle und den date/time Zeitpunkt der letzten Aktualisierung. Jedes OpsItem beinhaltet auch die folgenden konfigurierbaren Funktionen:

  - Status: „Offen“, „In Bearbeitung“, „Abgeschlossen“ oder „Öffnen und In Bearbeitung“.
  - Verwandte Ressourcen: Eine verwandte Ressource ist die betroffene Ressource oder die Ressource, die das Ereignis ausgelöst hat, durch das das EventBridge Ereignis ausgelöst wurde. OpsItem Jedes OpsItem enthält einen Related resources (Zugehörige Ressourcen)-Abschnitt, in dem OpsCenter automatisch die Amazon-Ressourcennamen (ARN) der zugehörigen Ressource auflistet. Sie können verwandte Ressourcen auch manuell angeben ARNs . Bei einigen ARN Typen erstellt OpsCenter automatisch einen Deeplink, der Details über die Ressource anzeigt, ohne dass andere Konsolenseiten zum Einsehen dieser Information aufgerufen werden müssen. Wenn Sie beispielsweise den ARN einer EC2 Instance angeben, können Sie alle von EC2 -bereitgestellten Details zu dieser Instance in OpsCenter anzeigen. Sie können weitere zugehörige Ressourcen manuell hinzufügen. ARNs Jede OpsItem kann maximal 100 zugehörige Ressourcen auflisten ARNs. Weitere Informationen finden Sie unter Hinzufügen zugehöriger Ressourcen zu einem OpsItem.
  - Verwandt und ähnlich OpsItems: Mit der OpsItems Funktion „Verwandt“ können Sie angebenOpsItems, welche in irgendeiner Weise mit der aktuellen Version verwandt sindOpsItem. IDs Die Funktion Similar (ähnliche) OpsItem überprüft automatisch OpsItem-Titel und Beschreibungen und listet anschließend andere OpsItems auf, die möglicherweise im Zusammenhang stehen oder für Sie von Interesse sind.
  - Durchsuchbare und private Betriebsdaten: Betriebsdaten sind benutzerdefinierte Daten, die nützliche Referenzdetails über die OpsItem bereitstellen. Sie können beispielsweise Protokolldateien, Fehlerzeichenfolgen, Lizenzschlüssel, Tipps zur Fehlerbehebung oder andere relevante Daten angeben. Geben Sie Betriebsdaten als Schlüssel-Wert-Paare ein. Der Schlüssel verfügt über eine maximale Länge von 128 Zeichen. Der Wert verfügt über eine maximale Größe von 20 KB.

    Diese benutzerdefinierten Daten sind mit Einschränkungen durchsuchbar. Bezüglich der Funktion Searchable operational data (Durchsuchbare Betriebsdaten) können alle Benutzer

mit Zugriff auf die Seite OpsItem-Übersicht (gemäß BeschreibeOpsItems API-Operation) die angegebenen Daten einsehen oder nach diesen suchen. Bei der Funktion Private Betriebsdaten sind die Daten nur für Benutzer sichtbar, die Zugriff auf die haben OpsItem (wie durch den GetOpsItemAPI-Vorgang bereitgestellt).

- Deduplizierung: Durch die Angabe verwandter Ressourcen verwendet OpsCenter eine integrierte Logik, damit die Erstellung doppelter OpsItems vermieden wird. OpsCenter enthält auch eine Funktion namens Operational insights (Betriebliche Einblick)e, die Informationen über OpsItems-Duplikate anzeigt. Um die Anzahl der Duplikate OpsItems in Ihrem Konto weiter zu begrenzen, können Sie manuell eine Deduplizierungszeichenfolge für eine EventBridge Ereignisregel angeben. Weitere Informationen finden Sie unter Verwalten von OpsItems-Duplikaten.

- Massenbearbeitung von OpsItems: Sie können mehrere OpsItems in OpsCenter auswählen und eines der folgenden Felder bearbeiten: Status, Priority (Priorität), Severity (Schweregrad), Category (Kategorie).
- Einfache Korrekturmaßnahmen mithilfe von Runbooks

  Jedes OpsItem enthält einen Abschnitt Runbooks mit einer Liste von Systems Manager Automation-Runbooks, mit denen Sie häufig auftretende Probleme mit Ressourcen automatisch beheben können. AWS Wenn Sie ein öffnenOpsItem, wählen Sie dafür eine AWS Ressource aus und klicken dann in der Konsole auf die Schaltfläche Automatisierung ausführen. Daraufhin wird OpsCenter eine Liste von Automations-Runbooks angezeigt, die Sie auf der AWS Ressource ausführen können, die das generiert hat. OpsItem OpsItem Nach der Ausführung eines Automation-Runbooks aus einem OpsItem wird das Runbook automatisch der zugehörigen Ressource des OpsItem zur späteren Verwendung zugeordnet. Wenn Sie außerdem automatisch OpsItem Regeln einrichten, EventBridge indem Sie mithilfe vonOpsCenter, dann EventBridge automatisch Runbooks für allgemeine Ereignisse zuordnen. OpsCenterführt eine 30-Tage-Aufzeichnung der Automations-Runbooks, die für ein bestimmtes Ereignis ausgeführt wurden. OpsItem Weitere Informationen finden Sie unter Beheben von OpsItem-Problemen.
- Änderungsbenachrichtigung: Sie können die ARN eines Amazon Simple Notification Service (SNS)-Themas angeben und Benachrichtigungen jederzeit veröffentlichen, sobald ein OpsItem geändert oder bearbeitet wurde. Das SNS-Thema muss genauso existieren wie AWS-Region das. OpsItem
- Umfassende OpsItem-Suchfunktionen: OpsCenter bietet mehrere Suchoptionen, mit denen Sie OpsItems schnell finden können. Im Folgenden finden Sie einige Beispiele dafür, wie Sie eine Suche durchführen können: OpsItem-ID, Titel, Zeitpunkt der letzten Änderung, Betriebsdaten-Wert, Quelle und Automatisierungs-ID einer Runbook-Ausführung, um nur einige zu nennen. Sie können Suchergebnisse weiter einschränken, indem Sie Statusfilter verwenden.

- OpsItem Zusammenfassungsberichte

  OpsCenter enthält eine Zusammenfassungsberichtsseite, die automatisch die folgenden Abschnitte angezeigt:

  - Status summary (Statuszusammenfassung): eine OpsItems-Zusammenfassung nach Status („Offen“, „In Bearbeitung“, „Abgeschlossen“, „Öffnen und In Bearbeitung“).
  - Quellen mit den meisten geöffneten OpsItems Inhalten: eine Aufschlüsselung der am häufigsten geöffneten AWS OpsItems Dienste.
  - OpsItems durch Quelle und Alter: eine Anzahl von OpsItems, gruppiert nach Quelle und Tagen seit der Erstellung.

  Weitere Informationen zum Anzeigen von OpsCenter-Zusammenfassungsberichten finden Sie unter Anzeigen von OpsCenter-Zusammenfassungsberichten.

- Support für Protokollierungs- und Prüfungsfunktionen

  AWS-Konto Durch die Integration mit anderen AWS Diensten können Sie OpsCenter Benutzeraktionen in Ihrem System überprüfen und protokollieren. Weitere Informationen finden Sie unter Anzeigen von OpsCenter-Protokollen und Berichten.

- Konsole PowerShell, CLI und SDK-Zugriff auf OpsCenter das Tool

  Sie können damit arbeiten, OpsCenter indem Sie die AWS Systems Manager Konsole AWS Command Line Interface (AWS CLI) oder das AWS SDK Ihrer Wahl verwenden. AWS -Tools für PowerShell

Lässt sich OpsCenter in mein vorhandenes Fallbearbeitungssystem integrieren?

OpsCenter ist so konzipiert, dass es Ihre vorhandenen Fallbearbeitungssysteme ergänzt. Sie können OpsItems in Ihre bestehendes Fallbearbeitungssystem integrieren, indem Sie öffentliche API-Operationen verwenden. Sie können auch manuelle Lebenszyklus-Workflows in Ihren derzeitigen Systemen pflegen und OpsCenter als Untersuchungs- und Korrektur-Hub verwenden.

Für weitere Informationen zu OpsCenter öffentlichen API-Operationen finden Sie in den folgenden API-Operationen in der AWS Systems Manager API-Referenz.

- CreateOpsItem
- DescribeOpsItems
- GetOpsItem

- GetOpsSummary
- UpdateOpsItem

Entstehen Kosten für die Verwendung von OpsCenter?

Ja. Weitere Informationen finden Sie unter AWS Systems Manager – Preise.

Funktioniert OpsCenter auf meinen On-Premises-Servern und hybriden verwalteten Knoten?

Ja. Sie können OpsCenter verwenden, um Probleme mit Ihren verwalteten On-Premises-Knoten zu untersuchen und zu beheben, die für Systems Manager konfiguriert sind. Weitere Informationen zum Einrichten und Konfigurieren On-Premises-Server und virtueller Computer für Systems Manager finden Sie unter Verwalten von Knoten in Hybrid- und Multi-Cloud-Umgebungen mit Systems Manager.

Was sind die Kontingente für OpsCenter?

Sie können Kontingente für alle Systems-Manager-Tools unter Service Quotas für Systems Manager im Allgemeine Amazon Web Services-Referenz anzeigen. Sofern nicht anders angegeben, gilt jedes Kontingent spezifisch für eine Region.

Einrichten von OpsCenter

AWS Systems Manager verwendet eine integrierte Einrichtungsumgebung, um Ihnen den Einstieg in OpsCenter und Explorer, Tools in Systems Manager, zu erleichtern. Explorer ist ein anpassbares Betriebs-Dashboard, das Informationen über Ihre AWS-Ressourcen meldet. In dieser Dokumentation wird die Einrichtung von Explorer und OpsCenter als Integrierte Einrichtung bezeichnet.

Für die Einrichtung müssen Sie die integrierte Einrichtung verwenden, um OpsCenter mit Explorer einzurichten. Integriertes Setup ist nur in der AWS Systems Manager-Konsole verfügbar. Sie können Explorer und OpsCenter nicht programmatisch einrichten. Weitere Informationen finden Sie unter Erste Schritte mit Systems Manager Explorer und OpsCenter.

Bevor Sie beginnen

Bei der Einrichtung von OpsCenter aktivieren Sie Standardregeln in Amazon EventBridge, die OpsItems automatisch erstellen. Die folgende Tabelle beschreibt die Standard-EventBridge-Regeln, die OpsItems automatisch erstellen. Sie können EventBridge-Regeln im OpsCenter auf der Seite Einstellungen unter OpsItem-Regeln deaktivieren.

> **Important**
>
> Wir stellen Ihnen OpsItems in Rechnung, die mit Standardregeln erstellt werden. Weitere Informationen finden Sie unter AWS Systems Manager – Preise.

Regelname	Beschreibung
SSMOpsItems-Autoscaling-instance-launch-failure	Diese Regel erstellt OpsItems, wenn der Start einer Auto-Scaling-EC2-Instance fehlschlägt.
SSMOpsItems-Autoscaling-instance-termination-failure	Diese Regel erstellt OpsItems, wenn das Beenden einer Auto-Scaling-EC2-Instance fehlschlägt.
SSMOpsItems-EBS-snapshot-copy-failed	Diese Regel erstellt OpsItems, wenn das System einen Snapshot des Amazon Elastic Block Store (Amazon EBS) nicht erfolgreich kopieren konnte.
SSMOpsItems-EBS-snapshot-creation-failed	Diese Regel erstellt OpsItems, wenn das System einen Amazon-EBS-Snapshot nicht erfolgreich erstellen konnte.
SSMOpsItems-EBS-volume-performance-issue	Diese Regel entspricht einer AWS Health-Tracking-Regel. Die Regel erstellt OpsItems, wenn es ein Leistungsproblem mit einem Amazon-EBS-Volume gibt (Zustandsereignis = `AWS_EBS_DEGRADED_EBS_VOLUME_PERFORMANCE`).
SSMOpsItems-EC2-issue	Diese Regel entspricht einer AWS Health-Tracking-Regel für unerwartete Ereignisse, die sich auf AWS-Dienste oder -Ressourcen auswirken. Die Regel erstellt OpsItems, wenn beispielsweise ein Dienst Mitteilungen über betriebliche Probleme sendet, die zu einer Beeinträchtigung des Dienstes führen, oder

Regelname	Beschreibung
	um auf lokale Probleme auf Ressourcenebene aufmerksam zu machen. Diese Regel erstellt beispielsweise ein OpsItem für das folgende Ereignis: `AWS_EC2_OPERATIONAL_ISSUE` .
SSMOpsItems-EC2-scheduled-change	Diese Regel entspricht einer AWS Health-Tracking-Regel. AWS kann Ereignisse für Ihre Instances planen, z. B. Neustart, Stopp oder Start. Die Regel erstellt OpsItems für geplante Ereignisse in EC2. Weitere Informationen zu geplanten Ereignissen finden Sie unter Geplante Ereignisse für Ihre Instances im Benutzerhandbuch zu Amazon EC2.
SSMOpsItems-RDS-issue	Diese Regel entspricht einer AWS Health-Tracking-Regel für unerwartete Ereignisse, die sich auf AWS-Services oder -Ressourcen auswirken. Die Regel erstellt OpsItems, wenn beispielsweise ein Service Mitteilungen über betriebliche Probleme sendet, die zu einer Beeinträchtigung des Dienstes führen, oder um auf lokale Probleme auf Ressourcenebene aufmerksam zu machen. Diese Regel erstellt beispielsweise ein OpsItem für die folgenden Ereignisse: `AWS_RDS_MYSQL_DATABASE_CRASHING_REPEATEDLY` , `AWS_RDS_EXPORT_TASK_FAILED` und `AWS_RDS_CONNECTIVITY_ISSUE` .

Regelname	Beschreibung
SSMOpsItems-RDS-scheduled-change	Diese Regel entspricht einer AWS Health-Tracking-Regel. Die Regel erstellt OpsItems für geplante Ereignisse in Amazon RDS. Geplante Ereignisse bieten Informationen über bevorstehende Änderungen an Ihren Amazon-RDS-Ressourcen. Bei einigen Ereignissen wird Ihnen empfohlen, Maßnahmen zu ergreifen, um Unterbrechungen des Dienstes zu vermeiden. Andere Ereignisse treten automatisch auf, ohne dass Sie etwas tun müssen. Ihre Ressource ist während der geplanten Änderungsaktivität möglicherweise vorübergehend nicht verfügbar. Diese Regel erstellt beispielsweise ein OpsItem für die folgenden Ereignisse: `AWS_RDS_SYSTEM_UPGRADE_SCHEDULED` und `AWS_RDS_MAINTENANCE_SCHEDULED`. Weitere Informationen zu geplanten Ereignissen finden Sie im AWS Health-Benutzerhandbuch unter Ereignistypkategorien.
SSMOpsItems-SSM-maintenance-window-execution-failed	Diese Regel erstellt OpsItems, wenn die Verarbeitung des Systems-Manager-Wartungsfensters fehlschlägt.
SSMOpsItems-SSM-maintenance-window-execution-timedout	Diese Regel erstellt OpsItems, wenn beim Start des Systems-Manager-Wartungsfensters eine Zeitüberschreitung auftritt.

Führen Sie die folgenden Schritte aus, um OpsCenter einzurichten.

So konfigurieren Sie OpsCenter:

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.

3. Wählen Sie auf der OpsCenter-Startseite Erste Schritte aus.
4. Wählen Sie auf der OpsCenter-Einrichtungsseite die Option Diese Option aktivieren, damit Explorer AWS Config konfiguriert und Amazon CloudWatch Events automatisch OpsItems auf der Grundlage häufig verwendeter Regeln und Ereignisse erstellt. Wenn Sie diese Option nicht auswählen, bleibt OpsCenter deaktiviert.

   > **Note**
   >
   > Amazon EventBridge (ehemals Amazon CloudWatch Events) bietet alle Funktionen von CloudWatch Events und einige neue Feautres, wie z. B. benutzerdefinierte Event Buses, Ereignisquellen von Drittanbietern und Schemaregistrierung.

5. Wählen Sie Enable (Aktivieren)OpsCenter aus.

Nachdem Sie OpsCenter aktiviert haben, können Sie Folgendes in den Einstellungen vornehmen:

- Erstellen Sie CloudWatch-Warnungen mit der Schaltfläche CloudWatch-Konsole öffnen. Weitere Informationen finden Sie unter Konfigurieren von CloudWatch-Alarmen zum Erstellen von OpsItems.
- Aktivieren Sie betriebliche Einblicke. Weitere Informationen finden Sie unter Analyse betrieblicher Einblicke zur Reduzierung von OpsItems.
- Aktivieren Sie AWS Security Hub CSPM, um Warnungen zu finden. Weitere Informationen finden Sie unter Verständnis der OpsCenter Integration mit AWS Security Hub CSPM.

Inhalt

- (Optional) Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems
- (Optional) Einrichten von Amazon SNS für den Empfang von Benachrichtigungen zu OpsItems

(Optional) Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems

Sie können Systems Manager verwendenOpsCenter, um mehrere AWS-Konten in einem ausgewählten System zentral zu verwalten OpsItems AWS-Region. Diese Funktion ist verfügbar, nachdem Sie Ihre Organisation in eingerichtet haben AWS Organizations. AWS Organizations ist ein Kontoverwaltungsdienst, mit dem Sie mehrere AWS Konten in einer Organisation zusammenfassen können, die Sie erstellen und zentral verwalten. AWS Organizations umfasst Funktionen zur

Kontoverwaltung und konsolidierten Fakturierung, mit denen Sie die Haushalts-, Sicherheits- und Compliance-Anforderungen Ihres Unternehmens besser erfüllen können. Weitere Informationen finden Sie unter Was ist AWS Organizations? im AWS Organizations Benutzerhandbuch

Benutzer, die dem AWS Organizations Verwaltungskonto angehören, können ein delegiertes Administratorkonto für Systems Manager einrichten. Im Kontext von OpsCenter können delegierte Administratoren OpsItems in Mitgliederkonten erstellen, bearbeiten und anzeigen. Der delegierte Administrator kann auch Systems Manager Automation-Runbooks verwenden, um Probleme mit generierten AWS Ressourcen gemeinsam zu lösen OpsItems oder zu beheben. OpsItems

> **Note**
>
> Sie können nur ein Konto als delegierten Administrator für Systems Manager zuweisen. Weitere Informationen finden Sie unter Einen AWS Organizations delegierten Administrator für Systems Manager erstellen.

Systems Manager bietet die folgenden Methoden für die Einrichtung von OpsCenter, um OpsItems zentral über mehrere AWS-Konten zu verwalten.

- Quick Setup: Quick Setup, ein Tool in Systems Manager, vereinfacht die Einrichtungs- und Konfigurationsaufgaben für Systems-Manager-Tools. Weitere Informationen finden Sie unter AWS Systems Manager Quick Setup.

  Quick Setup für OpsCenter hilft Ihnen bei der Ausführung der folgenden Aufgaben für die kontenübergreifende Verwaltung von OpsItems:

  - Ein Konto als delegierter Administrator registrieren (wenn der delegierte Administrator nicht bereits benannt wurde)
  - Erstellung der erforderlichen AWS Identity and Access Management (IAM-) Richtlinien und Rollen
  - Geben Sie eine AWS Organizations Organisation oder Organisationseinheiten (OUs) an, in denen ein delegierter Administrator die Verwaltung OpsItems kontenübergreifend durchführen kann

  Weitere Informationen finden Sie unter (Optional) Konfigurieren Sie OpsCenter für die kontenübergreifende Verwaltung von OpsItems mithilfe von Quick Setup.

> **Note**
>
> Quick Setup ist nicht überall verfügbar AWS-Regionen , wo Systems Manager derzeit verfügbar ist. Wenn Quick Setup in einer Region, in der Sie OpsCenter für die zentrale Verwaltung von OpsItems über mehrere Konten hinweg konfigurieren möchten, nicht verfügbar ist, müssen Sie die manuelle Methode verwenden. Eine Liste der verfügbaren AWS-Regionen Quick Setup finden Sie unterVerfügbarkeit von Quick Setup in AWS-Regionen.

- Manuelle Einrichtung: Wenn Quick Setup in der Region, in der Sie OpsCenter für die zentrale Verwaltung von OpsItems über Konten hinweg konfigurieren möchten, nicht verfügbar ist, können Sie dafür den manuellen Vorgang verwenden. Weitere Informationen finden Sie unter (Optional) Manuelle Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems.

(Optional) Konfigurieren Sie OpsCenter für die kontenübergreifende Verwaltung von OpsItems mithilfe von Quick Setup

Quick Setup, ein Tool in AWS Systems Manager, vereinfacht die Einrichtungs- und Konfigurationsaufgaben für Systems Manager Manager-Tools. Quick Setupfor OpsCenter unterstützt Sie bei der Ausführung der folgenden Aufgaben für die OpsItems kontenübergreifende Verwaltung:

- Angeben des delegierten Administratorkontos
- Erstellung der erforderlichen AWS Identity and Access Management (IAM-) Richtlinien und Rollen
- Angabe einer AWS Organizations Organisation oder einer Teilmenge von Mitgliedskonten, die ein delegierter Administrator kontenübergreifend verwalten kann OpsItems

Wenn Sie OpsCenter für die kontenübergreifende Verwaltung von OpsItems mithilfe von Quick Setup konfigurieren, erstellt Quick Setup die folgenden Ressourcen in den angegebenen Konten. Diese Ressourcen gewähren den angegebenen Konten die Erlaubnis, mit Automation-Runbooks zu arbeiten OpsItems und diese zu verwenden, um Probleme bei AWS der Generierung von Ressourcen zu beheben. OpsItems

Ressourcen	Konten
`AWSServiceRoleForAmazonSSM_AccountDiscovery` AWS Identity and Access Management (IAM) serviceverknüpfte Rolle Weitere Informationen über diese Rolle finden Sie unter Verwenden von Rollen zum Sammeln von AWS-Konto-Informationen für OpsCenter und Explorer.	AWS Organizations Verwaltungskonto und delegiertes Administratorkonto
`OpsItem-CrossAccountManagementRole` -IAM-Rolle `AWS-SystemsManager-AutomationAdministrationRole` -IAM-Rolle	Delegiertes Administratorkonto
`OpsItem-CrossAccountExecutionRole` -IAM-Rolle `AWS-SystemsManager-AutomationExecutionRole` -IAM-Rolle `AWS::SSM::ResourcePolicy` Systems-Manage-Ressourcenrichtlinie für die standardmäßige OpsItem-Gruppe (`OpsItemGroup`)	Alle AWS Organizations Mitgliedskonten

> **Note**
>
> Wenn Sie zuvor für die manuelle Verwaltung OpsItems von Konten konfiguriert OpsCenter haben, müssen Sie die in den Schritten 4 und 5 dieses Vorgangs erstellten AWS CloudFormation Stapel oder Stack-Sets löschen. Wenn diese Ressourcen in Ihrem Konto vorhanden sind, wenn Sie das folgende Verfahren ausführen, kann Quick Setup die kontenübergreifende Verwaltung von OpsItem nicht ordnungsgemäß konfigurieren.

So konfigurieren Sie OpsCenter, um OpsItems kontenübergreifend mithilfe von Quick Setup zu verwalten

1. Melden Sie sich AWS-Managementkonsole mit dem AWS Organizations Verwaltungskonto bei an.
2. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
3. Wählen Sie im Navigationsbereich Quick Setup aus.
4. Wählen Sie die Registerkarte Bibliothek.
5. Scrollen Sie nach unten und finden Sie die OpsCenter-Konfigurationskachel. Wählen Sie Erstellen aus.
6. Geben Sie auf der Seite Quick Setup OpsCenter im Bereich Delegierter Administrator eine Konto-ID ein. Wenn Sie dieses Feld nicht bearbeiten können, wurde bereits ein delegiertes Administratorkonto für Systems Manager angegeben.
7. Wählen Sie im Abschnitt Targets (Ziele) eine Option aus. Wenn Sie Benutzerdefiniert wählen, wählen Sie die Organisationseinheiten (OU) aus, die OpsItems kontenübergreifend verwalten sollen.
8. Wählen Sie Erstellen aus.

Quick Setuperstellt die OpsCenter Konfiguration und stellt die erforderlichen AWS Ressourcen für die angegebenen OUs Ressourcen bereit.

> **Note**
>
> Wenn Sie OpsItems nicht über mehrere Konten hinweg verwalten möchten, können Sie die Konfiguration von Quick Setup löschen. Wenn Sie die Konfiguration löschen, löscht Quick Setup die folgenden IAM-Richtlinien und -Rollen, die bei der ursprünglichen Bereitstellung der Konfiguration erstellt wurden:
>
> - `OpsItem-CrossAccountManagementRole` aus dem delegierten Administratorkonto
> - `OpsItem-CrossAccountExecutionRole` und `SSM::ResourcePolicy` aus allen Organizations-Mitgliedskonten
>
> Quick Setup entfernt die Konfiguration aus allen Organisationseinheiten und AWS-Regionen , in denen die Konfiguration ursprünglich bereitgestellt wurde.

Behebung von Problemen mit einer Quick Setup-Konfiguration für OpsCenter

Dieser Abschnitt enthält Informationen zur Behebung von Problemen bei der Konfiguration der kontenübergreifenden OpsItem-Verwaltung mithilfe von Quick Setup.

Themen

- Die Bereitstellung für folgende Geräte ist StackSets fehlgeschlagen: DelegatedAdmin
- Quick Setup-Konfigurationsstatus zeigt Fehlgeschlagen

Die Bereitstellung für folgende Geräte ist StackSets fehlgeschlagen: DelegatedAdmin

Wenn Sie eine OpsCenter-Konfiguration erstellen, stellt Quick Setup zwei AWS CloudFormation -Stack-Sets im Verwaltungskonto Organizations bereit. Die Stack-Sets verwenden das folgende Präfix: `AWS-QuickSetup-SSMOpsCenter`. Wenn Quick Setup den folgenden Fehler anzeigt: `Deployment to these StackSets failed: delegatedAdmin`, gehen Sie wie folgt vor, um dieses Problem zu beheben.

Um einen failed:DelegatedAdmin-Fehler StackSets zu beheben

1. Wenn Sie den `Deployment to these StackSets failed: delegatedAdmin` Fehler in einem roten Banner in der Quick Setup Konsole erhalten haben, melden Sie sich mit dem delegierten Administratorkonto und der AWS-Region als Heimatregion angegebenen Region an. Quick Setup
2. Öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
3. Wählen Sie den Stack aus, der von der Quick Setup-Konfiguration erstellt wurde. Der Stack-Name beinhaltet Folgendes: AWS- QuickSetup - SSMOps Center.

   > **Note**
   >
   > CloudFormation Löscht manchmal fehlgeschlagene Stack-Bereitstellungen. Wenn der Stack in der Tabelle Stacks nicht verfügbar ist, wählen Sie Gelöscht in der Filterliste aus.

4. Zeigen Sie den Status und den Statusgrund an. Weitere Informationen zum Stack-Status finden Sie unter Stack-Statuscodes im Benutzerhandbuch von AWS CloudFormation .
5. Um nachzuvollziehen, welcher Schritt genau fehlgeschlagen ist, sehen Sie sich auf der Registerkarte Ereignisse den Status der einzelnen Ereignisse an. Weitere Informationen finden Sie unter Fehlerbehebung im AWS CloudFormation -Benutzerhandbuch.

> **Note**
>
> Wenn Sie den Bereitstellungsfehler nicht mithilfe der Schritte CloudFormation zur Fehlerbehebung beheben können, löschen Sie die Konfiguration und versuchen Sie es erneut.

Quick Setup-Konfigurationsstatus zeigt Fehlgeschlagen

Wenn in der Tabelle mit den Konfigurationsdetails auf der Seite mit den Konfigurationsdetails der Konfigurationsstatus angezeigt wird`Failed`, melden Sie sich in der AWS-Konto Region an, in der der Fehler aufgetreten ist.

Einen Quick Setup-Fehler beim Erstellen einer OpsCenter-Konfiguration beheben

1. Melden Sie sich bei der AWS-Konto und in der Region AWS-Region an, in der der Fehler aufgetreten ist.
2. Öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
3. Wählen Sie den Stack aus, der von der Quick Setup-Konfiguration erstellt wurde. Der Stack-Name beinhaltet Folgendes: AWS- QuickSetup - SSMOps Center.

   > **Note**
   >
   > CloudFormation Löscht manchmal fehlgeschlagene Stack-Bereitstellungen. Wenn der Stack in der Tabelle Stacks nicht verfügbar ist, wählen Sie Gelöscht in der Filterliste aus.

4. Zeigen Sie den Status und den Statusgrund an. Weitere Informationen zum Stack-Status finden Sie unter Stack-Statuscodes im Benutzerhandbuch von AWS CloudFormation .
5. Um nachzuvollziehen, welcher Schritt genau fehlgeschlagen ist, sehen Sie sich auf der Registerkarte Ereignisse den Status der einzelnen Ereignisse an. Weitere Informationen finden Sie unter Fehlerbehebung im AWS CloudFormation -Benutzerhandbuch.

Die Konfiguration des Mitgliedskontos zeigt ResourcePolicyLimitExceededException

Wenn der Stack-Status `ResourcePolicyLimitExceededException` anzeigt, wurde das Konto zuvor über die manuelle Methode in die kontoübergreifende OpsCenter-Verwaltung aufgenommen. Um dieses Problem zu beheben, müssen Sie die AWS CloudFormation Stacks oder Stack-Sets

löschen, die Sie in den Schritten 4 und 5 des manuellen Onboarding-Prozesses erstellt haben. Weitere Informationen finden Sie unter Löschen eines Stack-Sets und Löschen eines Stacks auf der CloudFormation Konsole im AWS CloudFormation Benutzerhandbuch.

(Optional) Manuelle Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems

In diesem Abschnitt wird beschrieben, wie Sie OpsCenter manuell für die kontoübergreifende Verwaltung von OpsItem konfigurieren. Dieser Prozess wird zwar weiterhin unterstützt, wurde jedoch durch einen neueren Prozess ersetzt, der Systems Manager Quick Setup verwendet. Weitere Informationen finden Sie unter (Optional) Konfigurieren Sie OpsCenter für die kontenübergreifende Verwaltung von OpsItems mithilfe von Quick Setup.

Sie können ein zentrales Konto einrichten, um manuelle OpsItems für Mitgliedskonten zu erstellen und diese OpsItems zu verwalten und zu beheben. Das zentrale Konto kann das AWS Organizations Verwaltungskonto oder sowohl das AWS Organizations Verwaltungskonto als auch das delegierte Administratorkonto von Systems Manager sein. Wir empfehlen, dass Sie das delegierte Administratorkonto von Systems Manager als zentrales Konto verwenden. Sie können dieses Feature erst verwenden, nachdem Sie AWS Organizations konfiguriert haben.

Mit AWS Organizations können Sie mehrere zu einer Organisation AWS-Konten zusammenfassen, die Sie zentral erstellen und verwalten. Der Benutzer des zentralen Kontos kann OpsItems für alle ausgewählten Mitgliedskonten gleichzeitig Konten erstellen und diese OpsItems verwalten.

Verwenden Sie den Prozess in diesem Abschnitt, um den Systems Manager Manager-Dienstprinzipal in Organizations zu aktivieren und AWS Identity and Access Management (IAM) -Berechtigungen für die OpsItems kontenübergreifende Arbeit zu konfigurieren.

Themen

- Bevor Sie beginnen
- Schritt 1: Erstellen einer Ressourcen-Datensynchronisierung
- Schritt 2: Aktivieren des Systems Manager Manager-Dienstprinzipals in AWS Organizations
- Schritt 3: Erstellen der AWSServiceRoleForAmazonSSM_AccountDiscovery-serviceverknüpften Rolle
- Schritt 4: Konfigurieren von Berechtigungen für die kontenübergreifende Arbeit mit OpsItems
- Schritt 5: Konfigurieren von Berechtigungen für das kontenübergreifende Arbeiten mit zugehörigen Ressourcen

> **Note**
>
> Nur OpsItems vom Typ `/aws/issue` werden bei der kontenübergreifenden Arbeit in OpsCenter unterstützt.

Bevor Sie beginnen

Bevor Sie OpsCenter für die Arbeit mit OpsItems kontenübergreifend einrichten, stellen Sie sicher, dass Sie Folgendes eingerichtet haben:

- Ein delegiertes Administratorkonto für Systems Manager. Weitere Informationen finden Sie unter Konfigurierung eines delegierten Administrators für Explorer.
- Eine Organisation, die in Organizations eingerichtet und konfiguriert wurde. Weitere Informationen finden Sie unter Erstellen und Verwalten einer Organisation im AWS Organizations - Benutzerhandbuch.
- Sie haben Systems Manager Automation so konfiguriert, dass Automatisierungs-Runbooks für mehrere AWS-Regionen AWS Konten ausgeführt werden. Weitere Informationen finden Sie unter Automatisierungen in mehreren Konten AWS-Regionen ausführen.

Schritt 1: Erstellen einer Ressourcen-Datensynchronisierung

Nach der Einrichtung und Konfiguration können Sie die Daten OpsCenter für eine gesamte Organisation aggregieren AWS OrganizationsOpsItems, indem Sie eine Ressourcendatensynchronisierung erstellen. Weitere Informationen finden Sie unter Erstellen einer Ressourcendatensynchronisierung. Achten Sie beim Erstellen der Synchronisierung darauf, im Abschnitt Konten hinzufügen die Option Alle Konten aus meiner AWS Organizations -Konfiguration einbeziehen auszuwählen.

Schritt 2: Aktivieren des Systems Manager Manager-Dienstprinzipals in AWS Organizations

Damit ein Benutzer OpsItems kontenübergreifend arbeiten kann, muss der Systems Manager Manager-Dienstprinzipal aktiviert sein AWS Organizations. Wenn Sie Systems Manager zuvor mit anderen Tools für Szenarios mit mehreren Konten konfiguriert haben, ist der Systems-Manager-Service-Prinzipal möglicherweise bereits in Organizations konfiguriert. Führen Sie zur Überprüfung die folgenden Befehle von AWS Command Line Interface (AWS CLI) aus. Wenn Sie Systems Manager nicht für andere Szenarien mit mehreren Konten konfiguriert haben, fahren Sie mit

dem nächsten Schritt fort: So aktivieren Sie den Systems-Manager-Service-Prinzipal in AWS Organizations.

So überprüfen Sie, ob der Systems Manager Manager-Dienstprinzipal aktiviert ist in AWS Organizations

1. Laden Sie die neueste Version von AWS CLI auf Ihren lokalen Computer herunter.
2. Öffnen Sie den AWS CLI und führen Sie den folgenden Befehl aus, um Ihre Anmeldeinformationen und eine anzugeben AWS-Region.

   ```
   aws configure
   ```

   Sie werden aufgefordert, Folgendes anzugeben: Ersetzen Sie im folgenden Beispiel jede *user input placeholder* durch Ihre eigenen Informationen.

   ```
   AWS Access Key ID [None]: key_name
   AWS Secret Access Key [None]: key_name
   Default region name [None]: region
   Default output format [None]: ENTER
   ```

3. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Systems-Manager-Service-Prinzipal für AWS Organizations aktiviert ist.

   ```
   aws organizations list-aws-service-access-for-organization
   ```

   Der Befehl gibt ähnliche Informationen wie im folgenden Beispiel zurück.

   ```
   {
       "EnabledServicePrincipals": [
           {
               "ServicePrincipal":
    "member.org.stacksets.cloudformation.amazonaws.com",
               "DateEnabled": "2020-12-11T16:32:27.732000-08:00"
           },
           {
               "ServicePrincipal": "opsdatasync.ssm.amazonaws.com",
               "DateEnabled": "2022-01-19T12:30:48.352000-08:00"
           },
           {
               "ServicePrincipal": "ssm.amazonaws.com",
   ```

```
            "DateEnabled": "2020-12-11T16:32:26.599000-08:00"
        }
    ]
}
```

So aktivieren Sie den Systems Manager Manager-Dienstprinzipal in AWS Organizations

Wenn Sie den Systems-Manager-Service-Prinzipal für Organizations noch nicht konfiguriert haben, verwenden Sie dazu das folgende Verfahren. Weitere Informationen zu diesem Befehl finden Sie enable-aws-service-accessin der AWS CLI Befehlsreferenz.

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben. Informationen finden Sie unter Installation der CLI und Konfiguration der CLI.
2. Laden Sie die neueste Version von AWS CLI auf Ihren lokalen Computer herunter.
3. Öffnen Sie den AWS CLI und führen Sie den folgenden Befehl aus, um Ihre Anmeldeinformationen und eine anzugeben AWS-Region.

   ```
   aws configure
   ```

   Sie werden aufgefordert, Folgendes anzugeben: Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch Ihre eigenen Informationen.

   ```
   AWS Access Key ID [None]: key_name
   AWS Secret Access Key [None]: key_name
   Default region name [None]: region
   Default output format [None]: ENTER
   ```

4. Führen Sie den folgenden Befehl aus, um den Systems-Manager-Service-Prinzipal für AWS Organizations zu aktivieren.

   ```
   aws organizations enable-aws-service-access --service-principal "ssm.amazonaws.com"
   ```

Schritt 3: Erstellen der **`AWSServiceRoleForAmazonSSM_AccountDiscovery`**-serviceverknüpften Rolle

Eine dienstbezogene Rolle wie die `AWSServiceRoleForAmazonSSM_AccountDiscovery` Rolle ist ein einzigartiger Typ von IAM-Rolle, die direkt mit einer verknüpft ist AWS-Service, z. B. Systems Manager. Mit Diensten verknüpfte Rollen sind vom Dienst

vordefiniert und enthalten alle Berechtigungen, die der Dienst benötigt, um andere AWS-Services in Ihrem Namen anzurufen. Weitere Informationen zur serviceverknüpften `AWSServiceRoleForAmazonSSM_AccountDiscovery`-Rolle finden Sie unter Berechtigungen von serviceverknüpften Rollen für Systems Manager-Kontoerkennung.

Verwenden Sie das folgende Verfahren, um die serviceverknüpfte `AWSServiceRoleForAmazonSSM_AccountDiscovery`-Rolle mithilfe der AWS CLI zu erstellen. Weitere Informationen zu dem in diesem Verfahren verwendeten Befehl finden Sie create-service-linked-rolein der AWS CLI Befehlsreferenz.

So erstellen Sie die serviceverknüpfte **`AWSServiceRoleForAmazonSSM_AccountDiscovery`**-Rolle

1. Melden Sie sich beim AWS Organizations Verwaltungskonto an.
2. Führen Sie den folgenden Befehl aus, während Sie mit dem Organizations-Verwaltungskonto angemeldet sind.

```
aws iam create-service-linked-role \
    --aws-service-name accountdiscovery.ssm.amazonaws.com \
    --description "Systems Manager account discovery for AWS Organizations service-linked role"
```

Schritt 4: Konfigurieren von Berechtigungen für die kontenübergreifende Arbeit mit OpsItems

Verwenden Sie AWS CloudFormation Stacksets, um eine `OpsItemGroup` Ressourcenrichtlinie und eine IAM-Ausführungsrolle zu erstellen, mit OpsItems denen Benutzer kontenübergreifend arbeiten können. Laden Sie zunächst die `OpsCenterCrossAccountMembers.zip`-Datei herunter und entpacken Sie sie. Diese Datei enthält die Vorlagendatei. `OpsCenterCrossAccountMembers.yaml` CloudFormation Wenn Sie mithilfe dieser Vorlage ein Stack-Set erstellen, CloudFormation werden automatisch die `OpsItemCrossAccountResourcePolicy` Ressourcenrichtlinie und die `OpsItemCrossAccountExecutionRole` Ausführungsrolle im Konto erstellt. Weitere Informationen zum Erstellen eines Stack-Sets finden Sie unter Erstellen eines Stack-Sets im AWS CloudFormation - Benutzerhandbuch.

> **Important**
>
> Berücksichtigen Sie für diese Aufgabe die folgenden wichtigen Informationen:

- Sie müssen das Stackset bereitstellen, während Sie beim AWS Organizations - Verwaltungskonto angemeldet sind.
- Sie müssen dieses Verfahren wiederholen, während Sie bei jedem Konto angemeldet sind, das Sie für die Arbeit mit OpsItems kontenübergreifend festlegen möchten, einschließlich des delegierten Administratorkontos.
- Wenn Sie die kontoübergreifende OpsItems Verwaltung in verschiedenen Bereichen aktivieren möchten AWS-Regionen, wählen Sie im Abschnitt Regionen angeben der Vorlage die Option Alle Regionen hinzufügen aus. Die kontoübergreifende OpsItem-Verwaltung wird für Opt-in-Regionen nicht unterstützt.

Schritt 5: Konfigurieren von Berechtigungen für das kontenübergreifende Arbeiten mit zugehörigen Ressourcen

Und OpsItem kann detaillierte Informationen zu betroffenen Ressourcen wie Amazon Elastic Compute Cloud (Amazon EC2) -Instances oder Amazon Simple Storage Service (Amazon S3) -Buckets enthalten. Die `OpsItemCrossAccountExecutionRole`-Ausführungsrolle, die Sie im vorherigen Schritt 4 erstellt haben, stellt OpsCenter mit schreibgeschützten Berechtigungen für Mitgliedskonten zum Anzeigen zugehöriger Ressourcen bereit. Sie müssen auch eine IAM-Rolle erstellen, um Verwaltungskonten die Berechtigung zum Anzeigen und Interagieren mit verwandten Ressourcen zu gewähren. Dies werden Sie in dieser Aufgabe durchführen.

Laden Sie zunächst die `OpsCenterCrossAccountManagementRole.zip`-Datei herunter und entpacken Sie sie. Diese Datei enthält die `OpsCenterCrossAccountManagementRole.yaml` CloudFormation Vorlagendatei. Wenn Sie mithilfe dieser Vorlage einen Stack erstellen, CloudFormation wird automatisch die `OpsCenterCrossAccountManagementRole` IAM-Rolle im Konto erstellt. Weitere Informationen zum Erstellen eines Stacks finden Sie im AWS CloudFormation Benutzerhandbuch unter Erstellen eines Stacks auf der AWS CloudFormation Konsole.

Important

Berücksichtigen Sie für diese Aufgabe die folgenden wichtigen Informationen:

- Wenn Sie beabsichtigen, ein Konto als delegierter Administrator für anzugebenOpsCenter, müssen Sie dies AWS-Konto bei der Erstellung des Stacks angeben.

- Sie müssen dieses Verfahren ausführen, während Sie beim AWS Organizations -Verwaltungskonto angemeldet sind, und erneut, während Sie beim delegierten Administratorkonto angemeldet sind.

(Optional) Einrichten von Amazon SNS für den Empfang von Benachrichtigungen zu OpsItems

Sie können OpsCenter so konfigurieren, dass Benachrichtigungen an ein Amazon Simple Notification Service (Amazon SNS)-Thema gesendet werden, wenn das System eine neues OpsItem erstellt oder ein vorhandenes OpsItem aktualisiert.

Führen Sie die folgenden Schritte aus, um Benachrichtigungen für OpsItems zu erhalten.

- Schritt 1: Erstellen und Abonnieren eines Amazon-SNS-Themas
- Schritt 2: Aktualisieren der Amazon SNS-Zugriffsrichtlinie
- Schritt 3: Aktualisieren der AWS KMS -Zugriffsrichtlinie

> **Note**
>
> Wenn Sie in Schritt 2 die serverseitige Verschlüsselung AWS Key Management Service (AWS KMS) aktivieren, müssen Sie Schritt 3 abschließen. Andernfalls können Sie Schritt 3 überspringen.

- Schritt 4: Aktivieren Sie OpsItems-Standardregeln zum Senden von Benachrichtigungen für neue OpsItems

Schritt 1: Erstellen und Abonnieren eines Amazon-SNS-Themas

Um Benachrichtigungen zu erhalten, müssen Sie ein Amazon SNS-Thema erstellen und abonnieren. Weitere Informationen finden Sie unter Erstellen eines Amazon-SNS-Themas und Abonnieren eines Amazon-SNS-Themas im Entwicklerhandbuch für Amazon Simple Notification Service.

> **Note**
>
> Wenn Sie mehrere Konten verwenden, müssen Sie OpsCenter in jeder Region AWS-Regionen oder jedem Konto, für das Sie OpsItem Benachrichtigungen erhalten möchten, ein Amazon SNS SNS-Thema erstellen und abonnieren.

Schritt 2: Aktualisieren der Amazon SNS-Zugriffsrichtlinie

Sie müssen ein Amazon-SNS-Thema mit OpsItems verknüpfen. Gehen Sie wie folgt vor, um eine Amazon-SNS-Zugriffsrichtlinie einzurichten, damit OpsItems-Benachrichtigungen von Systems Manager für das Amazon-SNS-Thema veröffentlichen kann, das Sie in Schritt 1 erstellt haben.

1. Melden Sie sich bei https://console.aws.amazon.com/sns/v3/home an AWS-Managementkonsole und öffnen Sie die Amazon SNS SNS-Konsole.
2. Wählen Sie im Navigationsbereich Themen aus.
3. Wählen Sie das Thema aus, das Sie in Schritt 1 erstellt haben, und klicken Sie dann auf Bearbeiten.
4. Erweitern Sie die Option Zugriffsrichtlinie.
5. Fügen Sie der vorhandenen Richtlinie den folgenden `Sid`-Block hinzu. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen.

```
{
      "Sid": "Allow OpsCenter to publish to this topic",
      "Effect": "Allow",
      "Principal": {
        "Service": "ssm.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:region:account ID:topic name", // Account ID of the
 SNS topic owner
      "Condition": {
      "StringEquals": {
        "AWS:SourceAccount": "account ID" //  Account ID of the OpsItem owner
      }
   }
}
```

> **Note**
>
> Der `aws:SourceAccount` globale Bedingungsschlüssel schützt vor dem Szenario Confused Deputy. Um diesen Bedingungsschlüssel zu verwenden, setzen Sie den Wert auf die Konto-ID des OpsItem-Eigentümers. Weitere Informationen finden Sie unter Confused Deputy im IAM-Benutzerhandbuch.

6. Wählen Sie Änderungen speichern aus.

Das System sendet jetzt Benachrichtigungen an das Amazon SNS-Thema, wenn OpsItems erstellt oder aktualisiert werden.

> **Important**
>
> Wenn Sie das Amazon SNS SNS-Thema in Schritt 2 mit einem AWS Key Management Service (AWS KMS) serverseitigen Verschlüsselungsschlüssel konfigurieren, führen Sie Schritt 3 aus. Andernfalls können Sie Schritt 3 überspringen.

Schritt 3: Aktualisieren der AWS KMS -Zugriffsrichtlinie

Wenn Sie die AWS KMS serverseitige Verschlüsselung für Ihr Amazon SNS SNS-Thema aktiviert haben, müssen Sie auch die Zugriffsrichtlinie aktualisieren AWS KMS key , die Sie bei der Konfiguration des Themas ausgewählt haben. Gehen Sie wie folgt vor, um die Zugriffsrichtlinie zu aktualisieren, damit Systems Manager OpsItem-Benachrichtigungen für das Amazon-SNS-Thema veröffentlichen kann, das Sie in Schritt 1 erstellt haben.

> **Note**
>
> OpsCenter unterstützt keine Veröffentlichung von OpsItems in einem Amazon-SNS-Thema, das mit einem Von AWS verwalteter Schlüssel konfiguriert ist.

1. Öffnen Sie die AWS KMS Konsole unter /kms. https://console.aws.amazon.com
2. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
4. Wählen Sie die ID des KMS-Schlüssels aus, den Sie bei der Erstellung des Themas ausgewählt haben.
5. Wählen Sie im Abschnitt Key policy (Schlüsselrichtlinie) die Option Switch to policy view (Zur Richtlinienansicht wechseln) aus.
6. Wählen Sie Bearbeiten aus.
7. Fügen Sie der vorhandenen Richtlinie den folgenden `Sid`-Block hinzu. Ersetzen Sie jeden *`example resource placeholder`* durch Ihre Informationen.

```
{
```

```
    "Sid": "Allow OpsItems to decrypt the key",
    "Effect": "Allow",
    "Principal": {
      "Service": "ssm.amazonaws.com"
    },
    "Action": ["kms:Decrypt", "kms:GenerateDataKey*"],
     "Resource": "arn:aws:kms:region:account ID:key/key ID"
  }
```

Im folgenden Beispiel wird der neue Block in Zeile 14 eingegeben.

8. Wählen Sie Änderungen speichern aus.

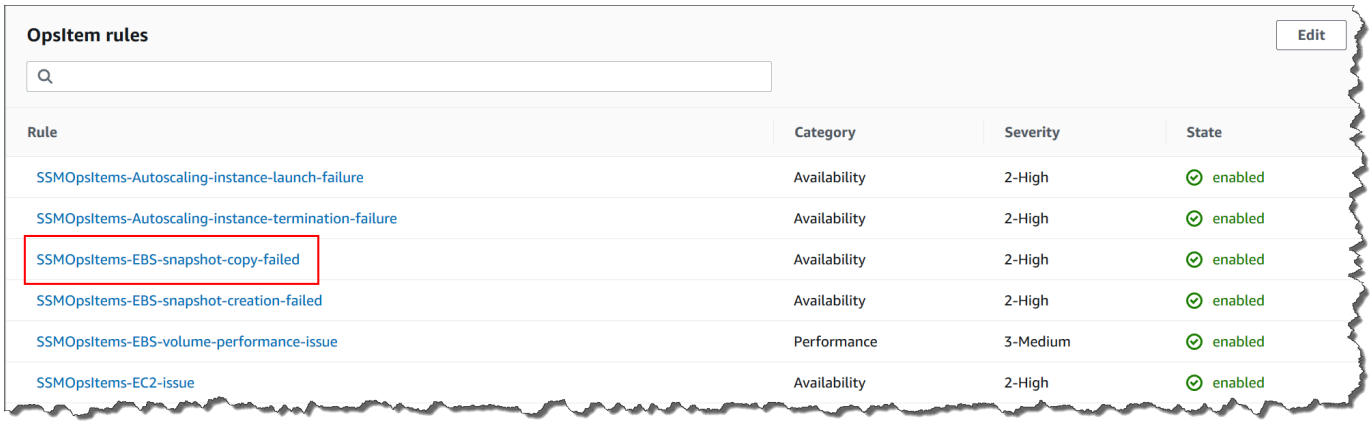
Schritt 4: Aktivieren Sie OpsItems-Standardregeln zum Senden von Benachrichtigungen für neue OpsItems

OpsItemsStandardregeln in Amazon EventBridge sind nicht mit einem Amazon-Ressourcennamen (ARN) für Amazon SNS-Benachrichtigungen konfiguriert. Gehen Sie wie folgt vor, um eine Regel in EventBridge zu bearbeiten und einen `notifications`-Block einzugeben.

So fügen Sie einer OpsItem-Standardregel einen Benachrichtigungsblock hinzu

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie die Registerkarte OpsItems und dann Configure sources (Quellen konfigurieren) aus.

4. Wählen Sie den Namen der Quellregel aus, die Sie mit einem `notifications`-Block konfigurieren möchten, wie im folgenden Beispiel gezeigt.



   Die Regel wird in Amazon geöffnet EventBridge.

5. Wählen Sie auf der Regeldetailseite auf der Registerkarte Targets (Ziele) die Option Edit (Bearbeiten) aus.
6. Wählen Sie im Bereich Additional settings (Zusätzliche Einstellungen) die Option Configure input transformer (Eingabetransformator konfigurieren).
7. Fügen Sie im Feld Vorlage einen `notifications`-Block im folgenden Format hinzu.

```
"notifications":[{"arn":"arn:aws:sns:region:account ID:topic name"}],
```

   Ein Beispiel:

```
"notifications":[{"arn":"arn:aws:sns:us-west-2:1234567890:MySNSTopic"}],
```

   Geben Sie den Benachrichtigungsblock vor dem `resources`-Block ein, wie im folgenden Beispiel für die Region USA West (Oregon) (us-west-2) gezeigt.

```
{
    "title": "EBS snapshot copy failed",
    "description": "CloudWatch Event Rule SSMOpsItems-EBS-snapshot-copy-failed was
 triggered. Your EBS snapshot copy has failed. See below for more details.",
    "category": "Availability",
    "severity": "2",
    "source": "EC2",
    "notifications": [{
        "arn": "arn:aws:sns:us-west-2:1234567890:MySNSTopic"
    }],
```

```
    "resources": <resources>,
    "operationalData": {
        "/aws/dedup": {
            "type": "SearchableString",
            "value": "{\"dedupString\":\"SSMOpsItems-EBS-snapshot-copy-failed\"}"
        },
        "/aws/automations": {
            "value": "[ { \"automationType\": \"AWS:SSM:Automation\",
 \"automationId\": \"AWS-CopySnapshot\" } ]"
        },
        "failure-cause": {
            "value": <failure - cause>
        },
        "source": {
            "value": <source>
        },
        "start-time": {
            "value": <start - time>
        },
        "end-time": {
            "value": <end - time>
        }
    }
}
```

8. Wählen Sie Bestätigen aus.
9. Wählen Sie Weiter aus.
10. Wählen Sie Weiter aus.
11. Wählen Sie Regel aktualisieren aus.

Wenn das System das nächste Mal ein OpsItem für die Standardregel erstellt, wird eine Benachrichtigung an das Amazon-SNS-Thema veröffentlicht.

Integrieren von OpsCenter in anderen AWS-Services

OpsCenter, ein Tool in AWS Systems Manager, lässt sich in mehrere Tools integrieren, AWS-Services um Probleme mit AWS Ressourcen zu diagnostizieren und zu beheben. Sie müssen das einrichten, AWS-Service bevor Sie es mit OpsCenter integrieren.

Standardmäßig AWS-Services sind die folgenden Komponenten integriert OpsCenter und können OpsItems automatisch erstellt werden:

- Amazon CloudWatch
- Einblicke in CloudWatch Amazon-Anwendungen
- Amazon EventBridge
- AWS Config
- AWS Systems Manager Incident Manager

Sie müssen die folgenden Services mit OpsCenter integrieren, um OpsItems automatisch zu erstellen:

- DevOpsAmazon-Guru
- AWS Security Hub CSPM

Wenn einer dieser Services ein OpsItem erstellt, können Sie das OpsItem von OpsCenter verwalten und beheben. Weitere Informationen erhalten Sie unter Verwalten von OpsItems und Beheben von OpsItem-Problemen.

Weitere Informationen zu den einzelnen AWS-Service Optionen und deren Integration OpsCenter finden Sie in den folgenden Themen.

Themen

- Die OpsCenter Integration mit Amazon verstehen CloudWatch
- Grundlegendes OpsCenter zur Integration mit Amazon CloudWatch Application Insights
- OpsCenterDie Integration mit Amazon DevOps Guru verstehen
- Die OpsCenter Integration mit Amazon verstehen EventBridge
- Grundlegendes OpsCenter zur Integration mit AWS Config
- Verständnis der OpsCenter Integration mit AWS Security Hub CSPM
- OpsCenter-Integration mit Incident Manager verstehen

Die OpsCenter Integration mit Amazon verstehen CloudWatch

Amazon CloudWatch überwacht Ihre AWS Ressourcen und Services und zeigt Kennzahlen zu allen Ressourcen an AWS-Service , die Sie nutzen. CloudWatch erzeugt eineOpsItem, wenn ein Alarm in den Alarmzustand übergeht. Sie können beispielsweise einen Alarm konfigurieren, um automatisch eine OpsItem zu erstellen, wenn ein Anstieg der HTTP-Fehler von Ihrem Application Load Balancer generiert wird.

In der folgenden Liste OpsItems sind einige Alarme aufgeführt CloudWatch , deren Erstellung Sie konfigurieren können:

- Amazon DynamoDB: Lese- und Schreibaktionen in der Datenbank erreichen einen Schwellenwert
- Amazon EC2: Die CPU-Auslastung erreicht einen Schwellenwert
- AWS Abrechnung: Die geschätzten Gebühren erreichen einen Schwellenwert
- Amazon EC2: Eine Instance besteht eine Statusprüfung nicht
- Amazon Elastic Block Store (EBS): Die Festplattenspeichernutzung erreicht einen Schwellenwert

Sie können eine Warnung erstellen oder eine vorhandene Warnung bearbeiten, um ein OpsItem erstellen. Weitere Informationen finden Sie unter Konfigurieren von CloudWatch-Alarmen zum Erstellen von OpsItems.

Wenn Sie die OpsCenter Verwendung von Integrated Setup aktivieren, wird sie in CloudWatch integriertOpsCenter.

Grundlegendes OpsCenter zur Integration mit Amazon CloudWatch Application Insights

Mit Amazon CloudWatch Application Insights können Sie die am besten geeigneten Monitore für Ihre Anwendungsressourcen einrichten, um Daten kontinuierlich auf Anzeichen von Problemen mit Ihren Anwendungen zu analysieren. Wenn Sie Anwendungsressourcen in CloudWatch Application Insights konfigurieren, können Sie wählen, ob das System automatisch erstellt OpsItems werden sollOpsCenter. Für jedes Problem, das in der Anwendung entdeckt wird, wird ein OpsItem auf der OpsCenter-Konsole erstellt. Weitere Informationen finden Sie unter Einrichtung, Konfiguration und Verwaltung Ihrer Überwachungsanwendung im CloudWatch Amazon-Benutzerhandbuch.

> **Note**
>
> Ab dem 16. Oktober 2023 verwenden der Titel und die Beschreibung von OpsItems Created by CloudWatch Application Insights nun das folgende verbesserte Format:

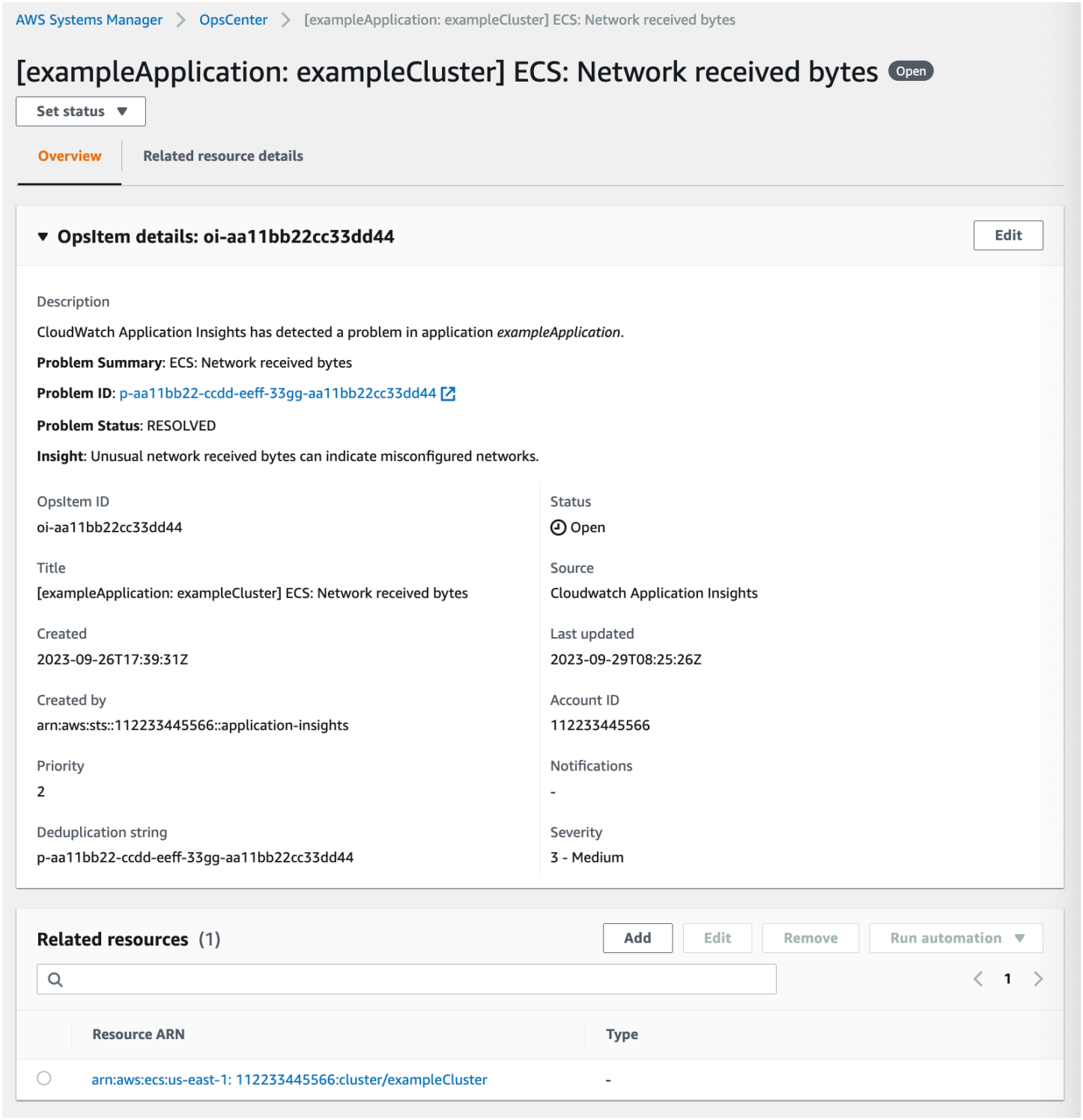
```
OpsItem title: [<APPLICATION NAME>: <RESOURCE ID>] <PROBLEM SUMMARY>

OpsItem description:

CloudWatch Application Insights has detected a problem in application <APPLICATION
 NAME>.
```

```
Problem summary: <PROBLEM SUMMARY>
Problem ID: <PROBLEM ID> (hyperlinks to the Application Insights problem summary page)
Problem Status: <PROBLEM STATUS>
Insight: <INSIGHT>
```

Ein Beispiel:



OpsCenterDie Integration mit Amazon DevOps Guru verstehen

Amazon DevOps Guru verwendet maschinelles Lernen, um Ihre Betriebsdaten, Anwendungsmetriken und Anwendungsereignisse zu analysieren und Verhaltensweisen zu identifizieren, die von normalen Betriebsmustern abweichen. Wenn Sie DevOps Guru die Generierung einer Eingabe OpsItem ermöglichenOpsCenter, generiert jede Erkenntnis eine neueOpsItem. Sie können OpsCenter verwenden, um Ihre OpsItems zu verwalten.

DevOpsGuru erstellt automatischOpsItems. Sie können Amazon DevOps Guru zum Erstellen aktivierenQuick Setup, OpsItems indem Sie, ein Tool im Systems Manager, verwenden. Das System erstellt OpsItems mithilfe der serviceverknüpften Rolle AWSServiceRoleForDevOpsGuru AWS Identity and Access Management (IAM).

Zur Integration mit Guru OpsCenter DevOps

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Quick Setup aus.
3. Wählen Sie auf der Seite „ DevOpsGuru-Konfigurationsoptionen anpassen“ den Tab Bibliothek aus.
4. Wählen Sie im DevOpsGuru-Bereich die Option Erstellen aus.
5. Wählen Sie für Konfigurationsoptionen die Option Aktivieren aus AWS Systems Manager OpsItems.
6. Wählen Sie nach Abschluss der Einrichtung Erstellen aus.

Die OpsCenter Integration mit Amazon verstehen EventBridge

Amazon EventBridge liefert eine Reihe von Ereignissen, die Änderungen an AWS Ressourcen beschreiben. Wenn Sie die OpsCenter Verwendung von Integrated Setup aktivieren, wird es in EventBridge Standardregeln integriert EventBridge und aktiviert. OpsCenter EventBridge Erstellt auf der Grundlage dieser RegelnOpsItems. Mithilfe von Regeln können Sie Ereignisse filtern und zur Untersuchung und Behebung an OpsCenter weiterleiten.

> **Note**
>
> Amazon EventBridge (ehemals Amazon CloudWatch Events) bietet alle Funktionen von CloudWatch Events und einige neue Funktionen, wie benutzerdefinierte Event-Busse, Eventquellen von Drittanbietern und Schemaregistrierung.

Im Folgenden finden Sie einige Regeln, die Sie konfigurieren können EventBridge , um eine zu erstellenOpsItem:

- Security Hub CSPM: Sicherheitswarnung ausgegeben
- Amazon DynamoDB: ein Drosselungsereignis
- Amazon Elastic Compute Cloud Auto Scaling: Instance konnte nicht gestartet werden
- Systems Manager: Fehler beim Ausführen einer Automatisierung
- AWS Health: eine Warnung für geplante Wartungsarbeiten
- Amazon EC2: Instanzstatus wurde von Running auf Stop geändert

Basierend auf Ihren Anforderungen können Sie entweder eine Regel erstellen oder eine vorhandene Regel bearbeiten, um ein OpsItems zu erstellen. Anweisungen zum Bearbeiten einer Regel zum Erstellen eines OpsItem finden Sie unter Zu erstellende EventBridge Regeln konfigurieren OpsItems.

Grundlegendes OpsCenter zur Integration mit AWS Config

AWS Config bietet einen detaillierten Überblick über die Konfiguration der AWS Ressourcen in Ihrem AWS-Konto.

AWS Config integriert sich nicht direkt inOpsCenter. Stattdessen erstellen Sie eine AWS Config Regel, die ein Ereignis an Amazon sendet EventBridge, z. B. wenn eine nicht konforme Instance AWS Config erkannt wird. Dieses Ereignis wird dann EventBridge anhand einer von Ihnen erstellten EventBridge Regel bewertet. Wenn die Regel zutrifft, wird EventBridge das Ereignis in ein Ereignis umgewandelt OpsItem und an dieses OpsCenter als Zielziel übertragen.

Mit diesem OpsItem können Sie Details der nicht konformen Ressource nachverfolgen, Untersuchungsmaßnahmen aufzeichnen und Zugriff auf konsistente Korrekturmaßnahmen gewähren.

Verwandte Informationen

Zu erstellende EventBridge Regeln konfigurieren OpsItems

Verwendung von AWS Systems ManagerOpsCenter und AWS Config für die Überwachung der Einhaltung von Vorschriften

Verständnis der OpsCenter Integration mit AWS Security Hub CSPM

AWS Security Hub CSPM sammelt Sicherheitsdaten, sogenannte Erkenntnisse, von Across AWS-Konten und Services. Security Hub CSPM verwendet eine Reihe von Regeln zur Erkennung und Generierung von Ergebnissen und hilft Ihnen dabei, Sicherheitsprobleme für die von Ihnen verwalteten Ressourcen zu identifizieren, zu priorisieren und zu beheben. Nachdem Sie die Integration wie in diesem Thema beschrieben konfiguriert haben, erstellt Systems Manager OpsItems für Security Hub CSPM-Ergebnisse in. OpsCenter

> **Note**
>
> OpsCenterhat eine bidirektionale Integration mit Security Hub CSPM. Das heißt, wenn Sie das Feld Status oder Schweregrad für OpsItem ein Sicherheitsproblem aktualisieren, synchronisiert das System die Änderungen mit Security Hub CSPM. Ebenso werden alle Änderungen an einem Erkenntnis automatisch in den entsprechenden OpsItems in OpsCenter aktualisiert.
>
> Wenn aus einem Security Hub CSPM-Befund ein erstellt OpsItem wird, werden Security Hub CSPM-Metadaten automatisch zum Betriebsdatenfeld von hinzugefügt. OpsItem Wenn diese Metadaten gelöscht werden, funktionieren die bidirektionalen Updates nicht mehr.

Standardmäßig erstellt Systems Manager OpsItems für kritische und hochgradig schwerwiegende Befunde. Sie können OpsCenter für die Erstellung von OpsItems für Erkenntnisse mit mittlerem und niedrigem Schweregrad manuell konfigurieren. OpsCenter erstellt kein OpsItems zu einem informativen Ergebnis, da keine Behebung erforderlich ist. Weitere Informationen zu den Schweregraden von Security Hub CSPM finden Sie unter Schweregrad in der AWS Security Hub API-Referenz.

Bevor Sie beginnen

Bevor Sie OpsCenter die Konfiguration für die Erstellung auf der OpsItems Grundlage von Security Hub CSPM-Ergebnissen konfigurieren, stellen Sie sicher, dass Sie die Security Hub CSPM-Einrichtungsaufgaben abgeschlossen haben. Weitere Informationen finden Sie unter Security Hub CSPM einrichten im AWS Security Hub Benutzerhandbuch.

Wenn Sie Security Hub CSPM mit integrierenOpsCenter, erstellt das System OpsItems mithilfe der serviceverknüpften `AWSServiceRoleForSystemsManagerOpsDataSync` IAM-Rolle. Weitere Informationen über diese Rolle finden Sie unter Verwenden von Rollen zum Erstellen OpsData und OpsItems für Explorer.

Warning

Beachten Sie die folgenden wichtigen Informationen zur Preisgestaltung für die OpsCenter Integration mit Security Hub CSPM:

- Wenn Sie bei der Konfiguration OpsCenter und der Security Hub CSPM-Integration mit dem Security Hub CSPM-Administratorkonto angemeldet sind, erstellt das System OpsItems für Ergebnisse im Administrator - und allen Mitgliedskonten. Die OpsItems werden alle im Administratorkonto erstellt. Abhängig von einer Vielzahl von Faktoren kann dies zu einer unerwartet hohen Rechnungssumme führen. AWS

  Wenn Sie bei der Konfiguration der Integration in einem Mitgliedskonto angemeldet sind, erstellt das System nur OpsItems für Erkenntnisse in diesem individuellen Konto. Weitere Informationen zum Security Hub CSPM-Administratorkonto, zu Mitgliedskonten und deren Beziehung zum EventBridge Event-Feed für Ergebnisse finden Sie unter Typen der Security Hub-CSPM-Integration mit EventBridge im Benutzerhandbuch.AWS Security Hub
- Für jede Erkenntnis, das ein OpsItem erstellt, wird Ihnen der reguläre Preis für die Erstellung des OpsItem berechnet. Ihnen wird auch berechnet, wenn Sie das bearbeiten OpsItem oder wenn das entsprechende Ergebnis in Security Hub CSPM aktualisiert wird (was ein OpsItem Update auslöst).
- OpsItemsdie durch eine Integration mit erstellt wurden, AWS Security Hub CSPM sind derzeit nicht durch das maximale Kontingent von 500.000 OpsItems pro Konto in einer Region begrenzt. Es ist daher möglich, dass Security Hub CSPM-Benachrichtigungen in einem Konto mehr als 500.000 kostenpflichtige Gebühren OpsItems in jeder Region erstellen.

  Für Umgebungen mit hoher Produktion empfehlen wir daher, den Umfang der CSPM-Ergebnisse von Security Hub auf Probleme mit hohem Schweregrad zu beschränken.

So konfigurieren Sie OpsCenter die Erstellung von CSPM-Ergebnissen OpsItems für Security Hub

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie Einstellungen aus.
4. Wählen Sie im Abschnitt Security Hub CSPM-Ergebnisse die Option Bearbeiten aus.
5. Wählen Sie den Schieberegler, um Deaktiviert zu Aktiviert zu ändern.
6. Wenn Sie möchten, dass das System OpsItems für Erkenntnisse mit mittlerem oder niedrigem Schweregrad erstellt, schalten Sie diese Optionen ein.
7. Wählen Sie Speichern aus, um Ihre Konfiguration zu speichern.

Gehen Sie wie folgt vor, wenn Sie nicht mehr möchten, dass das System CSPM-Ergebnisse OpsItems für Security Hub erstellt.

Um den Empfang von OpsItems Security Hub CSPM-Ergebnissen zu beenden

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie Einstellungen aus.
4. Wählen Sie im Abschnitt Security Hub CSPM-Ergebnisse die Option Bearbeiten aus.
5. Wählen Sie den Schieberegler, um Aktiviert zu Deaktiviert zu ändern. Wenn Sie den Schieberegler nicht umschalten können, wurde Security Hub CSPM für Sie nicht aktiviert. AWS-Konto
6. Wählen Sie Speichern, um Ihre Konfiguration zu speichern. OpsCentererstellt nicht mehr auf OpsItems Grundlage der CSPM-Ergebnisse von Security Hub.

> **Important**
>
> Ein von Systems Manager delegierter Administrator oder das AWS Organizations Verwaltungskonto können Security Hub CSPM-Ergebnisse OpsCenter für mehrere Konten aktivieren und eine AWS-Regionen Ressourcendatensynchronisierung in erstellen. Explorer Wenn die Security Hub CSPM-Quelle aktiviert ist Explorer und eine Ressourcendatensynchronisierung existiert, die auf das Mitgliedskonto abzielt, in

dem Sie die Security Hub CSPM-Integration deaktiviert haben, haben die von Ihrem Administrator ausgewählten Einstellungen Vorrang. OpsCentererstellt weiterhin CSPM-Ergebnisse OpsItems für Security Hub. Wenn Sie die Erstellung von CSPM-Ergebnissen OpsItems für Security Hub in einem Mitgliedskonto beenden möchten, für das eine Ressourcendatensynchronisierung vorgesehen ist, wenden Sie sich an Ihren Administrator und bitten Sie ihn, Ihr Konto aus der Ressourcendatensynchronisierung zu entfernen oder die Security Hub CSPM-Quelle in zu deaktivieren. Explorer Weitere Informationen zum Ändern dieser Einstellung in Explorer finden Sie unter Bearbeiten von Systems-Manager-Explorer-Datenquellen.

OpsCenter-Integration mit Incident Manager verstehen

Incident Manager, ein Tool in AWS Systems Manager, bietet eine Incident-Management-Konsole, mit der Sie Vorfälle, die Ihre gehosteten Anwendungen betreffen, eindämmen und beheben können. AWS Ein Vorfall ist jede Art von ungeplanter Unterbrechung oder Beeinträchtigung der Qualität von Services. Nachdem Sie Incident Manager eingerichtet und konfiguriert haben, erstellt das System automatisch OpsItems in OpsCenter.

Wenn das System einen Vorfall in Incident Manager erstellt, erstellt es auch ein OpsItem in OpsCenter und zeigt den Vorfall als zugehöriges Element an. Wenn das OpsItem bereits vorhanden ist, erstellt Incident Manager kein OpsItem. Das erste OpsItem ist als übergeordnetes OpsItem bekannt. Wenn ein Vorfall an Größe und Umfang zunimmt, können Sie Vorfälle zu einem vorhandenen OpsItem hinzufügen. Bei Bedarf können Sie manuell einen Vorfall für ein OpsItem erstellen. Nachdem ein Vorfall abgeschlossen ist, können Sie in Incident Manager eine Analyse erstellen, um den Behebungsprozess für ähnliche Probleme zu überprüfen und zu verbessern.

Standardmäßig ist OpsCenter mit Incident Manager integriert. Wenn Incident Manager nicht eingerichtet ist, wird auf der OpsCenter Seite eine Meldung zur Einrichtung von Incident Manager angezeigt. Wenn Incident Manager ein OpsItem erstellt, können Sie das OpsItem von OpsCenter verwalten und korrigieren. Anweisungen zum Erstellen eines Vorfalls für ein OpsItem finden Sie unter Erstellen eines Vorfalls für ein OpsItem.

Geben Sie einen Namen für den Benutzer ein und klicken Sie dann auf OpsItems

Nachdem Sie OpsCenter, ein Tool in AWS Systems Manager, eingerichtet und in Ihre AWS-Services integriert haben, erstellen die AWS-Services automatisch OpsItems anhand von Standardregeln, Ereignissen oder Warnungen.

Sie können die Status und Schweregrade der standardmäßigen Amazon EventBridge-Regeln anzeigen. Falls erforderlich, können Sie diese Regeln in Amazon EventBridge erstellen oder bearbeiten. Sie können auch Warnungen von Amazon CloudWatch anzeigen und Warnungen erstellen oder bearbeiten. Mit Regeln und Warnungen können Sie Ereignisse konfigurieren, für die Sie OpsItems automatisch generieren möchten.

Wenn das System ein OpsItem erstellt, befindet es sich im Status Offen. Sie können den Status zu In Bearbeitung ändern, wenn Sie mit der Untersuchung des OpsItem beginnen und in Gelöst, nachdem Sie das OpsItem behoben haben. Weitere Informationen zum Konfigurieren von Alarmen und Regeln in AWS-Services zum Erstellen von OpsItems und zum manuellen Erstellen von OpsItems finden Sie in den folgenden Themen.

Themen

- Zu erstellende EventBridge Regeln konfigurieren OpsItems
- Konfigurieren von CloudWatch-Alarmen zum Erstellen von OpsItems
- Manuelles Erstellen der OpsItems

Zu erstellende EventBridge Regeln konfigurieren OpsItems

Wenn Amazon ein Ereignis EventBridge empfängt, erstellt es ein neues, das auf Standardregeln OpsItem basiert. Sie können eine Regel erstellen oder eine bestehende Regel bearbeiten, um sie OpsCenter als Ziel für ein EventBridge Ereignis festzulegen. Informationen zum Erstellen einer Event-Regel finden Sie unter Creating a rule for an AWS-Service im EventBridge Amazon-Benutzerhandbuch.

So konfigurieren Sie eine EventBridge Regel zum Erstellen OpsItems in OpsCenter

1. Öffnen Sie die EventBridge Amazon-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Regeln aus.
3. Auf der Seite Regeln wählen Sie für Event Bus die Option Standard.
4. Wählen Sie für Regeln eine Regel aus, indem Sie das Kontrollkästchen neben dessen Namen aktivieren.
5. Wählen Sie den Namen der Regel aus, um die Detailseite zu öffnen. Stellen Sie im Abschnitt Regeldetails sicher, dass der Status auf Aktiviert festgelegt ist.

> **Note**
>
> Falls erforderlich, können Sie den Status mit Bearbeiten in der oberen rechten Ecke der Seite aktualisieren.

6. Wählen Sie die Registerkarte Ziele.
7. Klicken Sie in der Registerkarte Targets (Ziele) auf Edit (Bearbeiten).
8. Wählen Sie für Zieltypen aus AWS-Service.
9. Für Select a target (Ziel auswählen), wählen Sie Systems Manager OpsItem.
10. Für viele Zieltypen ist die Erlaubnis EventBridge erforderlich, Ereignisse an das Ziel zu senden. In diesen Fällen EventBridge kann die AWS Identity and Access Management (IAM) -Rolle erstellt werden, die für die Ausführung Ihrer Regel erforderlich ist:

    - Um automatisch eine IAM-Rolle zu erstellen, wählen Sie Eine neue Rolle für diese spezifische Ressource erstellen.
    - Um eine IAM-Rolle zu verwenden, die Sie erstellt haben, um die EventBridge Berechtigung zum Erstellen OpsItems in zu erteilenOpsCenter, wählen Sie Bestehende Rolle verwenden.
11. Wählen Sie unter Zusätzliche Einstellungen für Zieleingabe konfigurieren die Option Eingabe-Transformator aus.

    Sie können die Option Eingabe-Transformer verwenden, um eine Deduplizierungszeichenfolge und andere wichtige Informationen für OpsItems anzugeben, z. B. Titel und Schweregrad.
12. Wählen Sie Configure input transformer (Eingabetransformator konfigurieren).
13. Geben Sie unter Zieleingabe-Transformator für Eingabepfad die Werte an, die aus dem auslösenden Ereignis analysiert werden sollen. Um beispielsweise die Startzeit, die Endzeit und andere Details des Ereignisses zu analysieren, das die Regel auslöst, verwenden Sie den folgenden JSON.

```
{
    "end-time": "$.detail.EndTime",
    "failure-cause": "$.detail.cause",
    "resources": "$.resources[0]",
    "source": "$.detail.source",
    "start-time": "$.detail.StartTime"
}
```

14. Geben Sie für Template (Vorlage) die Informationen an, die an das Ziel gesendet werden sollen. Verwenden Sie beispielsweise den folgenden JSON, um Informationen an OpsCenter zu übergeben. Die Informationen werden verwendet, um eine OpsItem zu erstellen.

> **Note**
>
> Wenn die Eingabevorlage im JSON-Format vorliegt, darf der Objektwert in der Vorlage keine Anführungszeichen enthalten. Beispielsweise dürfen die Werte für Ressourcen, Fehlerursache, Quelle, Startzeit und Endzeit nicht in Anführungszeichen stehen.

```
{
    "title": "EBS snapshot copy failed",
    "description": "CloudWatch Event Rule SSMOpsItems-EBS-snapshot-copy-failed was
 triggered. Your EBS snapshot copy has failed. See below for more details.",
    "category": "Availability",
    "severity": "2",
    "source": "EC2",
    "operationalData": {
        "/aws/dedup": {
            "type": "SearchableString",
            "value": "{\"dedupString\":\"SSMOpsItems-EBS-snapshot-copy-failed\"}"
        },
        "/aws/automations": {
            "value": "[ { \"automationType\": \"AWS:SSM:Automation\",
 \"automationId\": \"AWS-CopySnapshot\" } ]"
        },
        "failure-cause": {
            "value": <failure-cause>
        },
        "source": {
            "value": <source>
        },
        "start-time": {
            "value": <start-time>
        },
        "end-time": {
            "value": <end-time>
        },
         },
        "resources": {
            "value": <resources>
```

```
        }
    }
}
```

Weitere Informationen zu diesen Feldern finden Sie unter Transformieren der Zieleingabe im EventBridge Amazon-Benutzerhandbuch.

15. Wählen Sie Bestätigen aus.
16. Wählen Sie Weiter aus.
17. Wählen Sie Weiter aus.
18. Wählen Sie Regel aktualisieren aus.

Nachdem ein OpsItem aus einem Ereignis erstellt wurde, können Sie die Ereignisdetails einsehen. Öffnen Sie hierzu das OpsItem und blättern Sie nach unten zum Abschnitt Private operational data (Private Betriebsdaten). Weitere Informationen zum Konfigurieren der Optionen in einem OpsItem finden Sie unter Verwalten von OpsItems.

Konfigurieren von CloudWatch-Alarmen zum Erstellen von OpsItems

Während der integrierten Einrichtung von OpsCenter, einem Tool in AWS Systems Manager, aktivieren Sie Amazon CloudWatch, um automatisch OpsItems basierend auf allgemeinen Warnungen zu erstellen. Sie können eine Warnungerstellen oder eine vorhandenen Warnung bearbeiten, um OpsItems in OpsCenter zu erstellen.

CloudWatch erstellt automatisch eine neue serviceverknüpfte Rolle in AWS Identity and Access Management (IAM), wenn Sie eine Warnung konfigurieren, um OpsItems zu erstellen. Der Name der neuen Rolle lautet `AWSServiceRoleForCloudWatchAlarms_ActionSSM`. Weitere Informationen zu serviceverknüpften CloudWatch-Rollen finden Sie unter Verwenden von serviceverknüpften Rollen für CloudWatch im Amazon-CloudWatch-Benutzerhandbuch.

Wenn eine CloudWatch-Warnung ein OpsItem generiert, zeigt OpsItem die CloudWatch-Warnung an – '***alarm_name***' befindet sich im ALARM-Status.

Um Details zu einem bestimmten OpsItem anzuzeigen, wählen Sie das OpsItem und dann die Registerkarte Details zu zugehörigen Ressourcen aus. Sie können OpsItems manuell ändern, um Details wie den Schweregrad oder die Kategorie zu ändern. Wenn Sie jedoch den Schweregrad oder die Kategorie einer Warnung bearbeiten, kann Systems Manager den Schweregrad oder die Kategorie von OpsItems, die bereits aus der Warnung erstellt wurden, nicht aktualisieren. Wenn ein Alarm ein OpsItem erstellt hat und Sie eine Deduplizierungszeichenfolge angegeben haben, wird

der Alarm keine weiteren OpsItems erstellen, selbst wenn Sie den Alarm in CloudWatch bearbeiten. Wenn das OpsItem in OpsCenter gelöst ist, erstellt CloudWatch ein neues OpsItem.

Weitere Informationen zur Konfiguration von CloudWatch-Warnungen finden Sie in den folgenden Themen.

Themen

- Konfiguration eines zu erstellenden CloudWatch Alarms OpsItems (Konsole)
- Konfiguration eines vorhandenen CloudWatch Alarms zur Erstellung OpsItems (programmgesteuert)

Konfiguration eines zu erstellenden CloudWatch Alarms OpsItems (Konsole)

Sie können manuell einen Alarm erstellen oder einen vorhandenen Alarm aktualisieren, um ihn OpsItems von Amazon zu erstellen CloudWatch.

Um einen CloudWatch Alarm zu erstellen und Systems Manager als Ziel dieses Alarms zu konfigurieren

1. Führen Sie die Schritte 1—9 aus, wie im CloudWatch Amazon-Benutzerhandbuch unter Erstellen eines CloudWatch Alarms basierend auf einem statischen Schwellenwert beschrieben.
2. Wählen Sie im Bereich Systems Manager Manager-Aktion die Option Systems Manager OpsCenter Manager-Aktion hinzufügen aus.
3. Wählen Sie OpsItems.
4. Wählen Sie für Schweregrad eine Zahl von 1 bis 4 aus.
5. (Optional) Wählen Sie für Kategorie eine Kategorie für das OpsItem aus.
6. Führen Sie die Schritte 11—13 aus, wie unter Erstellen eines CloudWatch Alarms auf der Grundlage eines statischen Schwellenwerts im CloudWatch Amazon-Benutzerhandbuch beschrieben.
7. Klicken Sie auf Next (Weiter) und schließen Sie den Assistenten ab.

Bearbeiten eines vorhandenen Alarms und Konfigurieren des Systems Managers als Ziel dieses Alarms

1. Öffnen Sie die CloudWatch Konsole unter. https://console.aws.amazon.com/cloudwatch/
2. Klicken Sie im Navigationsbereich auf Alarms (Alarme).

3. Wählen Sie den Alarm aus, wählen Sie dann Actions und anschließend Edit.
4. (Optional) Ändern Sie Einstellungen in den Bereichen Metrics (Metriken) und Conditions (Bedingungen) und wählen Sie dann Next (Weiter).
5. Wählen Sie im Bereich Systems Manager Add Systems Manager OpsCenter action.
6. Wählen Sie für Schweregrad eine Zahl aus.

> **Note**
>
> Der Schweregrad ist ein benutzerdefinierter Wert. Sie oder Ihre Organisation bestimmen, was jeder Schweregrad bedeutet und welche Service-Level-Vereinbarungen mit jedem Schweregrad verknüpft sind.

7. (Optional) Wählen Sie für Category eine Option aus.
8. Klicken Sie auf Next (Weiter) und schließen Sie den Assistenten ab.

Konfiguration eines vorhandenen CloudWatch Alarms zur Erstellung OpsItems (programmgesteuert)

Sie können CloudWatch Amazon-Alarme so konfigurieren, dass sie OpsItems programmgesteuert erstellt werden, indem Sie AWS Command Line Interface (AWS CLI), AWS CloudFormation Vorlagen oder Java Codefragmente verwenden.

Topics

- Bevor Sie beginnen
- Konfiguration der zu erstellenden CloudWatch Alarme OpsItems (AWS CLI)
- Konfiguration von CloudWatch Alarmen zum Erstellen oder Aktualisieren OpsItems (CloudFormation)
- Konfiguration von CloudWatch Alarmen zum Erstellen oder Aktualisieren OpsItems (Java)

Bevor Sie beginnen

Wenn Sie eine vorhandenen Warnung programmgesteuert bearbeiten oder eine neuen Warnung erstellen, der OpsItems erstellt, müssen Sie einen Amazon-Ressourcennamen (ARN) angeben. Dieser ARN identifiziert Systems Manager OpsCenter als Ziel für OpsItems, die aus dem Alarm erstellt wurden. Sie können den ARN so anpassen, dass OpsItems, die aus dem Alarm erstellt wurden, bestimmte Informationen wie Schweregrad oder Kategorie enthalten. Jede ARN enthält die in der folgenden Tabelle beschriebenen Informationen.

Parameter	Details
`Region` (Erforderlich)	Der AWS-Region Ort, an dem der Alarm existiert. Beispiel: `us-west-2` . Weitere Informationen zu AWS-Regionen , in denen Sie OpsCenter verwenden können, finden Sie unter AWS Systems Manager -Endpunkte und Kontingente.
`account_ID` (Erforderlich)	Dieselbe AWS-Konto ID, mit der der Alarm erstellt wurde. Beispiel: `123456789012` . Der Konto-ID muss ein Doppelpunkt (`:`) und der Parameter `opsitem` folgen, wie in den folgenden Beispielen gezeigt.
`severity` (Erforderlich)	Ein benutzerdefinierter Schweregrad für OpsItems wurde aus dem Alarm erstellt. Zulässige Werte: `1`, `2`, `3`, `4`
`Category` (Optional)	Eine Kategorie für OpsItems, die aus dem Alarm erstellt wurde. Gültige Werte: `Availability` , `Cost`, `Performance` , `Recovery` und `Security`.

Erstellen Sie den ARN mit der folgenden Syntax. Dieser ARN enthält nicht den optionalen `Category`-Parameter.

```
arn:aws:ssm:Region:account_ID:opsitem:severity
```

Im Folgenden sehen Sie ein Beispiel.

```
arn:aws:ssm:us-west-2:123456789012:opsitem:3
```

Verwenden Sie die folgende Syntax, um einen ARN zu erstellen, der den optionalen `Category`-Parameter verwendet.

```
arn:aws:ssm:Region:account_ID:opsitem:severity#CATEGORY=category_name
```

Im Folgenden sehen Sie ein Beispiel.

```
arn:aws:ssm:us-west-2:123456789012:opsitem:3#CATEGORY=Security
```

Konfiguration der zu erstellenden CloudWatch Alarme OpsItems (AWS CLI)

Für diesen Befehl müssen Sie einen ARN für den `alarm-actions`-Parameter angeben. Informationen zum Erstellen des ARN finden Sie unter Bevor Sie beginnen.

Um einen zu CloudWatch erstellenden Alarm zu konfigurieren OpsItems (AWS CLI)

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus, um Informationen über den Alarm zu sammeln, den Sie konfigurieren möchten.

   ```
   aws cloudwatch describe-alarms --alarm-names "alarm name"
   ```
3. Führen Sie den folgenden Befehl aus, um einen Alarm zu aktualisieren. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   ```
   aws cloudwatch put-metric-alarm --alarm-name name \
   --alarm-description "description" \
   --metric-name name --namespace namespace \
   --statistic statistic --period value --threshold value \
   --comparison-operator value \
   --dimensions "dimensions" --evaluation-periods value \
       --alarm-actions
    arn:aws:ssm:Region:account_ID:opsitem:severity#CATEGORY=category_name \
   --unit unit
   ```

   Ein Beispiel:

   Linux & macOS

   ```
   aws cloudwatch put-metric-alarm --alarm-name cpu-mon \
   --alarm-description "Alarm when CPU exceeds 70 percent" \
   ```

```
--metric-name CPUUtilization --namespace AWS/EC2 \
--statistic Average --period 300 --threshold 70 \
--comparison-operator GreaterThanThreshold \
--dimensions "Name=InstanceId,Value=i-12345678" --evaluation-periods 2 \
--alarm-actions arn:aws:ssm:us-east-1:123456789012:opsitem:3#CATEGORY=Security \
--unit Percent
```

Windows

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon ^
--alarm-description "Alarm when CPU exceeds 70 percent" ^
--metric-name CPUUtilization --namespace AWS/EC2 ^
--statistic Average --period 300 --threshold 70 ^
--comparison-operator GreaterThanThreshold ^
--dimensions "Name=InstanceId,Value=i-12345678" --evaluation-periods 2 ^
--alarm-actions arn:aws:ssm:us-east-1:123456789012:opsitem:3#CATEGORY=Security ^
--unit Percent
```

Konfiguration von CloudWatch Alarmen zum Erstellen oder Aktualisieren OpsItems (CloudFormation)

Dieser Abschnitt enthält AWS CloudFormation Vorlagen, mit denen Sie CloudWatch Alarme so konfigurieren können, dass sie automatisch erstellt oder aktualisiert werdenOpsItems. Für jede Vorlage muss ein ARN für den `AlarmActions`-Parameter angegeben werden. Informationen zum Erstellen des ARN finden Sie unter Bevor Sie beginnen.

Metrischer Alarm — Verwenden Sie die folgende CloudFormation Vorlage, um einen CloudWatch metrischen Alarm zu erstellen oder zu aktualisieren. Der in dieser Vorlage angegebene Alarm überwacht die Statusprüfungen der Amazon Elastic Compute Cloud (Amazon EC2) -Instance. Wenn der Alarm in den `ALARM`-Zustand wecchselt, erstellt es ein OpsItem in OpsCenter.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters" : {
    "RecoveryInstance" : {
      "Description" : "The EC2 instance ID to associate this alarm with.",
      "Type" : "AWS::EC2::Instance::Id"
    }
  },
  "Resources": {
```

```
       "RecoveryTestAlarm": {
         "Type": "AWS::CloudWatch::Alarm",
         "Properties": {
           "AlarmDescription": "Run a recovery action when instance status check fails
 for 15 consecutive minutes.",
           "Namespace": "AWS/EC2" ,
           "MetricName": "StatusCheckFailed_System",
           "Statistic": "Minimum",
           "Period": "60",
           "EvaluationPeriods": "15",
           "ComparisonOperator": "GreaterThanThreshold",
           "Threshold": "0",
           "AlarmActions": [ {"Fn::Join" : ["",
 ["arn:arn:aws:ssm:Region:account_ID:opsitem:severity#CATEGORY=category_name",
 { "Ref" : "AWS::Partition" }, ":ssm:", { "Ref" : "AWS::Region" }, { "Ref" : "AWS::
 AccountId" }, ":opsitem:3" ]]} ],
           "Dimensions": [{"Name": "InstanceId","Value": {"Ref": "RecoveryInstance"}}]
         }
       }
     }
   }
```

Zusammengesetzter Alarm — Verwenden Sie die folgende CloudFormation Vorlage, um einen zusammengesetzten Alarm zu erstellen oder zu aktualisieren. Ein zusammengesetzter Alarm besteht aus mehreren Metrikalarmen. Wenn der Alarm in den ALARM-Zustand wecchselt, erstellt es ein OpsItem in OpsCenter.

```
"Resources":{
       "HighResourceUsage":{
          "Type":"AWS::CloudWatch::CompositeAlarm",
          "Properties":{
             "AlarmName":"HighResourceUsage",
             "AlarmRule":"(ALARM(HighCPUUsage) OR ALARM(HighMemoryUsage)) AND NOT
 ALARM(DeploymentInProgress)",

 "AlarmActions":"arn:aws:ssm:Region:account_ID:opsitem:severity#CATEGORY=category_name",
             "AlarmDescription":"Indicates that the system resource usage is high while
 no known deployment is in progress"
          },
          "DependsOn":[
             "DeploymentInProgress",
             "HighCPUUsage",
             "HighMemoryUsage"
```

```
         ]
      },
      "DeploymentInProgress":{
         "Type":"AWS::CloudWatch::CompositeAlarm",
         "Properties":{
            "AlarmName":"DeploymentInProgress",
            "AlarmRule":"FALSE",
            "AlarmDescription":"Manually updated to TRUE/FALSE to disable other
alarms"
         }
      },
      "HighCPUUsage":{
         "Type":"AWS::CloudWatch::Alarm",
         "Properties":{
            "AlarmDescription":"CPUusageishigh",
            "AlarmName":"HighCPUUsage",
            "ComparisonOperator":"GreaterThanThreshold",
            "EvaluationPeriods":1,
            "MetricName":"CPUUsage",
            "Namespace":"CustomNamespace",
            "Period":60,
            "Statistic":"Average",
            "Threshold":70,
            "TreatMissingData":"notBreaching"
         }
      },
      "HighMemoryUsage":{
         "Type":"AWS::CloudWatch::Alarm",
         "Properties":{
            "AlarmDescription":"Memoryusageishigh",
            "AlarmName":"HighMemoryUsage",
            "ComparisonOperator":"GreaterThanThreshold",
            "EvaluationPeriods":1,
            "MetricName":"MemoryUsage",
            "Namespace":"CustomNamespace",
            "Period":60,
            "Statistic":"Average",
            "Threshold":65,
            "TreatMissingData":"breaching"
         }
      }
   }
```

Konfiguration von CloudWatch Alarmen zum Erstellen oder Aktualisieren OpsItems (Java)

Dieser Abschnitt enthält Java Codefragmente, mit denen Sie CloudWatch Alarme so konfigurieren können, dass sie automatisch erstellt oder aktualisiert werden. OpsItems Für jeden Ausschnitt müssen Sie einen ARN für den `validSsmActionStr`-Parameter angeben. Informationen zum Erstellen des ARN finden Sie unter Bevor Sie beginnen.

Ein bestimmter Alarm — Verwenden Sie den folgenden Java Codeausschnitt, um einen Alarm zu erstellen oder zu aktualisieren. CloudWatch Der in dieser Vorlage angegebene Alarm überwacht die Statusprüfungen der EC2 Amazon-Instance. Wenn der Alarm in den `ALARM`-Zustand wecchselt, erstellt es ein OpsItem in OpsCenter.

```
import com.amazonaws.services.cloudwatch.AmazonCloudWatch;
    import com.amazonaws.services.cloudwatch.AmazonCloudWatchClientBuilder;
    import com.amazonaws.services.cloudwatch.model.ComparisonOperator;
    import com.amazonaws.services.cloudwatch.model.Dimension;
    import com.amazonaws.services.cloudwatch.model.PutMetricAlarmRequest;
    import com.amazonaws.services.cloudwatch.model.PutMetricAlarmResult;
    import com.amazonaws.services.cloudwatch.model.StandardUnit;
    import com.amazonaws.services.cloudwatch.model.Statistic;

    private void putMetricAlarmWithSsmAction() {
        final AmazonCloudWatch cw =
                AmazonCloudWatchClientBuilder.defaultClient();

        Dimension dimension = new Dimension()
                .withName("InstanceId")
                .withValue(instanceId);

        String validSsmActionStr =
 "arn:aws:ssm:Region:account_ID:opsitem:severity#CATEGORY=category_name";

        PutMetricAlarmRequest request = new PutMetricAlarmRequest()
                .withAlarmName(alarmName)
                .withComparisonOperator(
                        ComparisonOperator.GreaterThanThreshold)
                .withEvaluationPeriods(1)
                .withMetricName("CPUUtilization")
                .withNamespace("AWS/EC2")
                .withPeriod(60)
                .withStatistic(Statistic.Average)
                .withThreshold(70.0)
                .withActionsEnabled(false)
```

```
                .withAlarmDescription(
                        "Alarm when server CPU utilization exceeds 70%")
                .withUnit(StandardUnit.Seconds)
                .withDimensions(dimension)
                .withAlarmActions(validSsmActionStr);

        PutMetricAlarmResult response = cw.putMetricAlarm(request);
    }
```

Alle Alarme aktualisieren — Verwenden Sie den folgenden Java Codeausschnitt, um alle CloudWatch Alarme in Ihrem System AWS-Konto zu aktualisieren, OpsItems wenn ein Alarm in den `ALARM` Status wechselt.

```
import com.amazonaws.services.cloudwatch.AmazonCloudWatch;
    import com.amazonaws.services.cloudwatch.AmazonCloudWatchClientBuilder;
    import com.amazonaws.services.cloudwatch.model.DescribeAlarmsRequest;
    import com.amazonaws.services.cloudwatch.model.DescribeAlarmsResult;
    import com.amazonaws.services.cloudwatch.model.MetricAlarm;

    private void listMetricAlarmsAndAddSsmAction() {
        final AmazonCloudWatch cw = AmazonCloudWatchClientBuilder.defaultClient();

        boolean done = false;
        DescribeAlarmsRequest request = new DescribeAlarmsRequest();

        String validSsmActionStr =
 "arn:aws:ssm:Region:account_ID:opsitem:severity#CATEGORY=category_name";

        while(!done) {

            DescribeAlarmsResult response = cw.describeAlarms(request);

            for(MetricAlarm alarm : response.getMetricAlarms()) {
                // assuming there are no alarm actions added for the metric alarm
                alarm.setAlarmActions(ImmutableList.of(validSsmActionStr));
            }

            request.setNextToken(response.getNextToken());

            if(response.getNextToken() == null) {
                done = true;
            }
        }
```

```
    }
```

Manuelles Erstellen der OpsItems

Wenn Sie ein Betriebsproblem feststellen, können Sie in OpsCenter, einem Tool in AWS Systems Manager, manuell ein OpsItem erstellen, um das Problem zu verwalten und zu lösen.

Wenn Sie eine OpsCenter für eine kontoübergreifende Verwaltung einrichten, kann ein delegierter Administrator von Systems Manager oder einem AWS Organizations-Verwaltungskonto OpsItems für Mitgliederkonten erstellen. Weitere Informationen finden Sie unter (Optional) Manuelle Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems.

Sie können OpsItems mit der AWS Systems Manager-Konsole, der AWS Command Line Interface (AWS CLI), oder der AWS Tools for Windows PowerShell erstellen.

Themen

- Manuelles Erstellen von OpsItems (Konsole)
- Manuelles Erstellen von OpsItems (AWS CLI)
- Manuelles Erstellen von OpsItems (PowerShell)

Manuelles Erstellen von OpsItems (Konsole)

Sie können OpsItems manuell mit der AWS Systems Manager-Konsole erstellen. Wenn Sie ein OpsItem erstellen, wird es in Ihrem OpsCenter-Konto angezeigt. Wenn Sie OpsCenter für die kontoübergreifende Verwaltung einrichten, bietet OpsCenter dem delegierten Administrator oder dem Verwaltungskonto die Möglichkeit, OpsItems für ausgewählte Mitgliedskonten zu erstellen. Weitere Informationen finden Sie unter (Optional) Manuelle Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems.

So erstellen Sie ein OpsItem mit der AWS Systems Manager-Konsole

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie Create (Erstellen)OpsItem aus. Wenn diese Schaltfläche nicht angezeigt wird, wählen Sie die Registerkarte OpsItems und dann Create OpsItem aus.

4. (Optional) Wählen Sie Anderes Konto und dann das Konto aus, für das Sie das OpsItem erstellen möchten.

   > **Note**
   >
   > Dieser Schritt ist erforderlich, wenn Sie OpsItems für ein Mitgliedskonto erstellen.

5. Geben Sie unter Title (Titel) einen aussagekräftigen Namen ein, um den Zweck des OpsItem zu verstehen.
6. Geben Sie unter Source (Quelle) den Typ der betroffenen AWS-Ressource oder andere Informationen der Quelle ein, damit Benutzer die Herkunft des OpsItem verstehen.

   > **Note**
   >
   > Es ist nicht möglich, das Feld Source (Quelle) nach dem Erstellen des OpsItem zu bearbeiten.

7. (Optional) Wählen Sie unter Priority (Priorität) die Priorität aus.
8. (Optional) Wählen Sie für Severity (Schweregrad) den Schweregrad aus.
9. (Optional) Wählen Sie für Category (Kategorie) eine Kategorie aus.
10. Geben Sie unter Description (Beschreibung) Informationen zu diesem OpsItem ein, einschließlich der Schritte, um das Problem zu reproduzieren (falls zutreffend).

    > **Note**
    >
    > Die Konsole unterstützt die meisten Markdown-Formate im OpsItem-Beschreibungsfeld. Weitere Informationen finden Sie unter Verwenden von Markdown in der Konsole im Einsteiger-Handbuch Erste Schritte mit der AWS-Managementkonsole.

11. Geben Sie für Deduplizierungszeichenfolge Wörter ein, die das System bei der Überprüfung auf OpsItems-Duplikate überprüfen soll. Weitere Informationen zu Deduplizierungszeichenfolgen finden Sie unter Verwalten von OpsItems-Duplikaten.
12. (Optional) Geben Sie unter Benachrichtigungen den Amazon-Ressourcenname (ARN) des Amazon-SNS-Themas an, an das Benachrichtigungen gesendet werden sollen, wenn dieses OpsItem aktualisiert wird. Sie müssen einen Amazon SNS-ARN angeben, der sich in derselben AWS-Region wie das OpsItem befindet.

13. (Optional) Wählen Sie unter Zugehörige Ressourcen die Option Hinzufügen zur Angabe der ID oder des ARN der betroffenen Ressource und aller zugehörigen Ressourcen.
14. Wählen Sie ErstellenOpsItem aus.

Bei Erfolg wird auf der Seite das OpsItem angezeigt. Wenn ein delegiertes Administrator- oder Verwaltungskonto ein OpsItem für ausgewählte Mitglieder erstellt, werden die neuen OpsItems im OpsCenter des Administrator- und Mitgliedskontos angezeigt. Weitere Informationen zum Konfigurieren der Optionen in einem OpsItem finden Sie unter Verwalten von OpsItems.

Manuelles Erstellen von OpsItems (AWS CLI)

Im folgenden Verfahren wird das Erstellen eines OpsItem über die AWS Command Line Interface (AWS CLI) beschrieben.

Um einen zu erstellen OpsItem mit dem AWS CLI

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Öffnen Sie den AWS CLI und führen Sie den folgenden Befehl aus, um einen zu erstellenOpsItem. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

   ```
   aws ssm create-ops-item \
       --title "Descriptive_title" \
       --description "Information_about_the_issue" \
       --priority Number_between_1_and_5 \
       --source Source_of_the_issue \
       --operational-data Up_to_20_KB_of_data_or_path_to_JSON_file \
       --notifications Arn="SNS_ARN_in_same_Region" \
       --tags "Key=key_name,Value=a_value"
   ```

   Angabe von Betriebsdaten aus einer Datei

   Wenn Sie ein OpsItem erstellen, können Sie Betriebsdaten aus einer Datei angeben. Die Datei muss eine JSON-Datei sein, und der Inhalt der Datei muss das folgende Format aufweisen.

```
{
  "key_name": {
    "Type": "SearchableString",
    "Value": "Up to 20 KB of data"
  }
}
```

Ein Beispiel.

```
aws ssm create-ops-item ^
    --title "EC2 instance disk full" ^
    --description "Log clean up may have failed which caused the disk to be full" ^
    --priority 2 ^
    --source ec2 ^
    --operational-data file:///Users/TestUser1/Desktop/OpsItems/opsData.json ^
    --notifications Arn="arn:aws:sns:us-west-1:12345678:TestUser1" ^
    --tags "Key=EC2,Value=Production"
```

> **Note**
>
> Informationen zur Eingabe von JSON-formatierten Parametern in der Befehlszeile verschiedener lokaler Betriebssysteme finden Sie unter Verwenden von Anführungszeichen mit Zeichenfolgen in der AWS CLI im AWS Command Line Interface -Benutzerhandbuch.

Das System gibt unter anderem folgende Informationen zurück

```
{
    "OpsItemId": "oi-1a2b3c4d5e6f"
}
```

3. Führen Sie den folgenden Befehl aus, um Details zu dem von Ihnen erstellten OpsItem anzuzeigen.

```
aws ssm get-ops-item --ops-item-id ID
```

Das System gibt unter anderem folgende Informationen zurück

```
{
    "OpsItem": {
        "CreatedBy": "arn:aws:iam::12345678:user/TestUser",
        "CreatedTime": 1558386334.995,
        "Description": "Log clean up may have failed which caused the disk to be
 full",
        "LastModifiedBy": "arn:aws:iam::12345678:user/TestUser",
        "LastModifiedTime": 1558386334.995,
        "Notifications": [
            {
                "Arn": "arn:aws:sns:us-west-1:12345678:TestUser"
            }
        ],
        "Priority": 2,
        "RelatedOpsItems": [],
        "Status": "Open",
        "OpsItemId": "oi-1a2b3c4d5e6f",
        "Title": "EC2 instance disk full",
        "Source": "ec2",
        "OperationalData": {
            "EC2": {
                "Value": "12345",
                "Type": "SearchableString"
            }
        }
    }
}
```

4. Führen Sie den folgenden Befehl aus, um das OpsItem zu aktualisieren. Dieser Befehl ändert den Status von `Open` (Standardwert) zu `InProgress`.

```
aws ssm update-ops-item --ops-item-id ID --status InProgress
```

   Der Befehl hat keine Ausgabe.

5. Führen Sie den folgenden Befehl erneut aus, um zu überprüfen, ob der Status zu `InProgress` geändert wurde.

```
aws ssm get-ops-item --ops-item-id ID
```

Beispiele für die Erstellung eines OpsItem

Die folgenden Beispiele zeigen, wie Sie ein OpsItem mit dem Linux-Verwaltungsportal, macOS oder Windows Server erstellen und verwalten.

Linux-Verwaltungsportal oder macOS

Der folgende Befehl erstellt eineOpsItem, wenn eine Amazon Elastic Compute Cloud (Amazon EC2) - Instance-Festplatte voll ist.

```
aws ssm create-ops-item \
    --title "EC2 instance disk full" \
    --description "Log clean up may have failed which caused the disk to be full" \
    --priority 2 \
    --source ec2 \
    --operational-data '{"EC2":{"Value":"12345","Type":"SearchableString"}}' \
    --notifications Arn="arn:aws:sns:us-west-1:12345678:TestUser1" \
    --tags "Key=EC2,Value=ProductionServers"
```

Der folgende Befehl verwendet den `/aws/resources`-Schlüssel in `OperationalData`, um ein OpsItem mit einer Amazon-DynamoDB-bezogenen Ressource zu erstellen.

```
aws ssm create-ops-item \
    --title "EC2 instance disk full" \
    --description "Log clean up may have failed which caused the disk to be full" \
    --priority 2 \
    --source ec2 \
    --operational-data '{"/aws/resources":{"Value":"[{\"arn\": \"arn:aws:dynamodb:us-
west-2:12345678:table/OpsItems\"}]","Type":"SearchableString"}}' \
    --notifications Arn="arn:aws:sns:us-west-2:12345678:TestUser"
```

Der folgende Befehl verwendet den `/aws/automations`-Schlüssel in `OperationalData`, um ein OpsItem zu erstellen, das das `AWS-ASGEnterStandby`-Dokument als zugeordnetes Automation-Runbook angibt.

```
aws ssm create-ops-item \
    --title "EC2 instance disk full" \
    --description "Log clean up may have failed which caused the disk to be full" \
    --priority 2 \
    --source ec2 \
```

```
    --operational-data '{"/aws/automations":{"Value":"[{\"automationId
\": \"AWS-ASGEnterStandby\", \"automationType\": \"AWS::SSM::Automation
\"}]","Type":"SearchableString"}}' \
    --notifications Arn="arn:aws:sns:us-west-2:12345678:TestUser"
```

Windows

Der folgende Befehl erstellt ein OpsItem wenn eine Amazon Relational Database Service (Amazon RDS)-Instance nicht reagiert.

```
aws ssm create-ops-item ^
    --title "RDS instance not responding" ^
    --description "RDS instance not responding to ping" ^
    --priority 1 ^
    --source RDS ^
    --operational-data={\"RDS\":{\"Value\":\"abcd\",\"Type\":\"SearchableString\"}} ^
    --notifications Arn="arn:aws:sns:us-west-1:12345678:TestUser1" ^
    --tags "Key=RDS,Value=ProductionServers"
```

Der folgende Befehl verwendet den `/aws/resources` Schlüssel in`OperationalData`, um eine OpsItem mit einer EC2 Amazon-Instance verknüpfte Ressource zu erstellen.

```
aws ssm create-ops-item ^
    --title "EC2 instance disk full" ^
    --description "Log clean up may have failed which caused the disk to be full" ^
    --priority 2 ^
    --source ec2 ^
    --operational-data={\"/aws/resources\":{\"Value\":\"[{\\\"""arn\\""":\
\"""arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0\\"""}]\",\"Type\":
\"SearchableString\"}}
```

Der folgende Befehl verwendet den `/aws/automations`-Schlüssel in `OperationalData`, um ein OpsItem zu erstellen, das das `AWS-RestartEC2Instance`-Runbook als zugeordnetes Automation-Runbook angibt.

```
aws ssm create-ops-item ^
    --title "EC2 instance disk full" ^
    --description "Log clean up may have failed which caused the disk to be full" ^
    --priority 2 ^
    --source ec2 ^
```

```
    --operational-data={\"/aws/automations\":{\"Value\":\"[{\\"""automationId\\""":
\\"""AWS-RestartEC2Instance\\”"",\\"""automationType\\""":\\"""AWS::SSM::Automation\
\"""}]\",\"Type\":\"SearchableString\"}}
```

Manuelles Erstellen von OpsItems (PowerShell)

Im folgenden Verfahren wird das Erstellen eines OpsItem über AWS Tools for Windows PowerShell beschrieben.

Um eine OpsItem zu erstellen AWS Tools for Windows PowerShell

1. Öffnen Sie den folgenden Befehl AWS Tools for Windows PowerShell und führen Sie ihn aus, um Ihre Anmeldeinformationen anzugeben.

   ```
   Set-AWSCredentials –AccessKey key-name –SecretKey key-name
   ```

2. Führen Sie den folgenden Befehl aus, um den AWS-Region für Ihre PowerShell Sitzung festzulegen.

   ```
   Set-DefaultAWSRegion -Region Region
   ```

3. Führen Sie den folgenden Befehl aus, um ein neues OpsItem zu erstellen. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen. Dieser Befehl gibt ein Systems Manager Automation-Runbook zum Beheben dieses OpsItem an.

   ```
   $opsItem = New-Object Amazon.SimpleSystemsManagement.Model.OpsItemDataValue
   $opsItem.Type = [Amazon.SimpleSystemsManagement.OpsItemDataType]::SearchableString
   $opsItem.Value = '[{\"automationId\":\"runbook_name\",\"automationType\":
   \"AWS::SSM::Automation\"}]'
   $newHash = @{" /aws/
   automations"=[Amazon.SimpleSystemsManagement.Model.OpsItemDataValue]$opsItem}

   New-SSMOpsItem `
       -Title "title" `
       -Description "description" `
       -Priority priority_number `
       -Source AWS_service `
       -OperationalData $newHash
   ```

   Im Erfolgsfall gibt der Befehl die ID des neuen OpsItem aus.

Das folgende Beispiel gibt den Amazon-Ressourcennamen (ARN) einer beeinträchtigten Amazon Elastic Compute Cloud (Amazon EC2) -Instance an.

```
$opsItem = New-Object Amazon.SimpleSystemsManagement.Model.OpsItemDataValue
$opsItem.Type = [Amazon.SimpleSystemsManagement.OpsItemDataType]::SearchableString
$opsItem.Value = '[{\"arn\":\"arn:aws:ec2:us-east-1:123456789012:instance/
i-1234567890abcdef0\"}]'
$newHash = @{" /aws/
resources"=[Amazon.SimpleSystemsManagement.Model.OpsItemDataValue]$opsItem}
New-SSMOpsItem -Title "EC2 instance disk full still" -Description "Log clean up may
 have failed which caused the disk to be full" -Priority 2 -Source ec2 -OperationalData
 $newHash
```

Verwalten von OpsItems

OpsCenter, ein Tool in AWS Systems Manager, verfolgt OpsItems von ihrer Erstellung bis zur Auflösung. Wenn Sie eine kontoübergreifende Verwaltung einrichten für OpsCenter, kann ein delegierter Administrator oder ein Verwaltungskonto OpsItems von seinem Konto aus verwalten. Weitere Informationen finden Sie unter (Optional) Manuelle Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems.

Sie können OpsItems anzeigen und verwalten, indem Sie die folgenden Seiten in der Systems-Manager-Konsole nutzen:

- Zusammenfassung – Zeigt die Anzahl offener und in Bearbeitung befindlicher OpsItems, die Anzahl der OpsItems nach Quelle und Alter sowie betriebliche Einblicke an. Sie könnenOpsItems nach Quelle und OpsItems-Status filtern.

- OpsItems – Zeigt eine Liste von OpsItems mit mehreren Informationsfeldern an, z. B. Titel, ID, Priorität, Beschreibung, Quelle des OpsItem, sowie Datum und Uhrzeit der letzten Aktualisierung. Auf dieser Seite können Sie OpsItems manuell erstellen, Quellen konfigurieren, den Status eines OpsItem ändern und OpsItems nach neuen Vorfällen filtern. Sie können ein OpsItem auswählen, um dessen OpsItems-Detail-Seite anzuzeigen.

- OpsItem-Details – Bietet detaillierte Einblicke und Tools, mit denen Sie ein OpsItem verwalten können. Die OpsItems-Detailseite hat die folgenden Tabs:
  - Übersicht – Zeigt zugehörige Ressourcen, Runbooks, die in den letzten 30 Tagen ausgeführt wurden, und eine Liste der verfügbaren Runbooks an, die Sie ausführen können. Sie können

auch ähnliche OpsItems anzeigen, Betriebsdaten hinzufügen und verbundene OpsItems hinzufügen.

- Details zu verwandten Ressourcen — Zeigt Informationen über die Ressource aus verschiedenen AWS Diensten an. Erweitern Sie den Abschnitt Ressourcendetails, um Informationen zu dieser Ressource anzuzeigen, die von dem AWS Dienst bereitgestellt werden, der sie hostet. Sie können auch durch andere verwandte Ressourcen schalten, die mit diesem OpsItem verknüpft sind, indem Sie die Liste Related resources (verwandte Ressourcen) verwenden.

Weitere Informationen über die Verwaltung von OpsItems finden Sie in den folgenden Themen.

Themen

- Anzeigen von Details zu einem OpsItem
- Bearbeitung eines OpsItem
- Hinzufügen zugehöriger Ressourcen zu einem OpsItem
- Hinzufügen zugehöriger OpsItems zu einem OpsItem
- Hinzufügen von Betriebsdaten in ein OpsItem
- Erstellen eines Vorfalls für ein OpsItem
- Verwalten von OpsItems-Duplikaten
- Analyse betrieblicher Einblicke zur Reduzierung von OpsItems
- Anzeigen von OpsCenter-Protokollen und Berichten

Anzeigen von Details zu einem OpsItem

Um einen umfassenden Überblick über ein OpsItem zu erhalten, verwenden Sie die Seite OpsItem-Details in der OpsCenter-Konsole. Die Seite Übersicht zeigt die folgenden Informationen an:

- OpsItems-Details – Zeigt allgemeine Informationen für das ausgewählte OpsItem an.
- Zugehörige Ressourcen – Bei einer zugehörigen Ressource handelt es sich um die betroffene Ressource oder die Ressource, die das Ereignis ausgelöst hat, die das OpsItem erstellt hat.
- Automatisierungsausführungen in den letzten 30 Tagen – Eine Liste der Runbooks, die in den letzten 30 Tagen ausgeführt wurden.
- Runbooks – Sie können ein Runbook aus einer Liste verfügbarer Runbooks auswählen.

- Ähnlich OpsItems — Dies ist eine vom System generierte Liste von DingenOpsItems, die für Sie relevant oder von Interesse sein könnten. Zum Generieren der Liste scannt das System die Titel und Beschreibungen aller OpsItems und gibt OpsItems, die ähnliche Wörter verwenden, zurück.
- Betriebsdaten – Bei Betriebsdaten handelt es sich um benutzerdefinierte Daten, die nützliche Details über das OpsItem bereitstellen. Sie können beispielsweise Protokolldateien, Fehlerzeichenfolgen, Lizenzschlüssel, Tipps zur Fehlerbehebung oder andere relevante Daten angeben.
- OpsItemsVerwandt — Sie können angebenOpsItems, welche in irgendeiner Weise mit IDs der aktuellen Version verwandt sind. OpsItem
- Verwandte Ressourcendetails — Zeigt Datenanbieter an, darunter CloudWatch Amazon-Metriken und -Alarme, AWS CloudTrail Protokolle und Details von AWS Config.

So zeigen Sie Details zu einer OpsItem an

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie ein OpsItem aus, um dessen Details anzuzeigen.

Bearbeitung eines OpsItem

Der Abschnitt OpsItem-Details enthält Informationen über ein OpsItem, einschließlich der Beschreibung, des Titels, der Quelle, der OpsItem-ID und des Status.
Sie können OpsItem einzeln bearbeiten oder mehrere OpsItems auswählen und eines der folgenden Felder bearbeiten: Status, Priorität, Schweregrad, Kategorie.

Wenn Amazon eine EventBridge erstelltOpsItem, füllt es die Felder Titel, Quelle und Beschreibung aus. Sie können die Felder Titel und Beschreibung bearbeiten, jedoch nicht das Feld Quelle.

> **Note**
>
> Die Konsole unterstützt die meisten Markdown-Formatierungen im OpsItem Beschreibungsfeld. Weitere Informationen finden Sie unter Verwenden von Markdown in der Konsole im Handbuch Erste Schritte mit dem Handbuch AWS-Managementkonsole Erste Schritte.

Im Allgemeinen können Sie die folgenden konfigurierbaren Daten für ein OpsItem bearbeiten:

- Titel – Name des OpsItem. Die Quelle erstellt den Titel des OpsItem.
- Beschreibung – Informationen zu OpsItem, einschließlich (falls zutreffend) Schritte zur Reproduktion des Problems.
- Status – Status eines OpsItem. Eine Liste der gültigen Statuswerte finden Sie unter OpsItem-Status in der AWS Systems Manager -API-Referenz.
- Priorität – Die Priorität eines OpsItem kann zwischen 1 und 5 liegen. Wir empfehlen, dass Ihre Organisation festlegt, was jede Prioritätsstufe bedeutet, und eine entsprechende Service-Level-Vereinbarung für jede Stufe erstellt.
- Schweregrad – Der Schweregrad eines OpsItems kann zwischen 1 und 4 liegen, wobei 1 für kritisch, 2 für hoch, 3 für mittel und 4 für niedrig steht.
- Kategorie – Die Kategorie eines OpsItems kann Verfügbarkeit, Kosten, Leistung, Wiederherstellung oder Sicherheit sein.
- Benachrichtigungen – Wenn Sie ein OpsItem bearbeiten, können Sie den Amazon-Ressourcennamen (ARN) eines Themas von Amazon Simple Notification Service im Feld Benachrichtigungen angeben. Indem Sie einen ARN angeben, stellen Sie sicher, dass alle Beteiligte eine Benachrichtigung erhalten, wenn das OpsItem bearbeitet wird, z. B. eine Statusänderung. Weitere Informationen finden Sie im Amazon Simple Notification Service-Entwicklerhandbuch.

> **Important**
>
> Das Amazon SNS SNS-Thema muss genauso existieren AWS-Region wie dasOpsItem. Wenn das Thema und das OpsItem sich in verschiedenen Regionen befinden, wird vom System ein Fehler zurückgegeben.

OpsCenterhat eine bidirektionale Integration mit. AWS Security Hub CSPM Wenn Sie einen OpsItem Status und einen Schweregrad im Zusammenhang mit einer Sicherheitsfeststellung aktualisieren, werden diese Änderungen automatisch an Security Hub CSPM gesendet, um sicherzustellen, dass Sie immer die neuesten und korrekten Informationen sehen.

Wenn aus einem Security Hub CSPM-Befund ein erstellt OpsItem wird, werden Security Hub CSPM-Metadaten automatisch zum Betriebsdatenfeld von hinzugefügt. OpsItem Wenn diese Metadaten gelöscht werden, funktionieren die bidirektionalen Updates nicht mehr.

Bearbeiten von OpsItem-Details

1. Öffnen Sie die Konsole unter. AWS Systems Manager https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie eine OpsItem-ID, um die Detailseite zu öffnen oder mehrere OpsItems auszuwählen. Wenn Sie mehrere OpsItems wählen, können Sie nur den Status, die Priorität, den Schweregrad oder die Kategorie bearbeiten. Wenn Sie mehrere OpsItems bearbeiten, aktualisiert und speichert OpsCenter Ihre Änderungen, sobald Sie den neuen Status, die Priorität, den Schweregrad oder die Kategorie ausgewählt haben.
4. Wählen Sie im OpsItem Detailbereich Bearbeiten aus.
5. Bearbeiten Sie die Details des OpsItem entsprechend den Vorschriften und Richtlinien Ihrer Organisation.
6. Wenn Sie fertig sind, wählen Sie Speichern.

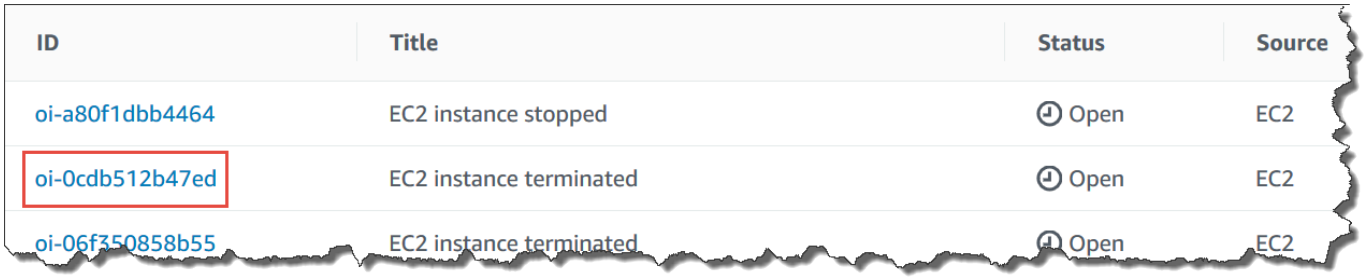
Hinzufügen zugehöriger Ressourcen zu einem OpsItem

Jedes OpsItem enthält einen Abschnitt Zugehörige Ressourcen, der den Amazon-Ressourcennamen (ARN) der zugehörigen Ressource auflistet. Eine verwandte Ressource ist die betroffene AWS Ressource, die untersucht werden muss.

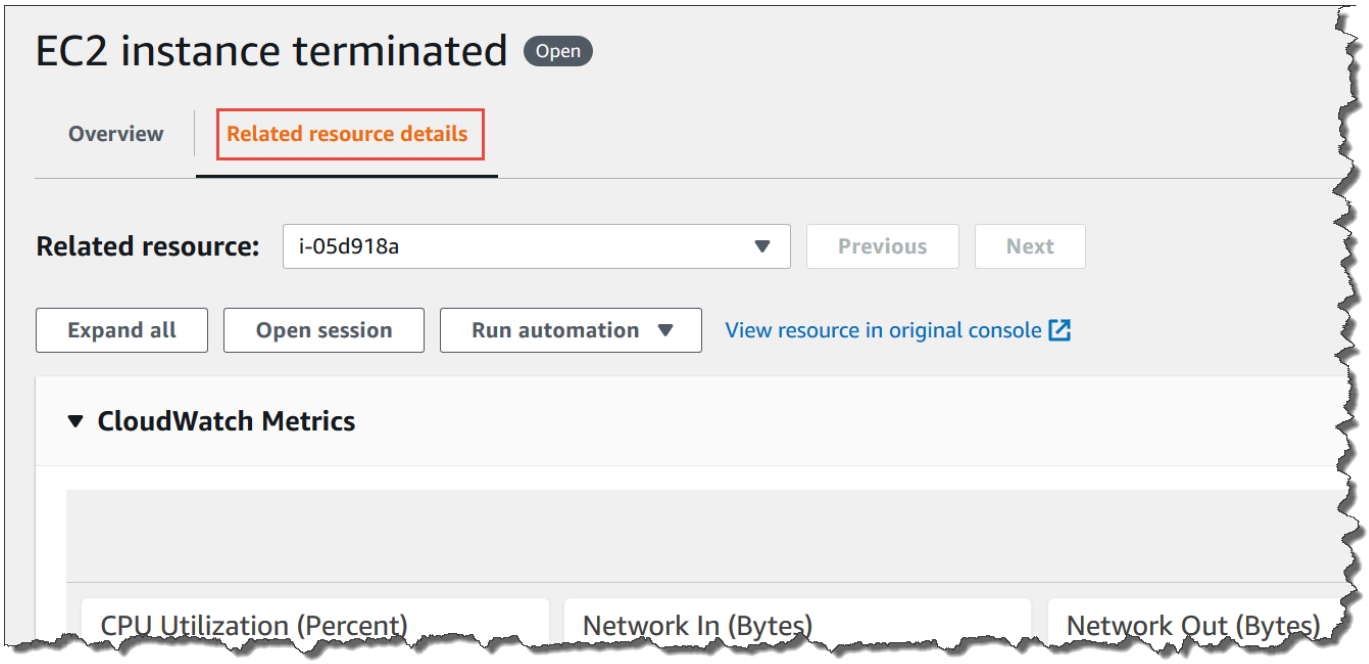
Wenn Amazon den EventBridge erstelltOpsItem, füllt das System den automatisch OpsItem mit dem ARN der Ressource. Sie können die zugehörigen Ressourcen manuell angeben ARNs . Bei einigen ARN-Typen erstellt OpsCenter automatisch einen Deep-Link, der Details zur Ressource direkt in der OpsCenter-Konsole anzeigt. Wenn Sie beispielsweise den ARN einer Amazon Elastic Compute Cloud (Amazon EC2) -Instance als zugehörige Ressource angeben, werden Details zu dieser EC2 Instance abgerufen. OpsCenter Auf diese Weise können Sie detaillierte Informationen zu Ihren betroffenen AWS Ressourcen einsehen, ohne die Website verlassen zu müssen. OpsCenter

So zeigen Sie zugehörige Ressourcen an und fügen sie einem OpsItem hinzu

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie die Registerkarte OpsItems aus.
4. Wählen Sie eine OpsItem-ID aus.



5. Zum Einsehen von Informationen über die betroffene Ressource wählen Sie die Registerkarte Related resources details (Details zugehörige Ressourcen) aus.



Diese Registerkarte zeigt Informationen über die Ressource aus mehreren AWS-Services an. Erweitern Sie den Abschnitt mit den Ressourcendetails, um Informationen zu dieser Ressource anzuzeigen, die von demjenigen bereitgestellt werden AWS-Service , der sie hostet. Sie können auch durch andere verwandte Ressourcen schalten, die mit diesem OpsItem verknüpft sind, indem Sie die Liste Related resources (verwandte Ressourcen) verwenden.

6. Zum Hinzufügen zugehöriger Ressourcen wählen Sie die Registerkarte Overview (Übersicht) aus.
7. Wählen Sie im Abschnitt Related resources (Zugehörige Ressourcen die Option Hinzufügen aus.
8. Wählen Sie für Resource type (Ressourcentyp) eine Ressource aus der Liste aus.
9. Geben Sie für Resource ID entweder die ID oder den Amazon-Ressourcennamen (ARN) ein. Die Art der ausgewählten Informationen hängt von der Ressource ab, die Sie im vorherigen Schritt ausgewählt haben.

> **Note**
>
> Sie können weitere verwandte Ressourcen manuell hinzufügen. ARNs Jede OpsItem kann maximal 100 zugehörige Ressourcen auflisten ARNs.

In der folgenden Tabelle sind die Ressourcentypen aufgeführt, die automatisch Deep-Links zu verwandten Ressourcen erstellen.

Unterstützte Ressourcentypen

Ressourcenname	ARN-Format
AWS Certificate Manager Zertifikat	`arn:aws:acm: region:account-id :certificate/ certificate-id`
Amazon EC2 Auto Scaling Scaling-Gruppe	`arn:aws:autoscaling: region:account-id :autoScalingGroup: groupid:autoScalingGroupName/ groupfriendlyname`
CloudFront Amazon-Vertrieb	`arn:aws:cloudfront:: account-id :*`
AWS CloudFormation stapeln	`arn:aws:cloudformation: region:account-id :stack/stackname /additionalidentifier`
Amazon CloudWatch Alarmanlage	`arn:aws:cloudwatch: region:account-id :alarm:alarm-name`
AWS CloudTrail Spur	`arn:aws:cloudtrail: region:account-id :trail/trailname`
AWS CodeBuild Projekt	`arn:aws:codebuild: region:account-id :resourcetype /resource`

Ressourcenname	ARN-Format
AWS CodePipeline	`arn:aws:codepipeline:  region:account-id :resource-specifier`
Einblick in Amazon DevOps Guru	`arn:aws:devops-guru:  region:account-id :insight/ proactive or reactive/resource-id`
Amazon-DynamoDB-Tabelle.	`arn:aws:dynamodb: region:account-id :table/tablename`
Kunden-Gateway für Amazon Elastic Compute Cloud (Amazon EC2)	`arn:aws:ec2: region:account-id :customer-gateway/  cgw-id`
EC2 Elastisches IP von Amazon	`arn:aws:ec2: region:account-id :eip/eipalloc-id`
EC2 Dedizierter Amazon-Host	`arn:aws:ec2: region:account-id :dedicated-host/ host-id`
EC2 Amazon-Instanz	`arn:aws:ec2: region:account-id :instance/ instance-id`
EC2 Amazon-Internet-Gateway	`arn:aws:ec2: region:account-id :internet-gateway/  igw-id`
EC2 Amazon-Netzwerkzugriffskontrollliste (Netzwerk-ACL)	`arn:aws:ec2: region:account-id :network-acl/ nacl-id`
EC2 Amazon-Netzwerkschnittstelle	`arn:aws:ec2: region:account-id :network-interface/  eni-id`

Ressourcenname	ARN-Format
EC2 Amazon-Routentabelle	`arn:aws:ec2: region:account-id :route-table/ route-table-id`
EC2 Amazon-Sicherheitsgruppe	`arn:aws:ec2: region:account-id :security-group/ security-group-id`
EC2 Amazon-Subnetz	`arn:aws:ec2: region:account-id :subnet/subnet-id`
EC2 Amazon-Volumen	`arn:aws:ec2: region:account-id :volume/volume-id`
Amazon EC2 VPC	`arn:aws:ec2: region:account-id :vpc/vpc-id`
Amazon EC2 VPN-Verbindung	`arn:aws:ec2: region:account-id :vpn-connection/ vpn-id`
Amazon EC2 VPN-Gateway	`arn:aws:ec2: region:account-id :vpn-gateway/ vgw-id`
AWS Elastic Beanstalk Anwendung	`arn:aws:elasticbeanstalk: region:account-id :application/ applicationname`
Elastic Load Balancing (Classic Load Balancer)	`arn:aws:elasticloadbalancing: region:account-id :loadbalancer/ name`

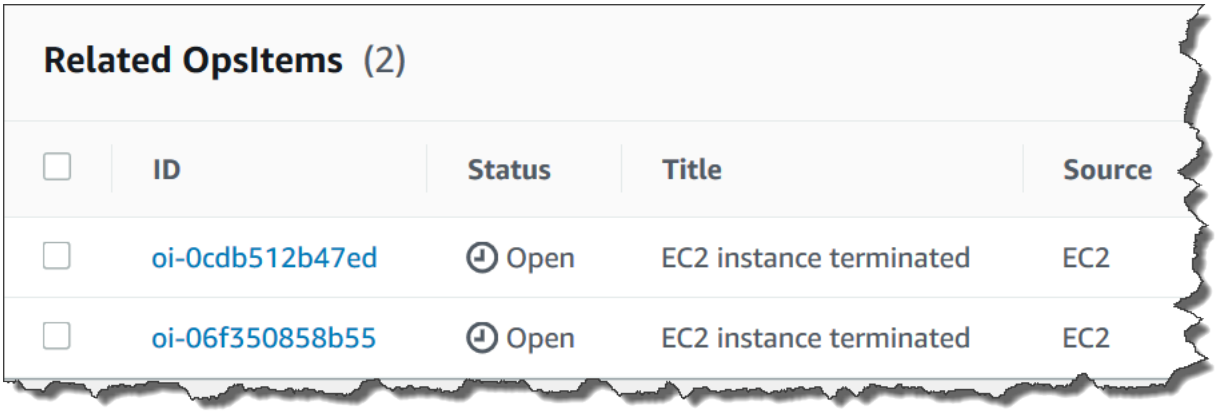
Ressourcenname	ARN-Format
Elastic Load Balancing (Application Load Balancer)	`arn:aws:elasticloadbalancing: region:account-id :loadbalancer/app/ load-balancer-name /load-balancer-id`
Elastic Load Balancing (Network Load Balancer)	`arn:aws:elasticloadbalancing: region:account-id :loadbalancer/net/ load-balancer-name /load-balancer-id`
AWS Identity and Access Management (IAM) - Gruppe	`arn:aws:iam:: account-id :group/group-name`
IAM-Richtlinie	`arn:aws:iam:: account-id :policy/policy-name`
IAM role (IAM-Rolle)	`arn:aws:iam:: account-id :role/role-name`
IAM-Benutzer	`arn:aws:iam:: account-id :user/user-name`
AWS Lambda Funktion	`arn:aws:lambda: region:account-id :function: function-name`
Amazon Relational Database Service (Amazon RDS)-Cluster	`arn:aws:rds: region:account-id :cluster: db-cluster-name`
Amazon-RDS-Datenbank-Instance	`arn:aws:rds: region:account-id :db:db-instance-name`

Ressourcenname	ARN-Format
Amazon RDS-Abonnement	`arn:aws:rds: region:account-id :es:subscription-name`
Amazon RDS-Sicherheitsgruppe	`arn:aws:rds: region:account-id :secgrp:security-group-name`
Amazon RDS-Clusters Snapshot	`arn:aws:rds: region:account-id :cluster-snapshot: cluster-snapshot-name`
Amazon RDS-Subnetzgruppe	`arn:aws:rds: region:account-id :subgrp:subnet-group-name`
Amazon-Redshift-Cluster	`arn:aws:redshift: region:account-id :cluster: cluster-name`
Amazon-Redshift-Parametergruppe	`arn:aws:redshift: region:account-id :parametergroup: parameter-group-name`
Amazon Redshift-Sicherheitsgruppe	`arn:aws:redshift: region:account-id :securitygroup: security-group-name`
Amazon-Redshift-Clsuter-Snapshots	`arn:aws:redshift: region:account-id :snapshot: cluster-name /snapshot-name`
Amazon-Redshift-Subnetzgruppen	`arn:aws:redshift: region:account-id :subnetgroup: subnet-group-name`
Amazon Simple Storage Service (Amazon S3)-Bucket	`arn:aws:s3::: bucket_name`

Ressourcenname	ARN-Format
AWS Config Aufzeichnung des AWS Systems Manager verwalteten Knoteninventars	`arn:aws:ssm: region:account-id :managed-instance-inventory / node_id`
Systems Manager State Manager-Zuordnung.	`arn:aws:ssm: region:account-id :association/ association_ID`

Hinzufügen zugehöriger OpsItems zu einem OpsItem

Mithilfe OpsItems der Seite „Weitere Informationen“ auf der Seite „OpsItemsDetails“ können Sie Betriebsprobleme untersuchen und den Kontext für ein Problem angeben. OpsItemskann auf unterschiedliche Weise miteinander verknüpft werden, z. B. durch eine Eltern-Kind-BeziehungOpsItems, durch eine Grundursache oder durch ein Duplikat. Sie können eines OpsItem mit einem anderen verknüpfen, um es im OpsItem Bereich „Verwandtes“ anzuzeigen. Sie können maximal 10 IDs für andere angeben, die sich auf OpsItems das aktuelle Objekt beziehen. OpsItem



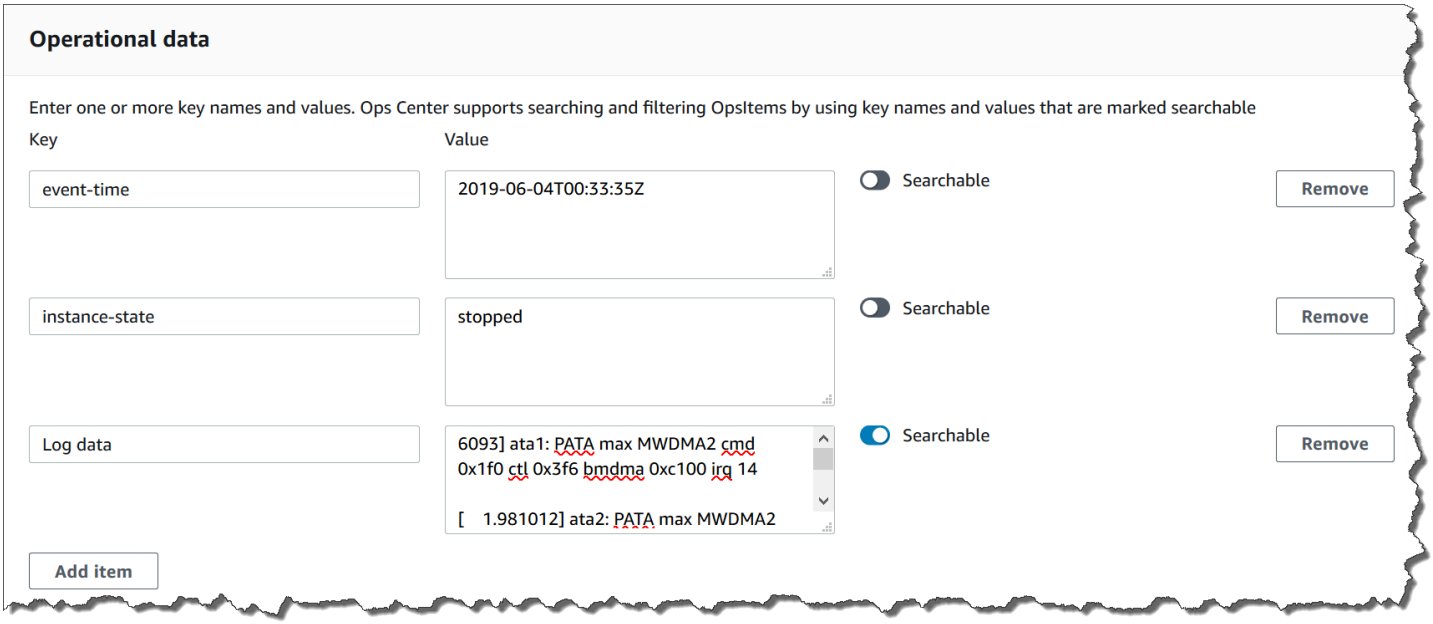
Hinzufügen von verwandten OpsItem

1. Öffnen Sie die Konsole unter AWS Systems Manager . https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie eine OpsItem-ID, um die Detailseite zu öffnen.
4. Wählen Sie im Abschnitt Related (Zugehörige) OpsItem die Option Add (Hinzufügen) aus.
5. Geben Sie unter OpsItem ID eine ID ein.

6. Wählen Sie Hinzufügen aus.

Hinzufügen von Betriebsdaten in ein OpsItem

Betriebsdaten sind benutzerdefinierte Daten, die nützliche Referenzdetails zu einem OpsItem bereitstellen. Sie können mehrere Schlüssel-Wert-Paare von Betriebsdaten eingeben. Sie können beispielsweise Protokolldateien, Fehlerzeichenfolgen, Lizenzschlüssel, Tipps zur Fehlerbehebung oder andere relevante Daten angeben. Die maximale Länge des Schlüssels kann 128 Zeichen und die maximale Größe des Werts 20 KB betragen.



Sie können die Daten für andere Benutzer im Konto durchsuchbar machen oder den Suchzugriff einschränken. Durchsuchbare Daten bedeuten, dass alle Benutzer mit Zugriff auf die OpsItem Übersichts-Seite (wie von der DescribeOpsItems-API-Operation bereitgestellt) die angegebenen Daten anzeigen und durchsuchen können. Betriebsdaten, die nicht durchsuchbar sind, sind nur für Benutzer sichtbar, die Zugriff auf die haben OpsItem (wie durch den GetOpsItemAPI-Vorgang bereitgestellt).

Einfügen von Betriebsdaten in ein OpsItem

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie eine OpsItem-ID aus, um dessen Detailseite zu öffnen.
4. Erweitern Sie Betriebsdaten.

5. Wenn für das OpsItem keine Betriebsdaten vorhanden sind, wählen Sie Hinzufügen aus. Wenn für das OpsItem bereits Betriebsdaten vorhanden sind, wählen Sie Manage (Verwalten) aus.

   Nachdem Sie die Betriebsdaten erstellt haben, können Sie den Schlüssel und den Wert bearbeiten oder zusätzliche Schlüssel-Wert-Paare hinzufügen, indem Sie Manage (Verwalten) auswählen.
6. Geben Sie unter Key (Schlüssel), ein oder mehrere Wörter an, damit Benutzer den Zweck der Daten verstehen.

   > **Important**
   >
   > Betriebsdatenschlüssel können nicht auf diese Weise beginnen: `amazon`, `aws`, `amzn`, `ssm`, `/amazon`, `/aws`, `/amzn`, `/ssm`.
7. Geben Sie unter Value (Wert) die Daten an.
8. Wählen Sie Speichern.

> **Note**
>
> Sie können OpsItems mithilfe des Operators Operational data (Operative Daten) auf der OpsItems-Seite filtern. Wählen Sie im Feld Suche die Option Betriebsdaten aus und geben Sie dann ein Schlüssel-Wert-Paar in JSON ein. Sie müssen das Schlüssel-Wert-Paar im folgenden Format eingeben: `{"key":"key_name","value":"a_value"}`

Erstellen eines Vorfalls für ein OpsItem

Gehen Sie wie folgt vor, um manuell einen Vorfall OpsItem zu erstellen, in AWS Systems Manager Incident Manager dem Sie ihn verfolgen und verwalten können. Dies ist ein Tool in AWS Systems Manager. Ein Vorfall ist jede Art von ungeplanter Unterbrechung oder Beeinträchtigung der Qualität von Services. Weitere Informationen zu Incident Manager finden Sie unter the section called “Integrieren von OpsCenter in anderen AWS-Services”.

Manuelles Erstellen eines Vorfalls für ein OpsItem

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.

3. Wenn Incident Manager ein OpsItem für Sie erstellt hat, wählen Sie es aus und fahren Sie mit Schritt 5 fort. Wenn nicht, wählen Sie Create OpsItem (Erstellen) und füllen Sie das Formular aus. Wenn diese Schaltfläche nicht angezeigt wird, wählen Sie die Registerkarte OpsItems und dann Create OpsItem aus.
4. Wenn Sie ein neues OpsItem erstellt haben, öffnen Sie es.
5. Wählen Sie Start Incident (Vorfall starten).
6. Wählen Für für Reaktionsplan den Incident-Manager-Reaktionsplan aus, den Sie diesem Vorfall zuweisen möchten.
7. (Optional) Geben Sie für Title (Titel) einen aussagekräftigen Namen ein, anhand dessen andere Teammitglieder die Art des Vorfalls verstehen können. Wenn Sie keinen neuen Titel eingeben erstellt OpsCenter das OpsItem und den entsprechenden Vorfall im Incident Manager unter Verwendung des Titels im Antwortplan.
8. (Optional) Wählen Sie für Incident impact eine Auswirkungsstufe für diesen Vorfall aus. Wenn Sie keine Auswirkungsstufe eingeben erstellt OpsCenter das OpsItem und den entsprechenden Vorfall im Incident Manager unter Verwendung der Auswirkungsstufe im Antwortplan.
9. Wählen Sie Starten.

Verwalten von OpsItems-Duplikaten

OpsCenter kann mehrere OpsItems-Duplikate für eine einzelne Quelle von mehreren AWS-Services erhalten. OpsCenter verwendet eine Kombination aus integrierter Logik und konfigurierbaren Deduplizierungszeichenfolgen, um die Erstellung von OpsItems-Duplikaten zu vermeiden. AWS Systems Manager wendet die integrierte Deduplizierungslogik an, wenn der API-Vorgang OpsItem erstellen aufgerufen wird.

AWS Systems Manager verwendet die folgende Deduplizierungslogik:

1. Beim Erstellen des OpsItem erstellt und speichert Systems Manager einen Hash-Wert basierend auf der Deduplizierungszeichenfolge und der Ressource, durch die das OpsItem ausgelöst wurde.
2. Wenn eine weitere Anfrage zur Erstellung eines OpsItem gestellt wird, prüft das System die Deduplizierungszeichenfolge der neuen Anfrage.
3. Wenn ein übereinstimmender Hash-Wert für diese Deduplizierungszeichenfolge vorhanden ist, überprüft Systems Manager den Status des vorhandenen OpsItem. Wenn der Status eines vorhandenen OpsItem offen oder in Bearbeitung ist, wird das OpsItem nicht erstellt. Wenn das vorhandene OpsItem gelöst wird, erstellt Systems Manager ein neues OpsItem.

Nachdem Sie ein OpsItem erstellt haben, können Sie die Deduplizierungszeichenfolgen in diesem OpsItem nicht ändern.

Zur Verwaltung von OpsItems-Duplikaten können Sie wie folgt vorgehen:

- Bearbeiten Sie die Deduplizierungszeichenfolge für eine Amazon-EventBridge-Regel, die auf OpsCenter ausgerichtet ist. Weitere Informationen finden Sie unter Bearbeiten einer Deduplizierungszeichenfolge in einer Standard-EventBridge-Regel.
- Geben Sie eine Deduplizierungszeichenfolge an, wenn Sie ein neues OpsItem manuell erstellen. Weitere Informationen finden Sie unter Angeben einer Deduplizierungszeichenfolge mit der AWS CLI.
- Überprüfen und beheben Sie OpsItems-Duplikate mit betrieblichen Einblicken. Sie können Runbooks verwenden, um OpsItems-Duplikate zu lösen.

  Um Sie bei der Auflösung von OpsItems-Duplikaten zu unterstützen und die Anzahl der von einer Quelle erstellten OpsItems zu reduzieren, stellt Systems-Manager-Automation-Runbooks bereit. Weitere Informationen finden Sie unter Lösen von OpsItems-Duplikaten basierend auf Erkenntnissen.

Bearbeiten einer Deduplizierungszeichenfolge in einer Standard-EventBridge-Regel

Führen Sie die folgenden Schritte aus, um eine Deduplizierungszeichenfolge für eine EventBridge-Regel mit Ziel OpsCenter zu erstellen.

So bearbeiten Sie eine Deduplizierungszeichenfolge für eine EventBridge-Regel

1. Melden Sie sich bei der AWS-Managementkonsole an und öffnen Sie die Amazon EventBridge-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Regeln aus.
3. Wählen Sie eine Regel und anschließend Edit (Bearbeiten) aus.
4. Rufen Sie die Seite Select target(s) (Ziel(e) auswählen) auf.
5. Wählen Sie im Bereich Additional settings (Zusätzliche Einstellungen) die Option Configure input transformer (Eingabetransformator konfigurieren).
6. Suchen Sie im Feld Template (Vorlage) den `"operationalData": { "/aws/dedup"` JSON-Eintrag und die Deduplizierungszeichenfolgen, die Sie bearbeiten möchten.

Der Eintrag der Deduplizierungszeichenfolge in EventBridge-Regeln verwendet das folgenden JSON-Format.

```
"operationalData": { "/aws/dedup": {"type": "SearchableString","value":
 "{\"dedupString\":\"Words the system should use to check for duplicate
 OpsItems\"}"}}
```

Ein Beispiel.

```
"operationalData": { "/aws/dedup": {"type": "SearchableString","value":
 "{\"dedupString\":\"SSMOpsCenter-EBS-volume-performance-issue\"}"}}
```

7. Bearbeiten Sie die Deduplizierungszeichenfolgen und wählen Sie dann Bestätigen aus.
8. Wählen Sie Weiter aus.
9. Wählen Sie Weiter aus.
10. Wählen Sie Regel aktualisieren aus.

Angeben einer Deduplizierungszeichenfolge mit der AWS CLI

Sie können eine Deduplizierungszeichenfolge angeben, wenn Sie ein neues OpsItem mit der AWS Systems Manager-Konsole oder der AWS CLI manuell erstellt haben. Weitere Informationen zum Eingeben von Deduplizierungszeichenfolgen, wenn Sie ein OpsItem manuell in der Konsole erstellen, finden Sie unter Manuelles Erstellen der OpsItems. Wenn Sie die AWS CLI verwenden, können Sie die Deduplizierungszeichenfolge für den `OperationalData`-Parameter eingeben. Die Parameter-Syntax verwendet JSON, wie im folgenden Beispiel gezeigt.

```
--operational-data '{"/aws/dedup":{"Value":"{\"dedupString\": \"Words the system should
 use to check for duplicate OpsItems\"}","Type":"SearchableString"}}'
```

Es folgt ein Beispiel für einen Befehl, mit dem die Deduplizierungszeichenfolge `disk full` angegeben wird.

Linux & macOS

```
aws ssm create-ops-item \
    --title "EC2 instance disk full" \
    --description "Log clean up may have failed which caused the disk to be full" \
```

```
    --priority 1 \
    --source ec2 \
    --operational-data '{"/aws/dedup":{"Value":"{\"dedupString\": \"disk full
\"}","Type":"SearchableString"}}' \
    --tags "Key=EC2,Value=ProductionServers" \
    --notifications Arn="arn:aws:sns:us-west-1:12345678:TestUser"
```

Windows

```
aws ssm create-ops-item ^
    --title "EC2 instance disk full" ^
    --description "Log clean up may have failed which caused the disk to be full" ^
    --priority 1 ^
    --source EC2 ^
    --operational-data={\"/aws/dedup\":{\"Value\":\"{\\\"""dedupString\\""":\\"""disk
 full\\"""}\",\"Type\":\"SearchableString\"}} ^
    --tags "Key=EC2,Value=ProductionServers" --notifications Arn="arn:aws:sns:us-
west-1:12345678:TestUser"
```

Analyse betrieblicher Einblicke zur Reduzierung von OpsItems

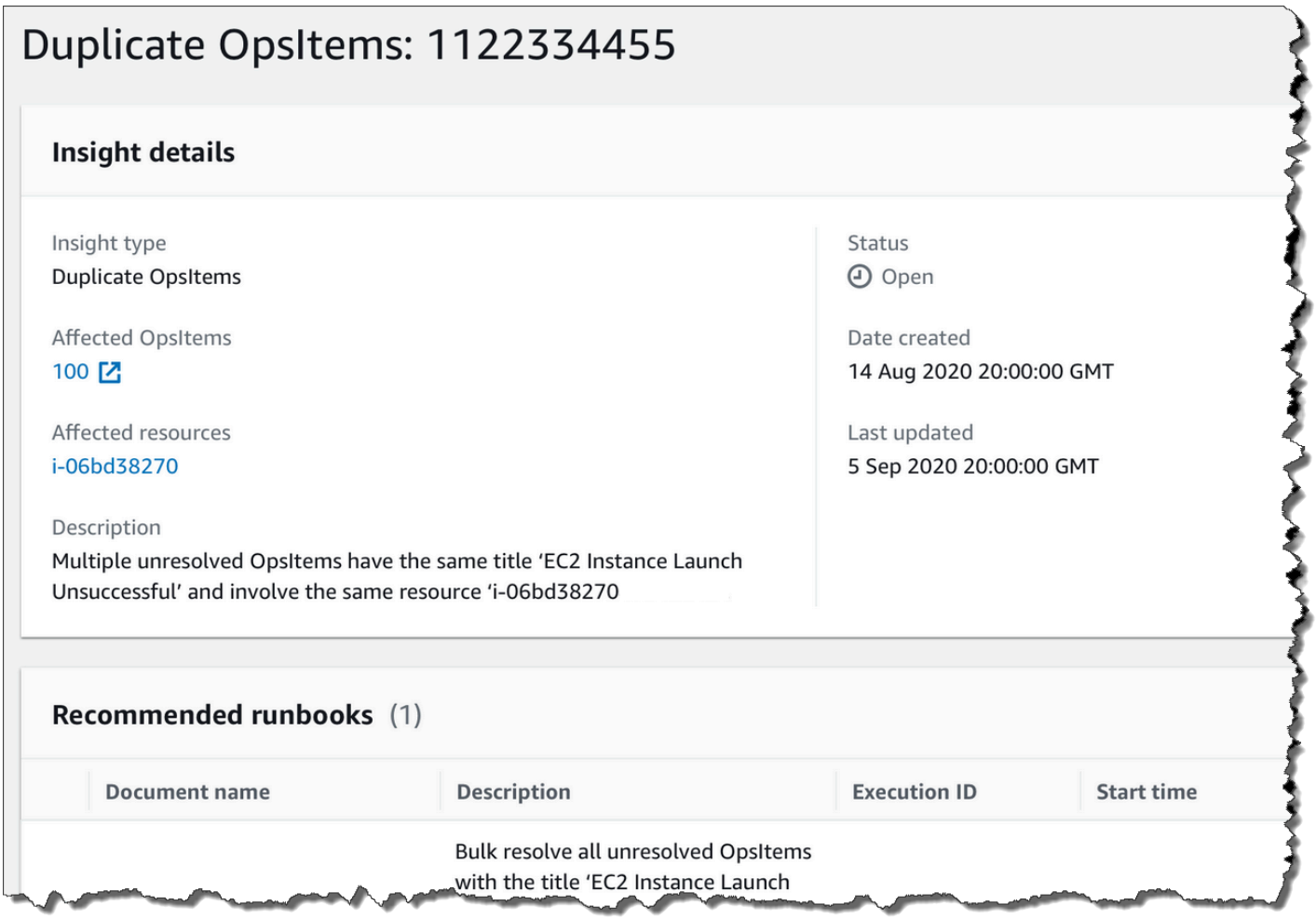
OpsCenter Operational Insights zeigt Informationen zu Duplikaten anOpsItems. OpsCenteranalysiert automatisch OpsItems in Ihrem Konto und generiert drei Arten von Erkenntnissen. Sie können diese Informationen im Abschnitt Betriebliche Einblicke auf der OpsCenter-Registerkarte Zusammenfassung einsehen.

- OpsItems-Duplikate – Ein Einblick wird generiert, wenn acht oder mehr OpsItems denselben Titel für dieselbe Ressource haben.
- Die häufigsten Titel – Ein Einblick wird generiert, wenn mehr als 50 OpsItems denselben Titel haben.
- Ressourcen, die am meisten generieren OpsItems — Ein Einblick wird generiert, wenn für eine AWS Ressource mehr als 10 geöffnet sindOpsItems. Diese Erkenntnisse und die entsprechenden Ressourcen werden in der OpsItems Tabelle Ressourcen mit den meisten Einnahmen auf der Registerkarte OpsCenter Zusammenfassung angezeigt. Die Ressourcen werden in absteigender Reihenfolge ihrer Anzahl an OpsItem aufgeführt.

> **Note**
>
> OpsCenter erstellt Einblicke für Ressourcen, die die meisten OpsItems erzeugen für die folgenden Ressourcentypen:
>
> - Amazon Elastic Compute Cloud (Amazon EC2) -Instanzen
> - EC2 Amazon-Sicherheitsgruppen
> - Amazon EC2 Auto Scaling Scaling-Gruppe
> - Datenbank von Amazon Relational Database Service (Amazon RDS)
> - Amazon-RDS-Cluster
> - AWS Lambda Funktion
> - Amazon-DynamoDB-Tabelle.
> - Load Balancer von Elastic Load Balancing
> - Amazon-Redshift-Cluster
> - AWS Certificate Manager Zertifikat
> - Volume von Amazon Elastic Block Store

OpsCenter erzwingt ein Limit von 15 Einblicken pro Typ. Wenn ein Typ dieses Limit erreicht, zeigt OpsCenter keine weiteren Einblicke für diesen Typ an. Um weitere Erkenntnisse zu erhalten, müssen Sie alle Daten aufklären, die mit einem OpsInsight solchen Typ OpsItems verknüpft sind. Wenn ein ausstehender Einblick aufgrund der Begrenzung auf 15 Einblicke nicht in der Konsole angezeigt werden kann, wird dieser Einblick sichtbar, nachdem ein anderer Einblick geschlossen wurde.

Wenn Sie eine Erkenntnis auswählen, zeigt OpsCenter Informationen über die betroffenen OpsItems und Ressourcen. Der folgende Screenshot zeigt ein Beispiel mit den Details einer Duplikats OpsItem-Erkenntniss.



Betriebliche Einblicke sind standardmäßig nicht aktiviert. Weitere Informationen zur Arbeit mit betrieblichen Einblicken finden Sie in den folgenden Themen.

Themen

- Aktivieren betrieblicher Einblicke
- Lösen von OpsItems-Duplikaten basierend auf Erkenntnissen
- Deaktivieren betrieblicher Erkenntnisse

Aktivieren betrieblicher Einblicke

Sie können betrieblicher Einblicke auf der Seite OpsCenter in der Systems-Manager-Konsole aktivieren. Wenn Sie betriebliche Einblicke aktivieren, erstellt Systems Manager eine neue serviceverknüpfte AWS Identity and Access Management (IAM)-Rolle mit dem Namen `AWSServiceRoleForAmazonSSM_OpsInsights`. Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Systems Manager verknüpft ist. Mit Diensten verknüpfte

Rollen sind vordefiniert und enthalten alle Berechtigungen, die der Dienst benötigt, um andere in AWS-Services Ihrem Namen anzurufen. Weitere Informationen zur serviceverknüpften `AWSServiceRoleForAmazonSSM_OpsInsights`-Rolle finden Sie unter Mithilfe von Rollen betriebliche Einblicke OpsItems in Systems Manager gewinnen OpsCenter.

> **Note**
>
> Beachten Sie die folgenden wichtigen Informationen:
>
> - Operative Einblicke werden Ihnen AWS-Konto in Rechnung gestellt. Weitere Informationen finden Sie unter AWS Systems Manager – Preise.
> - OpsCenter aktualisiert regelmäßig Erkenntnisse mithilfe eines Batch-Prozesses. Dies bedeutet, dass die in OpsCenter angezeigte Liste der Einblicke möglicherweise nicht synchron ist.

Gehen Sie folgendermaßen vor, um betriebliche Einblicke in OpsCenter zu aktivieren und anzuzeigen.

So aktivieren Sie betriebliche Einblicke und zeigen sie an

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie im Meldungsfeld Betrieblicher Einblick ist verfügbar die Option Aktivieren aus. Wenn Sie diese Meldung nicht sehen, scrollen Sie nach unten zum Abschnitt Betriebliche Einblicke und wählen Sie Aktivieren aus.
4. Nachdem Sie dieses Feature aktiviert haben, scrollen Sie auf der Registerkarte Zusammenfassung nach unten zum Abschnitt Betriebliche Einblicke.
5. Um eine gefilterte Liste von Einblicken anzuzeigen, wählen Sie den Link neben OpsItems duplizieren, Häufigste Titel oder Ressourcen, die die meisten OpsItems erzeugen. Um alle Erkenntnisse anzuzeigen, wählen Sie View all operational insights (Alle betrieblichen Erkenntnisse anzeigen aus).
6. Wählen Sie eine Erkenntnis-ID, um weitere Informationen anzuzeigen.

Lösen von OpsItems-Duplikaten basierend auf Erkenntnissen

Um Erkenntnisse zu lösen, müssen Sie zuerst alle mit einem Erkenntnis verbunden OpsItems lösen. Sie können das `AWS-BulkResolveOpsItemsForInsight`-Runbook zum Lösen von OpsItems verwenden, die einem Erkenntnis zugeordnet sind.

Systems Manager bietet die folgenden Automation-Runbooks, die Ihnen dabei helfen, OpsItems-Duplikate aufzulösen und die Anzahl der von einer Quelle erstellten OpsItems zu reduzieren:

- Das `AWS-BulkResolveOpsItems`-Runbook löst OpsItems, die mit einem angegebenen Filter übereinstimmen.
- Das `AWS-AddOpsItemDedupStringToEventBridgeRule` Runbook fügt eine Deduplizierungszeichenfolge für alle OpsItem Ziele hinzu, die einer bestimmten Amazon-Regel zugeordnet sind. EventBridge Dieses Runbook fügt keine Deduplizierungszeichenfolge hinzu, wenn eine Regel bereits über eine verfügt.
- Das `AWS-DisableEventBridgeRule` deaktiviert eine Regel, EventBridge wenn die Regel Dutzende oder Hunderte von generiert. OpsItems

Um einen operativen Einblick zu lösen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Scrollen Sie auf der Registerkarte Overview (Übersicht) bis zu Operational insights (Betriebliche Erkenntnisse) runter.
4. Wählen Sie Alle betrieblichen Einblicke anzeigen aus.
5. Wählen Sie eine Erkenntnis-ID, um weitere Informationen anzuzeigen.
6. Wählen Sie ein Runbook, und klicken Sie anschließend auf Ausführen.

Deaktivieren betrieblicher Erkenntnisse

Wenn Sie betriebliche Einblicke deaktivieren, stoppt das System die Erstellung neuer Einblicke und zeigt keine Einblicke in der Konsole an. Alle aktiven Einblicke verbleiben unverändert im System, obwohl sie in der Konsole nicht angezeigt werden. Wenn Sie dieses Feature erneut aktivieren, zeigt das System alle bisher nicht gelösten Einblicke an und beginnt mit der Erstellung neuer Einblicke. Verwenden Sie die folgende Vorgehensweise, um betriebliche Einblicke zu deaktivieren.

So deaktivieren Sie betriebliche Einblicke

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie Einstellungen aus.
4. Wählen Sie im Abschnitt Operational insights (Betriebliche Erkenntnisse) die Option Edit (Bearbeiten) und schalten Sie dann die Option Disable (deaktivieren) ein.
5. Wählen Sie Speichern.

Anzeigen von OpsCenter-Protokollen und Berichten

AWS CloudTrail protokolliert AWS Systems Manager-OpsCenter-API-Aufrufe an die Konsole, die AWS Command Line Interface (AWS CLI) und das SDK. Sie können die Informationen in der CloudTrail-Konsole oder in einem Amazon Simple Storage Service (Amazon S3)-Bucket anzeigen. Amazon S3 verwendet einen Bucket, um alle CloudTrail-Protokolle für Ihr Konto zu speichern.

Protokolle von OpsCenter-Aktionen zeigen die OpsItem-Aktivitäten „erstellen“, „aktualisieren“, „abrufen“ und „beschreiben“ an. Weitere Informationen zum Anzeigen und Verwenden von CloudTrail-Protokollen von Systems Manager-Aktivitäten finden Sie unter Protokollierung von AWS Systems Manager-API-Aufrufen mit AWS CloudTrail.

AWS Systems Manager OpsCenter liefert Ihnen die folgenden Informationen über OpsItems:

- OpsItem-Statuszusammenfassung – eine Zusammenfassung der OpsItems nach Status (Offen und in Bearbeitung, Offen oder In Bearbeitung).
- Quellen mit den meisten geöffneten OpsItems – Stellt eine Aufstellung der wichtigsten AWS-Services mit offenen OpsItems bereit.
- OpsItems nach Quelle und Alter – eine Anzahl von OpsItems, gruppiert nach Quelle und Tagen seit der Erstellung.

So zeigen Sie die Zusammenfassung des OpsCenter-Berichts an

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.

3. Wählen Sie auf der Seite OpsItems-Übersicht die Option Zusammenfassung aus.
4. Wählen Sie unter OpsItems by source and age (OpsItems nach Quelle und Alter) die Suchleiste zum Filtern von OpsItems nach Source (Quelle). Filtern Sie die Liste nach Status.

Löschen Sie OpsItems

Sie können eine Person löschen, OpsItem indem Sie den Vorgang „OpsItemAPI löschen“ mit dem AWS Command Line Interface oder dem AWS SDK aufrufen. Sie können ein OpsItem nicht in der AWS-Managementkonsole löschen. Um eine zu löschenOpsItem, muss Ihr AWS Identity and Access Management (IAM-) Benutzer, Ihre Gruppe oder Rolle entweder über Administratorrechte verfügen oder Ihnen muss die Berechtigung zum Aufrufen des `DeleteOpsItem` API-Vorgangs erteilt worden sein.

Important

Beachten Sie die folgenden wichtigen Informationen zu dieser Operation.

- Das Löschen eines OpsItem lässt sich nicht rückgängig machen. Sie können ein gelöschtes OpsItem nicht wiederherstellen.
- Bei dieser Operation wird ein Konsistenzmodell verwendet, was bedeutet, dass die Operation einige Minuten in Anspruch nehmen kann. Wenn Sie ein OpsItem löschen und sofort aufrufen, z. B. mit OpsItem abrufen, wird das gelöschte OpsItem möglicherweise immer noch in der Antwort angezeigt.
- Dieser Vorgang ist idempotent. Das System löst keine Ausnahme aus, wenn Sie diese Operation wiederholt für dasselbe OpsItem aufrufen. Wenn der erste Aufruf erfolgreich ist, geben alle weiteren Aufrufe die gleiche Antwort wie der erste Aufruf.
- Diese Operation unterstützt keine kontenübergreifenden Aufrufe. Ein delegierter Administrator oder ein Verwaltungskonto kann keine OpsItems in anderen Konten löschen, selbst wenn das OpsCenter für die kontenübergreifende Verwaltung eingerichtet wurde. Weitere Informationen zur kontoübergreifenden Verwaltung finden Sie unter (Optional) Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems.
- Wenn Sie die `OpsItemLimitExceededException` erhalten, können Sie eines oder mehrere OpsItems löschen, um die Gesamtzahl der OpsItems zu verringern und das Kontingent nicht zu überschreiten. Weitere Informationen zur dieser Ausnahme finden Sie unter Beheben von Problemen mit OpsCenter.

Löschen eines OpsItem

Gehen Sie wie folgt vor, um ein OpsItem zu löschen.

So löschen Sie einen OpsItem

1. Installieren und konfigurieren Sie den AWS CLI, falls Sie dies noch nicht getan haben. Weitere Informationen finden Sie unter Installieren oder Aktualisierung auf die neueste Version von AWS CLI.
2. Führen Sie den folgenden Befehl aus. *ID*Ersetzen Sie es durch die ID der DateiOpsItem, die Sie löschen möchten.

```
aws ssm delete-ops-item --ops-item-id ID
```

Wenn der Befehl erfolgreich ist, werden keine Daten zurückgegeben.

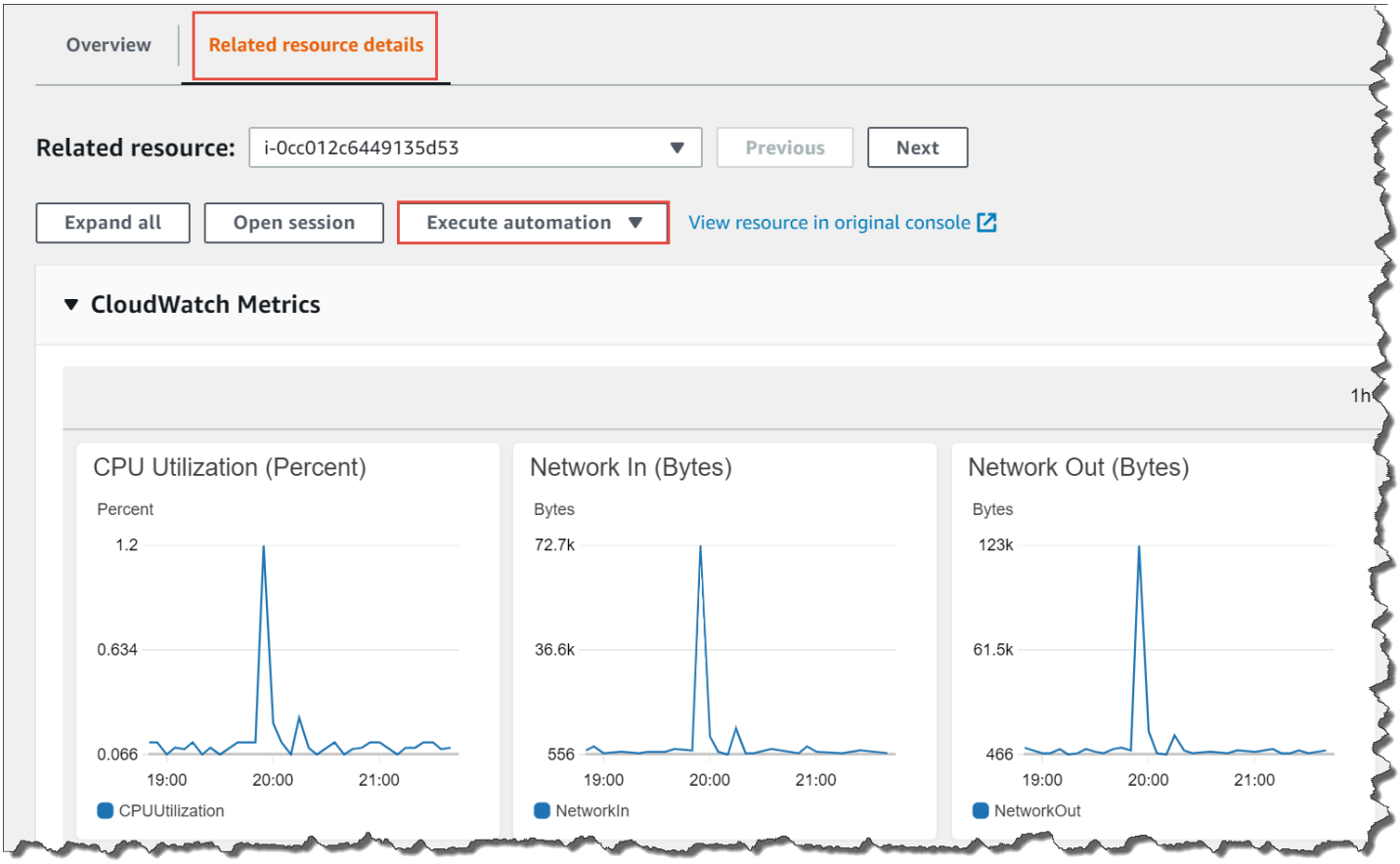
Beheben von OpsItem-Problemen

Mit AWS Systems Manager-Automation-Runbooks können Sie Probleme mit AWS-Ressourcen beheben, die in einem OpsItem identifiziert werden. Automation verwendet vordefinierte-Runbooks, um allgemeine Probleme von AWS-Ressourcen zu beheben.

Jedes OpsItem enthält den Abschnitt Runbooks, der eine Liste von Runbooks enthält, die Sie zur Problembehebung verwenden können. Wenn Sie ein Automation-Runbook aus der Liste auswählen, zeigt OpsCenter automatisch einige der Felder an, die zum Ausführen des Dokuments erforderlich sind. Wenn Sie ein Automation-Runbook ausführen, verknüpft das System das Runbook mit der zugehörigen Ressource von OpsItem. Wenn Amazon EventBridge ein OpsItem erstellt, ordnet es ein Runbook dem OpsItem zu. OpsCenter speichert eine 30-tägige Aufzeichnung von Automation-Runbooks für ein OpsItem.

Sie können einen Status auswählen, um wichtige Details zum Runbook anzuzeigen, z. B. den Grund, warum eine Automatisierung fehlgeschlagen ist, und welcher Schritt des Automation-Runbooks ausgeführt wurde, als der Fehler auftrat, wie im folgenden Beispiel gezeigt.

Die Seite Related resource details (Details zu verwandten Ressourcen) für ein ausgewähltes OpsItemschließt die Liste Run automation mit ein. Sie können aktuelle oder ressourcenspezifische Automation-Runbooks auswählen und ausführen, um Probleme zu beheben. Diese Seite enthält auch Datenanbieter, einschließlich Amazon-CloudWatch-Metriken und -Warnungen, AWS CloudTrail-Protokolle und Details aus AWS Config, um nur einige zu nennen.



Sie können Informationen zu einem Automation-Runbook einsehen, indem Sie entweder den Namen in der Konsole oder mithilfe von Runbook-Referenz für Systems Manager Automation auswählen.

Korrigieren eines OpsItem mit einem Runbook

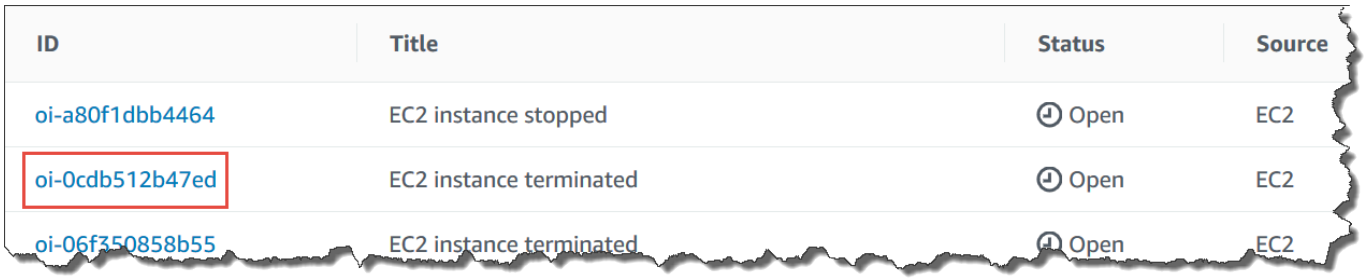
Bevor Sie ein Automation-Runbook verwenden, um ein OpsItem-Problem zu beheben, gehen Sie wie folgt vor:

- Überprüfen Sie, ob Sie über die Berechtigung zur Ausführung von Systems Manager Automation-Runbooks verfügen. Weitere Informationen finden Sie unter Einrichten der Automatisierung.
- Erfassen Sie Ressourcen-spezifische ID-Informationen für die Automatisierung, die Sie ausführen möchten. Beispiel: Wenn Sie eine Automatisierung ausführen möchten, die eine EC2-Instance erneut startet, müssen Sie die ID der neu zu startenden EC2-Instance angeben.

Ausführen eines Automation-Runbook, um ein OpsItem-Problem zu beheben

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie die OpsItem-ID, um die Detailseite zu öffnen.



4. Scrollen Sie zum Abschnitt Runbooks.
5. Verwenden Sie die Suchleiste oder die Zahlen oben rechts, um das Automation-Runbook zu finden, das Sie ausführen möchten.
6. Wählen Sie ein Runbook, und klicken Sie anschließend auf Execute (Ausführen).
7. Geben Sie die erforderlichen Informationen für das Runbook ein und klicken Sie anschließend auf Senden.

   Sobald Sie das Runbook gestartet haben, kehrt das System zum vorherigen Bildschirm zurück und zeigt den Status an.
8. Wählen Sie im Abschnitt Automatisierungsausführungen in den letzten 30 Tagen den Link Ausführungs-ID, um die einzelnen Schritte und den Status der Ausführung anzuzeigen.

Beheben eines OpsItem mithilfe eines zugeordneten Runbooks

Nachdem Sie ein Automation-Runbook von einem OpsItem ausgeführt haben, ordnet OpsCenter das Runbook dem OpsItem zu. Ein zugeordnetes Runbook wird in der Runbooks-Liste höher eingestuft als andere Runbooks.

Gehen Sie wie folgt vor, um ein Automation-Runbook auszuführen, das bereits mit einer zugehörigen Ressource in einem OpsItem verknüpft ist. Weitere Informationen zum Hinzufügen von zugehörigen Ressourcen finden Sie unter Verwalten von OpsItems.

Ausühren eines mit einer Ressource verknüpften Runbook zum Beheben eines OpsItem-Problems

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Öffnen Sie die OpsItem.

4. Wählen Sie im Abschnitt Related resources (Zugehörige Ressourcen) die Ressource aus, auf der Sie das Automaion-Runbook ausführen möchten.
5. Wählen Sie Run automation (Automatisierung ausführen) aus und anschließend das zugehörige Automation-Runbook, das Sie ausführen möchten.
6. Geben Sie die erforderlichen Informationen für das Runbook ein und klicken Sie anschließend auf Execute (Ausführen).

   Sobald Sie das Runbook gestartet haben, kehrt das System zum vorherigen Bildschirm zurück und zeigt den Status an.
7. Wählen Sie im Abschnitt Automatisierungsausführungen in den letzten 30 Tagen den Link Ausführungs-ID, um die einzelnen Schritte und den Status der Ausführung anzuzeigen.

Anzeigen von OpsCenter-Zusammenfassungsberichten

AWS Systems Manager OpsCenter enthält eine Zusammenfassungsseite, welche die folgenden Informationen automatisch anzeigt:

- OpsItem-Statusübersicht – Eine Zusammenfassung von OpsItems nach Status, z. B. `Open` und `In progress`.
- Quellen mit den am meisten geöffneten OpsItems – eine Aufstellung der wichtigsten AWS-Services mit offenen OpsItems.
- OpsItems nach Quelle und Alter – eine Anzahl von OpsItems, gruppiert nach Quelle und Tagen seit der Erstellung.

Um OpsCenter-Zusammenfassungsberichte anzuzeigen

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich OpsCenter und dann die Registerkarte Zusammenfassung aus.
3. Gehen Sie im Abschnitt OpsItems nach Quelle und Alter wie folgt vor:

   1. (Optional) Wählen Sie im Filterfeld Quelle, wählen Sie `Equal`, `Begin With` oder `Not Equal` aus, und geben Sie dann einen Suchparameter ein.
   2. Wählen Sie in der nebenstehenden Liste einen der folgenden Statuswerte aus:
      - `Open`

- `In progress`
- `Resolved`
- `Open and in progress`
- `All`

Beheben von Problemen mit OpsCenter

Hier finden Sie Informationen für die Behebung häufiger Fehler und Probleme mit OpsCenter.

Sie erhalten die OpsItemLimitExceededException

Wenn Ihr beim Aufrufen des CreateOpsItem API-Vorgangs die OpsItems zulässige Höchstzahl von erreicht AWS-Konto hat, erhalten Sie eine`OpsItemLimitExceededException`. OpsCentergibt die Ausnahme zurück, wenn Ihr Aufruf die maximale Anzahl von OpsItems für eines der folgenden Kontingente überschreiten würde:

- Gesamtzahl OpsItems AWS-Konto pro Region (einschließlich `Open` und `Resolved` OpsItems): 500.000
- Maximale Anzahl OpsItems AWS-Konto pro Monat: 10.000

Diese Kontingente gelten für OpsItems, die aus einer beliebigen Quelle erstellt wurden, mit folgenden Ausnahmen:

- OpsItemsaufgrund von AWS Security Hub CSPM Erkenntnissen erstellt
- OpsItems, die automatisch generiert werden, wenn ein Vorfall im Incident Manager geöffnet wird

OpsItems, die aus diesen Quellen erstellt werden, werden nicht auf Ihre OpsItem-Kontingente angerechnet, aber jedes OpsItem wird Ihnen in Rechnung gestellt.

Wenn Sie eine `OpsItemLimitExceededException` erhalten, können Sie OpsItems manuell löschen, bis Sie das Kontingent unterschritten haben. Ansonsten können Sie kein neues OpsItem erstellen. Auch hier wird durch das Löschen von CSPM-Ergebnissen oder Incident Manager-Vorfällen, die für Security Hub OpsItems erstellt wurden, die Gesamtzahl der durch die Kontingente OpsItems erzwungenen Fälle nicht reduziert. Sie müssen OpsItems aus anderen Quellen löschen. Informationen zum Löschen eines OpsItem finden Sie unter Löschen Sie OpsItems.

Sie erhalten eine große Rechnung von AWS für eine große Anzahl von automatisch generierten OpsItems

Wenn Sie die Integration mit konfiguriert haben AWS Security Hub CSPM, OpsCenter erstellt OpsItems für Security Hub CSPM-Ergebnisse. Abhängig von der Anzahl der Funde, die Security Hub CSPM generiert, und dem Konto, mit dem Sie bei der Konfiguration der Integration angemeldet waren, OpsCenter kann dies zu einem gewissen Preis eine große Anzahl OpsItems davon generieren. Im Folgenden finden Sie genauere Informationen zu den von Security Hub OpsItems generierten CSPM-Ergebnissen:

- Wenn Sie bei der Konfiguration OpsCenter und der Security Hub CSPM-Integration mit dem Security Hub CSPM-Administratorkonto angemeldet sind, erstellt das System OpsItems für Ergebnisse im Administrator - und allen Mitgliedskonten. Die OpsItems werden alle im Administratorkonto erstellt. Abhängig von einer Vielzahl von Faktoren kann dies zu einer unerwartet hohen Rechnung von AWS führen.

  Wenn Sie bei der Konfiguration der Integration in einem Mitgliedskonto angemeldet sind, erstellt das System nur OpsItems für Erkenntnisse in diesem individuellen Konto. Weitere Informationen zum Security Hub CSPM-Administratorkonto, zu Mitgliedskonten und deren Beziehung zum EventBridge Event-Feed für Ergebnisse finden Sie unter Typen der Security Hub-CSPM-Integration mit EventBridge im Benutzerhandbuch.AWS Security Hub
- Für jede Erkenntnis, das ein OpsItem erstellt, wird Ihnen der reguläre Preis für die Erstellung des OpsItem berechnet. Ihnen wird auch berechnet, wenn Sie das bearbeiten OpsItem oder wenn das entsprechende Ergebnis in Security Hub CSPM aktualisiert wird (was ein OpsItem Update auslöst).
- OpsItemsdie durch eine Integration mit erstellt wurden, AWS Security Hub CSPM sind derzeit nicht durch das maximale Kontingent von 500.000 OpsItems pro Konto in einer Region begrenzt. Es ist daher möglich, dass Security Hub CSPM-Benachrichtigungen in einem Konto mehr als 500.000 kostenpflichtige Gebühren OpsItems in jeder Region erstellen.

  Für Umgebungen mit hoher Produktion empfehlen wir daher, den Umfang der CSPM-Ergebnisse von Security Hub auf Probleme mit hohem Schweregrad zu beschränken.

> **Important**
>
> Wenn Sie der Meinung sind, dass eine große Anzahl irrtümlich erstellt OpsItems wurde und Ihre AWS Rechnung nicht gerechtfertigt ist, wenden Sie sich an. Support

Gehen Sie wie folgt vor, wenn Sie nicht mehr möchten, dass das System CSPM-Ergebnisse OpsItems für Security Hub erstellt.

Um den Empfang von OpsItems Security Hub CSPM-Ergebnissen zu beenden

1. Öffnen Sie die AWS Systems Manager Konsole unter. https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich OpsCenter aus.
3. Wählen Sie Einstellungen aus.
4. Wählen Sie im Abschnitt Security Hub CSPM-Ergebnisse die Option Bearbeiten aus.
5. Wählen Sie den Schieberegler, um Aktiviert zu Deaktiviert zu ändern. Wenn Sie den Schieberegler nicht umschalten können, wurde Security Hub CSPM für Sie nicht aktiviert. AWS-Konto
6. Wählen Sie Speichern, um Ihre Konfiguration zu speichern. OpsCentererstellt nicht mehr auf der OpsItems Grundlage der CSPM-Ergebnisse von Security Hub.

> **Important**
>
> Wenn die Einstellung wieder auf OpsCenter Aktiviert gesetzt wird und weiterhin OpsItems für Ergebnisse erstellt wird, melden Sie sich beim delegierten Systems Manager Manager-Administratorkonto oder beim AWS Organizations Verwaltungskonto an und wiederholen Sie diesen Vorgang. Sollten Sie keine Berechtigung haben, sich bei einem dieser Konten anzumelden, wenden Sie sich an Ihren Administrator und bitten ihn, diesen Vorgang zu wiederholen, um die Integration für Ihr Konto zu deaktivieren.

Amazon-CloudWatch-Dashboards, die von Systems Manager gehostet werden

Amazon CloudWatch-Dashboards sind anpassbare Startseiten in der CloudWatch-Konsole, mit denen Sie Ihre Ressourcen in einer einzigen Ansicht überwachen können, auch solche, die über verschiedene AWS-Regionen verteilt sind. Sie können CloudWatch-Dashboards verwenden, um benutzerdefinierte Ansichten der Metriken und Alarme für Ihre AWS Ressourcen zu erstellen. Mit Dashboards können Sie Folgendes erstellen:

- Eine einzige Ansicht für ausgewählte Metriken und Alarme, um Ihnen die Bewertung des Zustands Ihrer Ressourcen und Anwendungen in einer oder mehreren AWS-Regionen zu erleichtern. Sie können die für jede Metrik in jedem Diagramm verwendete Farbe auswählen, sodass Sie dieselbe Metrik über mehrere Diagramme hinweg verfolgen können.
- Ein operatives Playbook, das Teammitgliedern bei operativen Ereignissen Orientierungshilfen dazu bietet, wie auf bestimmte Vorfälle zu reagieren ist.
- Eine gemeinsame Ansicht wichtiger Maßnahmen für Ressourcen und Anwendungen, die von Teammitgliedern für einen schnelleren Kommunikationsfluss bei operativen Ereignissen gemeinsam genutzt wird.

Sie können Dashboards erstellen, indem Sie die Konsole, AWS Command Line Interface (AWS CLI), oder die CloudWatch `PutDashboard`-API verwenden. Weitere Informationen finden Sie unter Verwenden von Amazon CloudWatch-Dashboards im Benutzerhandbuch für Amazon CloudWatch.

Arbeiten mit SSM Agent

AWS Systems Manager Agent (SSM Agent) ist eine Amazon-Software, die auf Amazon Elastic Compute Cloud (Amazon EC2)-Instances, Edge-Geräten und On-Premises-Server und virtuellen Maschinen (VM) ausgeführt wird. SSM Agent ermöglicht es dem Systems Manager, diese Ressourcen zu aktualisieren, zu verwalten und zu konfigurieren. Der Agent verarbeitet Anforderungen vom Systems-Manager-Service in der AWS Cloud und führt diese dann wie in der Anforderung angegeben aus. Anschließend sendet SSM Agent Informationen zum Status und zur Ausführung zurück zum Systems-Manager-Service mithilfe des Amazon Message Gateway Service (`ssmmessages`). (Bei einem AWS-Regionen-Start vor 2024 können Status- und Ausführungsinformationen auch über Amazon Message Delivery Service(Servicepräfix: `ec2messages`) zurückgesendet werden.)

Wenn Sie den Datenverkehr überwachen, werden Sie sehen, dass Ihre verwalteten Knoten mit `ssmmessages.*`-Endpunkten und `ec2messages.*`-Endpunkten kommunizieren. Weitere Informationen finden Sie unter Referenz: ec2messages, ssmmessages und andere API-Operationen. Weitere Informationen über Portierung von SSM Agent-Protokollen zu Amazon CloudWatch Logs finden Sie unter Einloggen und Überwachen AWS Systems Manager.

Inhalt

- Technische Details über die SSM Agent erfahren
- Finden Sie AMIs mit dem vorinstallierten SSM Agent
- Arbeiten mit SSM Agent auf EC2-Instances für Linux
- Arbeiten mit SSM Agent EC2 On-Instances für macOS
- Arbeiten mit SSM Agent EC2 On-Instances für Windows Server
- Prüfen des SSM Agent-Status und Starten des Agenten
- Überprüfen der SSM Agent-Versionsnummer
- Anzeigen von SSM Agent-Protokollen
- Einschränken des Zugriffs auf Befehle auf Stammebene durch SSM Agent
- Automatisieren von Updates für SSM Agent
- Abonnieren von SSM Agent-Benachrichtigungen
- Fehlerbehebung für SSM Agent

Technische Details über die SSM Agent erfahren

Verwenden Sie die Informationen in diesem Thema, um Ihnen bei der Implementierung von AWS Systems Manager Agent (SSM Agent) zu helfen und zu verstehen, wie der Agent funktioniert.

Themen

- Verhalten der Anmeldeinformationen von SSM Agent-Version 3.2.x.x
- Priorität der SSM Agent-Anmeldeinformationen
- Konfigurieren von SSM Agent für die Verwendung mit dem Federal Information Processing Standard (FIPS)
- Über das lokale ssm-user-Konto
- SSM Agent und die Instance Metadata Service (IMDS)
- Behalten SSM Agent up-to-date
- Sicherstellen, dass das SSM Agent-Installationsverzeichnis nicht geändert, verschoben oder gelöscht wird
- SSM Agentfortlaufende Updates von AWS-Regionen
- SSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets
- SSM Agent auf GitHub
- Grundlegendes zum SSM Agent-Ruhezustand

Verhalten der Anmeldeinformationen von SSM Agent-Version 3.2.x.x

SSM Agent speichert einen Satz temporärer Anmeldeinformationen unter `/var/lib/amazon/ssm/credentials` (für Linux und macOS) oder `%PROGRAMFILES%\Amazon\SSM\credentials` (für Windows Server), wenn eine Instance mit der Standardkonfiguration für die Host-Verwaltung in Quick Setup eingebunden wird. Die temporären Anmeldeinformationen haben dieselben Berechtigungen wie die IAM-Rolle, die Sie für die Standardkonfiguration für die Host-Verwaltung ausgewählt haben. Auf Linux kann nur das Root-Konto auf diese Anmeldeinformationen zugreifen. Auf Windows Server, können nur das SYSTEM-Konto und lokale Administratoren auf diese Anmeldeinformationen zugreifen.

Priorität der SSM Agent-Anmeldeinformationen

In diesem Thema werden wichtige Informationen darüber beschrieben, wie SSM Agentdie Berechtigung zum Ausführen von Aktionen auf Ihren Ressourcen erhält.

> **Note**
>
> Die Support für Edge-Geräte unterscheidet sich geringfügig. Sie müssen Ihre Edge-Geräte für die Verwendung der AWS IoT Greengrass Core-Software konfigurieren, eine AWS Identity and Access Management (IAM) -Servicerolle konfigurieren und die Bereitstellung SSM Agent auf Ihren Geräten mithilfe AWS IoT Greengrass von. Weitere Informationen finden Sie unter Verwalten von Edge-Geräten mit Systems Manager.

Wenn SSM Agent ın einer Maschine installiert ist, sind Berechtigungen für die Kommunikation mit dem Systems-Manager-Service erforderlich. Auf Amazon Elastic Compute Cloud (Amazon EC2) -Instances werden diese Berechtigungen in einem Instance-Profil bereitgestellt, das an die Instance angehängt ist. Ruft auf einem EC2 Computer SSM Agent normalerweise die erforderlichen Berechtigungen aus der Datei mit gemeinsam genutzten Anmeldeinformationen ab, die sich unter `/root/.aws/credentials` (Linux undmacOS) oder `%USERPROFILE%\.aws\credentials` (Windows Server) befindet. Die erforderlichen Berechtigungen werden dieser Datei während des Hybrid-Aktivierungsvorgangs hinzugefügt. Wenn ein hybrider aktivierter Knoten abgemeldet wird, kann der Agent in den Ruhezustand wechseln. Weitere Informationen finden Sie unter Grundlegendes zum SSM Agent-Ruhezustand.

In seltenen Fällen kann es jedoch sein, dass eine Maschine Berechtigungen zu mehr als einem der Speicherorte enthält, in denen SSM Agent prüft, ob Berechtigungen zum Ausführen der Aufgaben vorhanden sind.

Angenommen, Sie haben eine EC2 Instanz so konfiguriert, dass sie von Systems Manager verwaltet wird. Diese Konfiguration umfasst das Anhängen eines Instance-Profils. Aber dann entscheiden Sie sich, diese Instance auch für Entwickler- oder Endbenutzer-Aufgaben zu verwenden und installieren die AWS Command Line Interface (AWS CLI) darauf. Diese Installation führt dazu, dass zusätzliche Berechtigungen zu einer Anmeldeinformationsdatei auf der Instance hinzugefügt werden.

Wenn Sie einen Systems Manager-Befehl für die Instance ausführen,versucht SSM Agent möglicherweise, Anmeldeinformationen zu verwenden, die sich von denen unterscheiden, die von ihm erwartet werden, z. B. aus einer Anmeldeinformationsdatei anstelle eines Instance-Profils. Das liegt daran, dass SSM Agent in der Reihenfolge nach Anmeldeinformationen sucht, die für die Standard-Anbieterkette für Anmeldeinformationen vorgeschrieben ist.

> **Note**
>
> Unter Linux und macOS wird SSM Agent als Root-Benutzer ausgeführt. Daher sind die Umgebungsvariablen und die Datei mit Anmeldeinformationen, nach denen SSM Agent bei diesem Prozess sucht, nur die des Stammbenutzers (`/root/.aws/credentials`). SSM Agent berücksichtigt bei der Suche nach Anmeldeinformationen weder die Umgebungsvariablen noch die Datei mit den Anmeldeinformationen anderer Benutzer der Instance.

Die Standard-Anbieterkette sucht in folgender Reihenfolge nach Anmeldeinformationen:

1. Umgebungsvariablen, wenn konfiguriert (`AWS_ACCESS_KEY_ID` und `AWS_SECRET_ACCESS_KEY`)
2. Freigegebene Anmeldeinformationen-Datei (`$HOME/.aws/credentials` für Linux und macOS oder `%USERPROFILE%\.aws\credentials` für Windows Server) mit Berechtigungen, die beispielsweise durch eine Hybrid-Aktivierung oder eine AWS CLI -Installation bereitgestellt wird.
3. Eine AWS Identity and Access Management (IAM) -Rolle für Aufgaben, wenn eine Anwendung vorhanden ist, die eine Amazon Elastic Container Service (Amazon ECS) -Aufgabendefinition oder RunTask API-Operation verwendet.
4. Ein Instance-Profil, das an eine EC2 Amazon-Instance angehängt ist.
5. Die für die Standardkonfiguration für die Host-Verwaltung gewählte IAM-Rolle.

Verwandte Informationen finden Sie in den folgenden Themen:

- Instanzprofile für EC2 Instanzen — Konfigurieren Sie die für Systems Manager erforderlichen Instanzberechtigungen
- Hybride Aktivierungen – Eine Hybridaktivierung erstellen, um Knoten bei Systems Manager zu registrieren
- AWS CLI Anmeldeinformationen — Konfiguration und Einstellungen für die Anmeldeinformationsdatei im AWS Command Line Interface Benutzerhandbuch
- Standard-Anbieterkette für Anmeldeinformationen – Festlegen von Anmeldeinformationen im AWS SDK für Go -Entwicklerhandbuch

> **Note**
>
> Dieses Thema im AWS SDK für Go -Entwicklerhandbuch beschreibt die Standard-Anbieterkette in Bezug auf das SDK for Go. Die gleichen Prinzipien gelten jedoch für die Auswertung von Anmeldeinformationen für SSM Agent.

Konfigurieren von SSM Agent für die Verwendung mit dem Federal Information Processing Standard (FIPS)

Wenn Sie Systems Manager mit nach dem Federal Information Processing Standard (FIPS) 140-3 validierten kryptografischen Modulen verwenden müssen, können Sie AWS Systems Manager Agent (SSM Agent) so konfigurieren, dass die FIPS-Endpunkte in unterstützten Regionen verwendet werden.

So konfigurieren Sie SSM Agent für die Verbindung zu FIPS 140-3-Endpunkten

1. Herstellen einer Verbindung zu Ihrem verwalteten Knoten.
2. Navigieren Sie zum Verzeichnis, das die `amazon-ssm-agent.json`-Datei enthält:
   - Linux: `/etc/amazon/ssm/`
   - macOS: `/opt/aws/ssm/`
   - Windows Server: `C:\Program Files\Amazon\SSM\`
3. Öffnen Sie die Datei mit dem Namen `amazon-ssm-agent.json`, um sie zu bearbeiten.

   > **Tip**
   >
   > Wenn noch keine `amazon-ssm-agent.json`-Datei vorhanden ist, kopieren Sie den Inhalt von `amazon-ssm-agent.json.template` in eine neue Datei mit dem Namen `amazon-ssm-agent.json`. Speichern Sie `amazon-ssm-agent.json` in demselben Verzeichnis, in dem sich `amazon-ssm-agent.json.template` befindet.

4. Fügen Sie der Datei den folgenden Inhalt hinzu. Ersetzen Sie die *`region`* Platzhalterwerte durch den entsprechenden Regionalcode für Ihre Partition:

```
{
    ---Existing file content, if any---
```

```
    "Mds": {
        "Endpoint": "ec2messages-fips.region.amazonaws.com",
    },
    "Ssm": {
        "Endpoint": "ssm-fips.region.amazonaws.com",
    },
    "Mgs": {
        "Endpoint": "ssmmessages-fips.region.amazonaws.com",
        "Region": "region"
    },
    "S3": {
        "Endpoint": "s3-fips.dualstack.region.amazonaws.com",
        "Region": region"
    },
    "Kms": {
        "Endpoint": "kms-fips.region.amazonaws.com"
    }
}
```

Zu den unterstützten Regionen gehören die folgenden:

- `us-east-1` für die Region USA Ost (Nord-Virginia)
- `us-east-2` für die Region USA Ost (Ohio)
- `us-west-1` für die Region USA West (Nordkalifornien)
- `us-west-2` für die Region USA West (Oregon)
- `ca-west-1` für die Region Kanada West (Calgary)

5. Speichern Sie die Datei und starten Sie SSM Agent.

Jedes Mal, wenn Sie die Konfiguration ändern, starten Sie SSM Agent neu.

Sie können andere Features von SSM Agent mit dem gleichen Verfahren personalisieren. Eine up-to-date Liste der verfügbaren Konfigurationseigenschaften und ihrer Standardwerte finden Sie unter Definitionen von Konfigurationseigenschaften im `amazon-ssm-agent` Repository unter GitHub.

Weitere Informationen zur AWS Unterstützung von FIPS finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

Über das lokale ssm-user-Konto

Ab Version 2.3.50.0 von SSM Agent, erstellt der Agent ein lokales Benutzerkonto namens `ssm-user` und fügt es dem `/etc/sudoers.d`-Verzeichnis (Linux und macOS) bzw. der Administratorengruppe (Windows Server) hinzu. Auf Agenten-Versionen vor 2.3.612.0 wird das Konto beim ersten Start bzw. Neustart des SSM Agent nach der Installation erstellt. Auf Version 2.3.612.0 und höher wird das `ssm-user`-Konto beim ersten Start einer Sitzung auf einer Instance erstellt. Dieser `ssm-user` ist der standardmäßige OS-Benutzer, wenn eine Sitzung in Session Manager, einem Tool in AWS Systems Manager, startet. Sie können die Berechtigungen ändern, indem Sie `ssm-user` einer Gruppe mit weniger Berechtigungen hinzufügen oder die `sudoers`-Datei entsprechend ändern. Das `ssm-user`-Konto wird bei einer Deinstallation von SSM Agent nicht aus dem System entfernt.

Unter Windows Server legt der SSM Agent beim Start einer jeden Sitzung ein neues Passwort für das `ssm-user`-Konto fest. Auf Linux-verwalteten Instances werden keine Passwörter für `ssm-user` festgelegt.

Ab SSM Agent Version 2.3.612.0 wird das `ssm-user`-Konto nicht automatisch auf Windows Server-Maschinen erstellt, die als Domain-Controller verwendet werden. Um Session Manager auf einem Windows Server-Domain-Controller zu verwenden, erstellen Sie das `ssm-user`-Konto manuell, sofern es noch nicht vorhanden ist, und weisen dem Benutzer Domain-Administratorberechtigungen zu.

> **Important**
>
> Damit das `ssm-user`-Konto erstellt werden kann, muss das Instance-Profil, das der Instance zugewiesen ist, über die erforderlichen Berechtigungen verfügen. Weitere Informationen finden Sie unter Schritt 2: Überprüfen oder Hinzufügen von Instance-Berechtigungen für Session Manager.

SSM Agent und die Instance Metadata Service (IMDS)

Systems Manager ist auf EC2 Instanzmetadaten angewiesen, um korrekt zu funktionieren. Systems Manager kann entweder über Version 1 oder Version 2 desInstance Metadata Service (IMDSv1 und IMDSv2) auf Instance-Metadaten zugreifen. Ihre Instanz muss auf die IPv4 Adresse des Instanz-Metadatendienstes zugreifen können: 169.254.169.254. Weitere Informationen finden Sie unter Instance-Metadaten und Benutzerdaten im EC2 Amazon-Benutzerhandbuch.

Behalten SSM Agent up-to-date

Wenn Systems Manager neue Tools hinzugefügt oder Aktualisierungen an den vorhandenen Tools vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Tools und Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent in Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die SSM Agent-Versionshinweise-Seite in GitHub, um Benachrichtigungen über SSM Agent-Updates zu erhalten.

> **Note**
>
> Wenn Systems Manager neue Tools hinzugefügt oder Aktualisierungen an den vorhandenen Tools vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Tools und Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent in Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die SSM Agent-Versionshinweise-Seite in GitHub, um Benachrichtigungen über SSM Agent-Updates zu erhalten.
> Amazon Machine Images (AMIs), die standardmäßig SSM Agent enthalten, können bis zu zwei Wochen benötigen, bis ein aktualisiertes AMI mit der neuesten Version von SSM Agent veröffentlicht wird. Wir empfehlen, dass Sie sogar noch häufigere automatisierte Aktualisierungen für den SSM Agent konfigurieren

Sicherstellen, dass das SSM Agent-Installationsverzeichnis nicht geändert, verschoben oder gelöscht wird

SSM Agent ist unter `/var/lib/amazon/ssm/` (Linux und macOS) und `%PROGRAMFILES%\Amazon\SSM\` (Windows Server) installiert. Diese Installationsverzeichnisse enthalten wichtige Dateien und Ordner, die von SSM Agent verwendet werden, wie z. B. eine Anmeldeinformationsdatei, Ressourcen für die prozessübergreifende Kommunikation (IPC) und Orchestrierungsordner. Nichts im Installationsverzeichnis sollte geändert, verschoben oder gelöscht werden. Andernfalls funktioniert SSM Agent möglicherweise nicht mehr richtig.

SSM Agentfortlaufende Updates von AWS-Regionen

Nachdem ein SSM Agent Update in seinem GitHub Repository verfügbar gemacht wurde, kann es bis zu zwei Wochen dauern, bis die aktualisierte Version zu unterschiedlichen AWS-Regionen Zeiten für alle verfügbar ist. Aus diesem Grund erhalten Sie möglicherweise die Fehlermeldung „Auf der aktuellen Plattform nicht unterstützt“ oder „Aktualisierung amazon-ssm-agent auf eine ältere Version, bitte aktivieren Sie die Option Downgrade zulassen, um fortzufahren“, wenn Sie versuchen, eine neue Version von SSM Agent in einer Region bereitzustellen.

Um die für Sie verfügbare Version von SSM Agent zu ermitteln, können Sie einen `curl`-Befehl ausführen.

Um die im globalen Download-Bucket verfügbare Version des Agenten anzuzeigen, führen Sie den folgenden Befehl aus.

```
curl https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/VERSION
```

Um die Version des Agenten anzuzeigen, die in einer bestimmten Region verfügbar ist, führen Sie den folgenden Befehl aus und ersetzen Sie ihn durch *`region`* die Region, in der Sie arbeiten, z. B. `us-east-2` für die Region USA Ost (Ohio).

```
curl https://s3.region.amazonaws.com/amazon-ssm-region/latest/VERSION
```

Sie können die `VERSION`-Datei auch direkt in Ihrem Browser ohne einen `curl`-Befehl öffnen.

SSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets

Bei der Ausführung verschiedener Systems Manager Manager-Operationen greift AWS Systems Manager Agent (SSM Agent) auf eine Reihe von Amazon Simple Storage Service (Amazon S3) -Buckets zu. Diese S3-Buckets sind öffentlich zugänglich und SSM Agent verbindet sich standardmäßig über `HTTP`-Aufrufe mit ihnen.

Wenn Sie jedoch einen Virtual Private Cloud (VPC) -Endpunkt in Ihren Systems Manager-Vorgängen verwenden, müssen Sie in einem Amazon Elastic Compute Cloud (Amazon EC2) -Instanzprofil für Systems Manager oder in einer Servicerolle für EC2 Nicht-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung eine ausdrückliche Genehmigung erteilen. Andernfalls haben Ihre Ressourcen keinen Zugriff auf diese öffentlichen Buckets.

Um Ihren verwalteten Knoten Zugriff auf diese Buckets zu gewähren, wenn Sie einen VPC-Endpunkt verwenden, erstellen Sie eine benutzerdefinierte Amazon S3 S3-Berechtigungsrichtlinie und fügen

sie dann Ihrem Instance-Profil (für EC2 Instances) oder Ihrer Servicerolle (für nicht EC2 verwaltete Knoten) hinzu.

Informationen zur Verwendung eines VPC-Endpunkts (Virtual Private Cloud) in Ihren Systems Manager-Vorgängen finden Sie unter Verbessern der Sicherheit von EC2 Instanzen mithilfe von VPC-Endpunkten für Systems Manager.

> **Note**
>
> Diese Berechtigungen ermöglichen nur den Zugriff auf die AWS verwalteten Buckets, die für erforderlich sind. SSM Agent Sie erteilen nicht die erforderlichen Berechtigungen für andere Amazon S3-Operationen. Außerdem gewähren sie auch keine Berechtigung für Ihren eigenen S3-Buckets.

Weitere Informationen finden Sie unter den folgenden Themen:

- Erforderliche Instance-Berechtigungen für Systems Manager konfigurieren
- Die für Systems Manager in Hybrid- und Multi-Cloud-Umgebungen erforderliche IAM-Servicerolle erstellen
- Referenz: Amazon-S3-Buckets für Patch-Operationen

Inhalt

- Erforderliche Bucket-Berechtigungen
- Beispiel
- Validierung von hybrid-aktivierten Maschinen mit einem Hardware-Fingerabdruck

Erforderliche Bucket-Berechtigungen

In der folgenden Tabelle werden die einzelnen S3-Buckets beschrieben, auf die SSM Agent möglicherweise für Systems Manager-Operationen zugreifen muss.

> **Note**
>
> *region*steht für den Bezeichner einer Region AWS Systems Manager, die von AWS-Region unterstützt wird, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der

unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

Von SSM Agent erforderte Amazon S3-Berechtigungen

S3 Bucket-ARN	Description
`arn:aws:s3:::aws-windows-downloads-` *region*`/*`	Erforderlich für einige SSM-Dokumente, die nur Windows Server-Betriebssysteme unterstützen, sowie einige für plattformübergreifende Unterstützung, wie `AWSEC2-ConfigureSTIG` .
`arn:aws:s3:::amazon-ssm-` *region*`/*`	Erforderlich zum Aktualisieren von SSM Agent-Installationen. Diese Buckets enthalten die SSM Agent-Installationspakete und die Installationsmanifeste, auf die in dem `AWS-UpdateSSMAgent` -Dokument und Plugin verwiesen wird. Wenn diese Berechtigungen nicht bereitgestellt werden, führt der SSM Agent einen HTTP-Aufruf zum Herunterladen des Updates durch.
`arn:aws:s3:::aws-ssm-` *region*`/*`	Bietet Zugriff auf den S3-Bucket, der die für die Verwendung mit Systems-Manager-Dokumenten (SSM-Befehlsdokumenten) erforderlich ist, einschließlich Vorgängen mit und ohne Patching. Beispiel: `arn:aws:s3:::aws-ssm-us-east-2/*` . Im Folgenden finden Sie einige häufig verwendete SSM-Dokumente, die Module aus diesen Buckets verwenden. • `AWS-RunPatchBaseline` • `AWS-RunPatchBaselineAssociation`

S3 Bucket-ARN	Description
	• `AWS-RunPatchBaselineWithHooks` • `AWS-ApplyPatchBaseline` (ein altes SSM-Dokument) • `AWS-ConfigureWindowsUpdate` • `AWS-FindWindowsUpdates` • `AWS-UpdateSSMAgent`

S3 Bucket-ARN	Description

`arn:aws:s3:::patch-baseline-snapshot-` *region*`/*`
 	
–oder–
 	
`arn:aws:s3:::patch-baseline-snapshot-` *region*`-`*unique-suffix*` /*`
	Bietet Zugriff auf den S3-Bucket, der Patch-Baseline-Snapshots enthält. Dies ist erforderlich, wenn Sie eines der folgenden SSM-Befehlsdokumente verwenden:
	
	• `AWS-RunPatchBaseline`
	• `AWS-RunPatchBaselineAssociation`
	• `AWS-RunPatchBaselineWithHooks`
	• `AWS-ApplyPatchBaseline` (ein altes SSM-Dokument)
	
	Die am häufigsten unterstützten Buckets AWS-Regionen verwenden das folgende Format:
	
	`arn:aws:s3:::patch-baseline-snapshot-` *region*
	
	Für einige Regionen ist ein zusätzliches eindeutiges Suffix im Bucket-Namen enthalten. Der Bucket-Name für die Region Naher Osten (Bahrain) (me-south-1) lautet beispielsweise wie folgt:
	
	• `patch-baseline-snapshot-me-south-1-uduvl7q8`
	
	Eine vollständige Liste der Bucket-Namen für Patch-Baseline-Snapshots finden Sie unter Buckets mit verwalteten Patch-Baseline-Snapshots AWS.

S3 Bucket-ARN	Description
	Note Wenn Sie eine On-Premises-Firewall verwenden und Patch Manager nutzen möchten, muss diese Firewall auch den Zugriff auf den geeigneten Patch-Baseline-Endpunkt zulassen.

S3 Bucket-ARN	Description
Für von Linux und Windows Server verwaltete Knoten: `arn:aws:s3:::aws-patch-manager- region-unique-suffix /*` Für EC2 Amazon-Instances fürmacOS: `arn:aws:s3:::aws-patchmanager-macos- region-unique-suffix /*`	Bietet Zugriff auf den S3-Bucket, der Module enthält, die von SSM-Befehlsdokumenten für Patchingvorgänge in Patch Manager verwendet werden. Jeder Bucket-Name enthält ein eindeutiges Suffix, z. B. `552881074` für Buckets in der Region USA Ost (Ohio) (us-east-2): • `arn:aws:s3:::aws-patch-manager-us-east-2-552881074/*` • `arn:aws:s3::: aws-patchmanager-macos-us-east-2-552881074/*` SSM-Dokumente Im Folgenden finden Sie einige häufig verwendete SSM-Dokumente, die Module aus diesen Buckets verwenden. • `AWS-RunPatchBaseline` • `AWS-RunPatchBaselineAssociation` • `AWS-RunPatchBaselineWithHooks` • `AWS-InstanceRebootWithHooks` • `AWS-PatchAsgInstance` • `AWS-PatchInstanceWithRollback` Vollständige Listen der AWS verwalteten S3-Buckets für Patch-Operationen finden Sie in den folgenden Themen: • Buckets, die SSM-Befehlsdokumente für Patch-Operationen enthalten (Linux und Windows Server)

S3 Bucket-ARN	Description
	• Buckets, die SSM-Befehlsdokumente für Patch-Operationen enthalten (macOS)

Beispiel

Das folgende Beispiel veranschaulicht, wie Zugriff auf die S3-Buckets erteilt wird, die in der Region USA Ost (Ohio) (us-east-2) für Systems Manager-Operationen benötigt werden. In den meisten Fällen müssen Sie diese Berechtigungen explizit in einem Instance-Profil oder einer Servicerolle nur dann bereitstellen, wenn Sie einen VPC Endpunkt verwenden.

> **Important**
>
> Wir empfehlen, dass Sie keine Platzhalterzeichen (*) für bestimmte Regionen in dieser Richtlinie verwenden. Verwenden Sie beispielsweise `arn:aws:s3:::aws-ssm-us-east-2/*` und nicht `arn:aws:s3:::aws-ssm-*/*`. Bei der Verwendung von Platzhaltern könnte Zugriff auf S3-Buckets erteilt werden, für die Sie keinen Zugriff gewähren möchten. Wenn Sie das Instance-Profil für mehr als eine Region verwenden möchten, empfehlen wir, den ersten `Statement`-Block für jede Region zu wiederholen.

JSON

```
{
        "Version":"2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": "s3:GetObject",
                "Resource": [
                    "arn:aws:s3:::aws-windows-downloads-us-east-2/*",
                    "arn:aws:s3:::amazon-ssm-us-east-2/*",
                    "arn:aws:s3:::aws-ssm-us-east-2/*",
                    "arn:aws:s3:::patch-baseline-snapshot-us-east-2/*",
                    "arn:aws:s3:::aws-patch-manager-us-east-2-552881074/*",
                    "arn:aws:s3:::aws-patchmanager-macos-us-east-2-552881074/*"
                ]
            }
```

```
        ]
    }
```

Validierung von hybrid-aktivierten Maschinen mit einem Hardware-Fingerabdruck

SSM AgentErfasst bei EC2 Nichtcomputern in einer Hybrid- und Multi-Cloud-Umgebung eine Reihe von Systemattributen (wird als Hardware-Hash bezeichnet) und verwendet diese Attribute, um einen Fingerabdruck zu berechnen. Der Fingerabdruck ist eine undurchsichtige Zeichenfolge, die der Agent an einen bestimmten Systems Manager APIs weitergibt. Dieser eindeutige Fingerabdruck ordnet den Abrufer einem bestimmten hybrid-aktivierten verwalteten Knoten zu. Der Agent speichert den Fingerabdruck und den Hardware-Hash auf der lokalen Festplatte an einem Speicherort namens Vault.

Der Agent berechnet den Hardware-Hash und den Fingerabdruck, wenn die Maschine für die Verwendung mit Systems Manager registriert wird. Anschließend wird der Fingerabdruck an den Systems Manager-Service zurückgegeben, wenn der Agent einen `RegisterManagedInstance`-Befehl sendet.

Später, wenn ein `RequestManagedInstanceRoleToken`-Befehl gesendet wird, überprüft der Agent den Fingerabdruck und den Hardware-Hash im Vault, um sicherzustellen, dass die aktuellen Computerattribute mit dem gespeicherten Hardware-Hash übereinstimmen. Wenn die aktuellen Computerattribute mit dem im Vault gespeicherten Hardware-Hash übereinstimmen, übergibt der Agent den Fingerabdruck aus dem Vault an `RegisterManagedInstance`, was zu einem erfolgreichen Aufruf führt.

Wenn die aktuellen Computerattribute nicht mit dem gespeicherten Hardware-Hash übereinstimmen, berechnet SSM Agent einen neuen Fingerabdruck, speichert den neuen Hardware-Hash und Fingerabdruck im Vault und übergibt den neuen Fingerabdruck an `RequestManagedInstanceRoleToken`. Dadurch schlägt *`RequestManagedInstanceRoleToken`* fehl und der Agent kann kein Rollen-Token für die Verbindung mit dem Systems Manager-Service abrufen.

Dieser Fehler ist vorgesehen und wird als Verifizierungsschritt verwendet, um zu verhindern, dass mehrere verwaltete Knoten als derselbe verwaltete Knoten mit dem Systems-Manager-Service kommunizieren.

Beim Vergleich der aktuellen Computerattribute mit dem im Vault gespeicherten Hardware-Hash verwendet der Agent die folgende Logik, um festzustellen, ob die alten und neuen Hashes übereinstimmen:

- Wenn die SID (System/Maschinen-ID) anders ist, gibt es keine Übereinstimmung.
- Wenn die IP-Adresse identisch ist, gibt es eine Übereinstimmung.
- Andernfalls wird der Prozentsatz der übereinstimmenden Computerattribute berechnet und mit dem vom Benutzer konfigurierten Ähnlichkeitsschwellenwert verglichen, um festzustellen, ob eine Übereinstimmung vorliegt.

Der Ähnlichkeitsschwellenwert wird im Vault als Teil des Hardware-Hash gespeichert.

Der Ähnlichkeitsschwellenwert kann festgelegt werden, nachdem eine Instance mit einem Befehl wie dem folgenden registriert wurde.

Auf Linux-Maschinen:

```
sudo amazon-ssm-agent -fingerprint -similarityThreshold 1
```

Auf Windows Server Maschinen, die Folgendes verwenden PowerShell:

```
cd "C:\Program Files\Amazon\SSM\" `
    .\amazon-ssm-agent.exe -fingerprint -similarityThreshold 1
```

> **Important**
>
> Wenn sich eine der Komponenten zur Berechnung des Fingerprints ändert, kann dies dazu führen, dass der Agent in den Ruhezustand versetzt wird. Um diesen Ruhezustand zu vermeiden, setzen Sie den Ähnlichkeitsschwellenwert auf einen niedrigen Wert, z.B. **1**. Weitere Informationen zum Ruhezustand finden Sie unter Grundlegendes zum SSM Agent-Ruhezustand.

SSM Agent auf GitHub

Der Quellcode für SSM Agent ist in GitHub verfügbar, damit Sie den Agenten entsprechend Ihren Anforderungen anpassen können. Wir möchten Sie bitten, uns eventuelle Änderungswünsche mitzuteilen. Amazon Web Services bietet jedoch keine Unterstützung für die Ausführung modifizierter Kopien dieser Software.

Grundlegendes zum SSM Agent-Ruhezustand

AWS Systems Manager Der Ruhezustand von Agent (SSM Agent) ist ein Betriebsmodus, der auftritt, wenn der Agent nicht ordnungsgemäß mit dem Systems Manager Manager-Dienst kommunizieren kann. Während des Ruhezustands reduziert der Agent seine Kommunikationshäufigkeit und wechselt in einen Bereitschaftszustand.

Wenn der SSM Agent-Ruhezustand eintritt

Der SSM Agent-Ruhezustand kann in folgenden Szenarien auftreten:

Abgemeldete Hybridknoten

Wenn Sie einen hybridaktivierten Knoten bei Systems Manager abmelden, kann der SSM Agent auf diesem Knoten sein Autorisierungstoken nicht aktualisieren. Dadurch wechselt der Agent in den Ruhezustand, da er sich beim Service nicht authentifizieren kann.

Änderungen am Hardware-Fingerabdruck

SSM Agent verwendet einen Hardware-Fingerabdruck zur Validierung von hybrid-aktivierten Maschinen. Wenn sich eine der Komponenten zur Berechnung des Fingerabdrucks erheblich ändert, kann dies dazu führen, dass der Agent aus Sicherheitsgründen in den Ruhezustand versetzt wird. Diese Aktion ist vorgesehen, um zu verhindern, dass mehrere verwaltete Knoten als derselbe Knoten mit dem Systems-Manager kommunizieren. Weitere Informationen finden Sie unter Validierung von hybrid-aktivierten Maschinen mit einem Hardware-Fingerabdruck.

SSM AgentRuhezustand auf Amazon-Instances EC2

Unter bestimmten Bedingungen kann es auch bei EC2 Amazon-Instances zu einem Ruhezustand kommen, z. B. wenn Verbindungsprobleme oder Authentifizierungsprobleme mit dem Systems Manager Manager-Service auftreten.

Kommunikationsverhalten im Ruhezustand

Wenn SSM Agent in den Ruhezustand eintritt, ändert sich das Kommunikationsmuster mit dem Systems-Manager-Service:

- Normaler Betrieb: Der Agent kommuniziert regelmäßig (normalerweise alle paar Minuten) mit Systems Manager, um nach neuen Befehlen zu suchen und den Status zu melden.
- Ruhezustand: Die Ping-Frequenz beginnt bei 5 Minuten und wird standardmäßig schrittweise auf einmal pro Stunde erhöht (konfigurierbar bis zu 24 Stunden). Diese reduzierte

Kommunikationsfrequenz trägt dazu bei, unnötigen Netzwerkverkehr zu minimieren, und ermöglicht es dem Agenten dennoch, sich möglicherweise zu erholen, wenn sich die Bedingungen ändern.

Während des Ruhezustands nimmt der Agent weiterhin Authentifizierungs- und Verbindungsversuche mit der reduzierten Häufigkeit vor, kann jedoch keine neuen Befehle verarbeiten oder detaillierte Statusinformationen melden, bis der Ruhezustand behoben ist.

Konfigurationsoptionen zur Verhinderung des Ruhezustands in Hybrid-Instances

Die primäre Konfigurationsoption zur Verhinderung des Ruhezustands aufgrund von Änderungen am Hardware-Fingerabdruck ist die Anpassung des Ähnlichkeitsschwellenwerts:

Auf Linux-Maschinen:

```
sudo amazon-ssm-agent -fingerprint -similarityThreshold 1
```

Auf Windows Server-Computern mit: PowerShell

```
cd "C:\Program Files\Amazon\SSM\" `
.\amazon-ssm-agent.exe -fingerprint -similarityThreshold 1
```

Der Ähnlichkeitsschwellenwert bestimmt, wie genau der Agent die aktuellen Maschinenattribute mit dem gespeicherten Hardware-Hash vergleicht:

- Höhere Werte erfordern mehr passende Attribute
- Niedrigere Werte (wie **1**) sind weniger streng und können dazu beitragen, den Ruhezustand zu vermeiden, der durch geringfügige Hardwareänderungen verursacht wird

Protokollierung und Überwachung des Ruhezustands

Wenn SSM Agent in den Ruhezustand eintritt, werden Protokolleinträge erstellt, anhand derer Sie den Ruhezustand identifizieren und beheben können:

- Agent-Protokolldateien: Ruhezustandsereignisse werden in den Standard-SSM Agent-Protokolldateien protokolliert. Weitere Informationen zu den Protokoll-Speicherorten finden Sie unter Probleme mithilfe von SSM Agent-Protokolldateien beheben.
- EC2 Amazon-Konsolenprotokollierung: Für EC2 Instances werden Ruhezustandsmeldungen jetzt in den Systemprotokollen der EC2 Amazon-Konsole protokolliert, was zusätzliche Einblicke in den Agentenstatus bietet. Um auf die Protokolle zuzugreifen, wählen Sie die Instance in

der EC2 Konsole aus und klicken Sie dann auf Aktionen, Überwachung und Fehlerbehebung, Systemprotokoll abrufen.

- Spezifische Protokolldateien: Wenn der Ruhezustand beginnt, werden spezielle Protokolldateien erstellt, die detaillierte Informationen über den Auslöser und den Status des Ruhezustands enthalten.

Überwachen Sie diese Protokollquellen, um Ereignisse im Ruhezustand frühzeitig zu erkennen und Korrekturmaßnahmen zu ergreifen, um den normalen Agentbetrieb wiederherzustellen.

Erholung aus dem Ruhezustand

Um sich aus dem Ruhezustand zu erholen, beheben Sie die zugrunde liegende Ursache:

- Für abgemeldete Hybridknoten: Registrieren Sie den Knoten erneut bei Systems Manager mit einem neuen Aktivierungscode und einer neuen ID, wie unter the section called “Abmelden und Neuregistrieren eines verwalteten Knotens (Linux)” und the section called “Abmelden und Neuregistrieren eines verwalteten Knotens (Windows Server)” beschrieben.
- Bei Problemen mit dem Hardware-Fingerabdruck: Passen Sie den Ähnlichkeitsschwellenwert wie oben unter Konfigurationsoptionen, um den Ruhezustand in Hybrid-Instances zu verhindern beschrieben, oder registrieren Sie den Knoten erneut, wenn wesentliche Änderungen an der Hardware vorgenommen wurden.
- Bei Verbindungsproblemen: Überprüfen Sie die Netzwerkkonnektivität und stellen Sie sicher, dass auf die erforderlichen Endpunkte zugegriffen werden kann. Weitere Informationen finden Sie unter Problembehandlung bei der Verfügbarkeit von verwalteten Knoten mit `ssm-cli`.

Nachdem Sie das zugrundeliegende Problem behoben haben, sollte der Agent den Ruhezustand automatisch beenden und den normalen Betrieb beim nächsten Kommunikationsversuch wieder aufnehmen.

Finden Sie AMIs mit dem vorinstallierten SSM Agent

AWS Systems Manager Agent (SSM Agent) ist auf einigen Amazon Machine Images (AMIs), die von Drittanbietern bereitgestellt werden AWS und denen sie vertrauen, vorinstalliert.

Wenn Sie beispielsweise eine Amazon Elastic Compute Cloud (Amazon EC2) -Instance starten, die AMI mit einem der folgenden Betriebssysteme erstellt wurde, werden Sie wahrscheinlich feststellen, dass die bereits installiert SSM Agent ist:

- AlmaLinux
- Amazon Linux 2
- Amazon Linux 2 ECS-optimierte Basis-AMIs
- Amazon Linux 2023 (AL2023)
- Amazon-EKS-optimierte Amazon Linux-AMIs
- macOS 13.x (Ventura), 14.x (Sonoma) und 15.x (Sequoia)
- SUSE Linux Enterprise Server (SLES) 15.3 und höher
- Ubuntu Server 16.04 LTS 64-Bit (Snap), 18.04, 20.04, 22.04 LTS, 24.04 LTS, 24.0 und 25.04
- Windows Server 2012 R2-AMIs, die im November 2016 oder später veröffentlicht wurden
- Windows Server 2016, 2019, 2022 (ohne Nano-Versionen) und 2025

> **Note**
>
> Die auf einem SSM Agent vorinstallierte Version von AMI ist möglicherweise nicht die neueste verfügbare Version. Als bewährte Methode empfehlen wir stets die Verwendung der neuesten verfügbaren Version von SSM Agent auf Ihren verwalteten Knoten. Weitere allgemeine Informationen zu Updates zur Automatisierung SSM Agent finden Sie unter Automatisieren von Updates für SSM Agent.
> SSM Agentist möglicherweise auf AWS verwalteten Geräten vorinstalliertAMIs, die nicht auf dieser Liste stehen. Dies weist normalerweise darauf hin, dass das Betriebssystem (OS) nicht vollständig von allen Systems-Manager-Tools unterstützt wird.
> SSM Agentist möglicherweise auch in AWS Marketplace oder im AMIs AMIs Community-Repository vorinstalliert, unterstützt diese aber AWS nicht. AMIs

Überprüfen des Status des SSM Agent

Je nachdem, wann diese initialisiert wurde, ist bei einer aus einem AMI in der vorhergehenden Liste erstellten Instance möglicherweise kein SSM Agent vorinstalliert. Es ist auch möglich, dass der Agent auf einer Instance vorinstalliert ist, der Agent jedoch nicht ausgeführt wird. Daher empfehlen wir Ihnen, den Status von SSM Agent zu überprüfen, bevor Sie versuchen, Systems Manager zum ersten Mal auf einer Instance zu verwenden.

Verwenden Sie das folgende Verfahren, um zu überprüfen, ob SSM Agent installiert ist und auf einer Instance ausgeführt wird. Wenn Sie feststellen, dass der Agent nicht installiert ist, können Sie ihn unter Linux, macOS und Windows Server-Instances manuell installieren.

So überprüfen Sie die Installation von SSM Agent auf einer Instance

1. Warten Sie nach dem Start einer neuen Instance einige Minuten, bis diese initialisiert ist.
2. Stellen Sie mit Ihrer bevorzugten Methode eine Verbindung zur Instance her. Sie können beispielsweise SSH verwenden, um eine Verbindung zu Linux-Instances herzustellen, oder Remote Desktop verwenden, um eine Verbindung zu Windows Server-Instances herzustellen.
3. Prüfen Sie den Status von SSM Agent, indem Sie den Befehl für den Betriebssystemtyp Ihrer Instance ausführen.

Betriebssystem	Befehl
Amazon Linux 2 und Amazon Linux 2023	`sudo systemctl status amazon-ssm-agent`
macOS	Es gibt keinen Befehl zum Überprüfen des SSM Agent-Status auf macOS. Sie können den Status überprüfen, indem Sie die Agent-Protokolldatei `/var/log/amazon/ssm/amazon-ssm-agent.log` suchen und auswerten.
SUSE Linux Enterprise Server	`sudo systemctl status amazon-ssm-agent`
Ubuntu Server (64 Bit – Deb)	`sudo systemctl status amazon-ssm-agent`
Ubuntu Server (64 Bit – Snap)	`sudo systemctl status snap.amazon-ssm-agent.amazon-ssm-agent.service`
Windows Server	`Get-Service AmazonSSMAgent`

> **Tip**
>
> Informationen zu den Befehlen zur Überprüfung des SSM Agent-Status für alle von Systems Manager unterstützten Betriebssystemtypen finden Sie unter Prüfen des SSM Agent-Status und Starten des Agenten.

4. Werten Sie die Befehlsausgabe aus, um den Status von SSM Agent zu erfahren.

   Status: Installiert und ausgeführt

   In den meisten Fällen zeigt die Befehlsausgabe an, dass der Agent installiert ist und ausgeführt wird.

   Das folgende Beispiel zeigt, dass SSM Agent auf einer Amazon-Linux-2-Instance installiert ist und ausgeführt wird.

   ```
   amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; enabled; vendor
    preset: enabled)
   Active: active (running) since Wed 2021-10-20 19:09:29 UTC; 4min 6s ago
   --truncated--
   ```

   Das folgende Beispiel zeigt, dass SSM Agent auf einer Windows Server-Instance installiert ist und ausgeführt wird.

   ```
   Status   Name               DisplayName
   ------   ----               -----------
   Running  AmazonSSMAgent     Amazon SSM Agent
   ```

   Status: Installiert, aber nicht ausgeführt

   In einigen Fällen gibt die Befehlsausgabe an, dass der Agent installiert ist, aber nicht ausgeführt wird.

   Das folgende Beispiel zeigt, dass SSM Agent auf einer Amazon-Linux-2-Instance installiert ist, aber nicht ausgeführt wird.

```
amazon-ssm-agent.service - amazon-ssm-agent
Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: enabled)
Active: inactive (dead) since Wed 2021-10-20 22:16:41 UTC; 18s ago
--truncated--
```

Das folgende Beispiel zeigt, dass SSM Agent installiert ist, aber nicht auf einer Windows Server-Instance ausgeführt wird.

```
Status   Name               DisplayName
------   ----               -----------
Stopped  AmazonSSMAgent     Amazon SSM Agent
```

Wenn der Agent installiert ist, aber nicht ausgeführt wird, können Sie ihn manuell aktivieren, indem Sie die Befehle für den Betriebssystemtyp Ihrer Instace verwenden.

Betriebssystem	Befehl

Amazon Linux 2 und Amazon Linux 2023	`sudo systemctl enable amazon-ssm-agent`
	
	`sudo systemctl start amazon-ssm-agent`

macOS	`sudo launchctl load -w /Library/LaunchDaemons/com.amazon.aws.ssm.plist`
	
	`sudo launchctl start com.amazon.aws.ssm`

Betriebssystem	Befehl

SUSE Linux Enterprise Server	`sudo systemctl enable amazon-ssm-agent`
	
	`sudo systemctl start amazon-ssm-agent`

Ubuntu Server (64 Bit – Deb)	`sudo systemctl enable amazon-ssm-agent`
	
	`sudo systemctl start amazon-ssm-agent`

Ubuntu Server (64 Bit – Snap)	`sudo snap start amazon-ssm-agent`
Windows Server	Führen Sie den folgenden Befehl in PowerShell aus. `Start-Service AmazonSSMAgent`

Status: Nicht installiert

In einigen Fällen gibt die Befehlsausgabe an, dass der Agent nicht installiert ist.

Das folgende Beispiel zeigt, dass SSM Agent nicht auf einer Amazon-Linux-2-Instance installiert ist.

```
Unit amazon-ssm-agent.service could not be found.
```

Das folgende Beispiel zeigt, dass SSM Agent nicht auf einer Windows Server-Instance installiert ist.

```
Get-Service : Cannot find any service with service name 'AmazonSSMAgent'.
--truncated--
```

Wenn der Agent nicht installiert ist, können Sie ihn manuell installieren, indem Sie das Verfahren für Ihren Betriebssystemtyp verwenden:

- Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux
- Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für macOS
- Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Windows Server

Arbeiten mit SSM Agent auf EC2-Instances für Linux

Der AWS Systems Manager-Agent (SSM Agent) verarbeitet Systems Manager-Anforderungen und konfiguriert Ihren Computer entsprechend der Angaben in der Anforderung. Verwenden Sie die Verfahren in den folgenden Themen, um SSM Agent auf Linux-Betriebssystemen zu installieren, zu konfigurieren oder zu deinstallieren.

Themen

- Verifizieren der Signatur von SSM Agent
- Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux
- Konfigurieren Sie SSM Agent, um einen Proxy in Linux-Knoten zu verwenden

Verifizieren der Signatur von SSM Agent

Die AWS Systems Manager Agent (SSM Agent) deb- und rpm-Installationspakete für Linux-Instances sind kryptografisch signiert. Sie können einen öffentlichen Schlüssel verwenden, um sicherzustellen, dass das Paket des Agenten original und unverändert ist. Wenn die Dateien beschädigt oder verändert werden, schlägt die Überprüfung fehl. Sie können die Signatur des Installer-Pakets entweder mit RPM oder GPG überprüfen. Die folgenden Informationen sind für SSM Agent-Versionen 3.1.1141.0 oder höher.

Die richtige Signaturdatei für die Architektur und das Betriebssystem Ihrer Instance finden Sie in der folgenden Tabelle.

*`region`*stellt den Bezeichner für eine Region dar AWS Systems Manager, die von AWS-Region unterstützt wird, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *`region`* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

Architektur	Betriebssystem	URL der Signaturdatei	Agent-Download-Dateiname

x86_64	AlmaLinux, Amazon Linux 2, Amazon Linux 2023,RHEL ,Oracle Linux, Rocky Linux	`https://s3.region.amazonaws.com/amazon-ssm-region/latest/linux_amd64/amazon-ssm-agent.rpm.sig`
		
		`https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_amd64/amazon-ssm-agent.rpm.sig`
			`amazon-ssm-agent.rpm`

x86_64	Debian Server, Ubuntu Server	`https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_amd64/amazon-ssm-agent.deb.sig`
		
		`https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/`
			`amazon-ssm-agent.deb`

Architektur	Betriebssystem	URL der Signaturdatei	Agent-Download-Dateiname
		`debian_amd64/amazon-ssm-agent.deb.sig`	

ARM64	Amazon Linux 2, Amazon Linux 2023, RHEL	`https://s3.region.amazonaws.com/amazon-ssm-region/latest/linux_arm64/amazon-ssm-agent.rpm.sig`
		
		`https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_arm64/amazon-ssm-agent.rpm.sig`
			`amazon-ssm-agent.rpm`

Überprüfen des SSM Agent-Pakets auf einem Linux-Server (v3.3.1802.0 und höher)

Bevor Sie beginnen

Die Verfahren für GPG und RPM in diesem Abschnitt gelten für SSM Agent-Version 3.3.1802.0 und höher. Bevor Sie die Signatur von SSM Agent mit dem folgenden Verfahren überprüfen, stellen Sie sicher, dass Sie das neueste Agent-Paket für Ihr Betriebssystem heruntergeladen haben. Beispiel, `https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_arm64/amazon-ssm-agent.rpm`. Weitere Informationen zum Herunterladen von SSM Agent-Paketen finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux.

Wenn Sie einen Grund haben, weiterhin die Agent-Version 3.3.1611.0 oder früher zu verwenden, befolgen Sie stattdessen die Anweisungen in Überprüfen des SSM Agent-Paket auf einem Linux-Server (v3.3.1611.0 und älter).

GPG

Überprüfen Sie das SSM Agent-Paket auf einem Linux-Server (v3.3.1802.0 und höher) wie folgt

1. Kopieren Sie einen der folgenden öffentlichen Schlüssel und speichern Sie ihn in einer Datei mit dem Namen `amazon-ssm-agent.gpg`.

> **Important**
>
> Der folgende öffentliche Schlüssel läuft am 15.07.2026 (15. Juli 2026) ab. Systems Manager wird in diesem Thema einen neuen öffentlichen Schlüssel veröffentlichen, bevor der alte abläuft. Wir empfehlen Ihnen, den RSS-Feed für dieses Thema zu abonnieren, um eine Benachrichtigung zu erhalten, wenn der neue Schlüssel verfügbar ist.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.22 (GNU/Linux)

mQINBGeRNq4BEACrlf5h6Pz+k+M+QCJJ2LfK7d2Tn9J8iJ9qBK2Vwvuxco1rpSO+
KEI3nTeysPuheximps8WOCADX4VlbsKxMZQLjQM4mA26m1Tiw9nAI4kod4bKjiuM
BMUTCD1wfnjH3zQi4kDUdbpfAEMiPgNLVLH85Wf+lhK+Zm+V38DYzLyVj03kX4wK
iG6RMoxzOBZa5gNsVq+j+oCUITGz/URxH713Rgo8WeoEegI0+7iCBLKg+PM0b7GV
2nzkwWJz796HdkqSg8BwXsYaLTrHxa2P1IpwPCisAkyO7gZaMd6Uj69dtMFO+V8a
Qee6b57qGuFKZw7h1Vvc85PbF1Gy/wNIpary57kUHBFUg1vYep/roJuEbJCq97r5
I2liLl4NAyrWb9r/TAVxlXvqM4iZUhxm8GAp0FywMdBr9ZECClKa5HxuVmlm0Wgl
TXoYTOZKeDg6ZoCvyhNxWneCNip74fohXymeFF5L/budhBwy5wuwSniOgTGLo/4C
VgZHWCcN+d0Q3bx/sl2QNqPg5/xzsxEtymXLdVdwLIsLdEQUnIvy8KTs5jol3Dwi
nnEEyhly6wdaw+qDOhkSOT/VnErrSMkYF8VJfa5GjhCBWKw9JVSkaP2CI/VHOgHM
MKROnulq0hRQBR7RmLYt98xu38BHJWMmF8Ga/HJuIxzD1VmkZOPvDDESUwARAQAB
tCdTU00gQWdlbnQgPHNzbS1hZ2VudC1zaWduZXJAYW1hem9uLmNvbT6JAj8EEwEC
ACkFAmeRNq4CGy8FCQLGmIAHCwkIBwMCAQYVCAIJCgsEFgIDAQIeAQIXgAAKCRBR
qOBQ0AUuXTdND/9qldQ1E3dYjBVXOnbhiUQL594bkS5VoEX7D4fZ5UMVZa5pGiz+
husnoRUS9rH1cSeq7aHJu9hSCMuMdvRpuoo0CwLB+7HtzJvAO2M01hcEkUYa6Qdj
njTzP0ZjnoenJmqF9SYmVqAI/VPa9mNQ1OJ+HQ3qh5i6w+FoWlVqEdXjZGrWijub
TqyN33i1Y26t7Os/x8I9fUeNx37y/7Kama8LTdtv9GhWiMVBg2IuVf27HCMYofrQ
m2uCGe61IhtsnhsYaYupmljl+6qgdiuCiS9BAsoIGtqTnu8lnKcGyGz6YnRszN+U
```

```
1bNE4w+UFpXWJF8ogpYcghJ06aW/LhjZnQSx3VliLdW8eOJzou41yWmiuL3ZY8eW
KAlD+7eYKS6N6fEJCeNO2VX2lcKtDfaOX+lqGIVyexKayMfpi+0frNzt/92YCpF5
3jkeS77vMMVqKIUiIp1OCGv3XsFpIr6Bt2c2throYPDoQL3zvq6vvG40BKeRQ4tT
Y+5vTc8MeNn3LdzTl9pusxTcKifrJq7f5FIsL2CpAX8uQ+Qz+XWsYQQ5PvyUDtOz
nU/MRZaP6HnqY42bzI9ZlKgXi9IE3MXIwoET9YyzFjkIDvat7SlB4uJCpeIqp/KM
OIrTMb7paGLYmBU6YqxNBkDWItNG7NeZzyhh/R/Qqb4vJaf4S+ZqD1RZXokCHAQQ
AQIABgUCZ5E2rwAKCRB90Jej2tf1/CdnD/46It+RNoE00TesZK5n2bijH5Eljw0E
4/UpMi1SV6t2zY7lIm7TcKNn18tynJNFqB6YXXOwSbBG/fbN2E9RaoUCZw23TmAv
amuHwrfsDqsHb7zzPF0bISYjqEDLQJj/gtEugUc6XY1dEpFSlWJIOvgryG04cFXI
uD2KY87ya4s1R+sEVAJ14K4RlUCiMmzJdR0NJNYJOwBi1gkLEp6jG86ttiG2U7fY
pE2ibV+c0GeIFq8PIzqqENsn9KBuRH5EcbdBwfnsj2XfM4aR3ZtRIdWXkKkdP9Rs
yU5dTF/Y7XPId5h8/gp00+DMlXFBinQ1jE7A7eDYviEFd1ba8P7dIom3Q3gzKiWu
KTGpnykShs5NvpQmvGUF6JqDHI4RK9s3kLqsNyZkhenJfRBrJ/45fQAuP4CRedkF
7PSfX0Xp7kDnKuyK6wEUEfXXrqmuLGDmigTXblO5qgdyMwkOLjiY9znBZbHoKs76
VplOoNgGnN19i3nuMcPf2npFICJv7kTIyn5Fh7pjWDCahl3U/PwoLjrrlEzpyStU
oXSZrK3kiAADEdSODXJl8KYU0Pb27JbRr1ZbWnxb+O39TOhtssstulkR0v+IDGDQ
rQE1b12sKgcNFSzInzWrNGu4S06WN8DYzlrTZ9aSHj+37ZqpXAevi8WOFXKPV3PA
E6+O8RI2451Dcg==
=aDkv
-----END PGP PUBLIC KEY BLOCK-----
```

2. Importieren Sie den öffentlichen Schlüssel in Ihren Schlüsselbund und notieren Sie den zurückgegebenen Schlüsselwert.

```
gpg --import amazon-ssm-agent.gpg
```

3. Überprüfen Sie den Fingerprint Achten Sie darauf, den Wert *key-value* durch den Wert aus dem vorherigen Schritt zu ersetzen. Es wird empfohlen, GPG zu verwenden, um den Fingerprint zu überprüfen, auch wenn Sie RPM verwenden, um das Installer-Paket zu überprüfen.

```
gpg --fingerprint key-value
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück:

```
pub   4096R/D0052E5D 2025-01-22 [expires: 2026-07-15]
      Key fingerprint = 4855 A9E6 8332 16D6 A77D  8FE4 51A8 E050 D005 2E5D
uid                  SSM Agent <ssm-agent-signer@amazon.com>
```

Der Fingerabdruck sollte wie folgt aussehen.

```
4855 A9E6 8332 16D6 A77D 8FE4 51A8 E050 D005 2E5D
```

Wenn die Fingerabdruck-Zeichenfolge nicht übereinstimmt, installieren Sie den Agent nicht. Kontakt AWS -Support.

4. Laden Sie die Signaturdatei entsprechend der Architektur und dem Betriebssystem Ihrer Instance herunter.
5. Überprüfen Sie die Installer-Paketsignatur. Achten Sie darauf, das *signature-filename* und *agent-download-filename* durch die Werte zu ersetzen, die Sie beim Herunterladen der Signaturdatei und des Agenten angegeben haben, wie in der Tabelle weiter oben in diesem Thema aufgeführt.

```
gpg --verify signature-filename agent-download-filename
```

Zum Beispiel für die x86_64-Architektur auf Amazon Linux 2:

```
gpg --verify amazon-ssm-agent.rpm.sig amazon-ssm-agent.rpm
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück:

```
gpg: Signature made Sat 08 Feb 2025 12:05:08 AM UTC using RSA key ID D0052E5D
gpg: Good signature from "SSM Agent <ssm-agent-signer@amazon.com>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 4855 A9E6 8332 16D6 A77D  8FE4 51A8 E050 D005 2E5D
```

Wenn die Ausgabe die Bezeichnung `BAD signature`enthält, überprüfen Sie, ob Sie das Verfahren korrekt durchgeführt haben. Wenn Sie weiterhin diese Antwort erhalten, wenden Sie sich an den Agenten Support und installieren Sie ihn nicht. Die Vertrauens-Warnmeldung bedeutet nicht, dass die Signatur ungültig ist, sondern nur, dass Sie den öffentlichen Schlüssel nicht überprüft haben. Beachten Sie die Warnung zu vertrauenswürdigen Inhalten. Wenn die Ausgabe die Phrase `Can't check signature: No public key` enthält, überprüfen Sie, ob Sie SSM Agent Version 3.1.1141.0 oder höher heruntergeladen haben.

RPM

Überprüfen Sie das SSM Agent-Paket auf einem Linux-Server (v3.3.1802.0 und höher) wie folgt

1. Kopieren Sie den folgenden Schlüssel und speichern Sie ihn in einer Datei mit dem Namen `amazon-ssm-agent.gpg`.

> **Important**
>
> Der folgende öffentliche Schlüssel läuft am 15.07.2026 (15. Juli 2026) ab. Systems Manager wird in diesem Thema einen neuen öffentlichen Schlüssel veröffentlichen, bevor der alte abläuft. Wir empfehlen Ihnen, den RSS-Feed für dieses Thema zu abonnieren, um eine Benachrichtigung zu erhalten, wenn der neue Schlüssel verfügbar ist.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.22 (GNU/Linux)

mQINBGeRNq4BEACrlf5h6Pz+k+M+QCJJ2LfK7d2Tn9J8iJ9qBK2Vwvuxco1rpSO+
KEI3nTeysPuheximps8WOCADX4VlbsKxMZQLjQM4mA26m1Tiw9nAI4kod4bKjiuM
BMUTCD1wfnjH3zQi4kDUdbpfAEMiPgNLVLH85Wf+lhK+Zm+V38DYzLyVj03kX4wK
iG6RMoxzOBZa5gNsVq+j+oCUITGz/URxH713Rgo8WeoEegI0+7iCBLKg+PM0b7GV
2nzkwWJz796HdkqSg8BwXsYaLTrHxa2P1IpwPCisAkyO7gZaMd6Uj69dtMFO+V8a
Qee6b57qGuFKZw7h1Vvc85PbF1Gy/wNIpary57kUHBFUg1vYep/roJuEbJCq97r5
I2liLl4NAyrWb9r/TAVxlXvqM4iZUhxm8GAp0FywMdBr9ZECClKa5HxuVmlm0Wgl
TXoYTOZKeDg6ZoCvyhNxWneCNip74fohXymeFF5L/budhBwy5wuwSniOgTGLo/4C
VgZHWCcN+d0Q3bx/sl2QNqPg5/xzsxEtymXLdVdwLIsLdEQUnIvy8KTs5jol3Dwi
nnEEyhly6wdaw+qDOhkSOT/VnErrSMkYF8VJfa5GjhCBWKw9JVSkaP2CI/VHOgHM
MKROnulq0hRQBR7RmLYt98xu38BHJWMmF8Ga/HJuIxzD1VmkZOPvDDESUwARAQAB
tCdTU00gQWdlbnQgPHNzbS1hZ2VudC1zaWduZXJAYW1hem9uLmNvbT6JAj8EEwEC
ACkFAmeRNq4CGy8FCQLGmIAHCwkIBwMCAQYVCAIJCgsEFgIDAQIeAQIXgAAKCRBR
qOBQ0AUuXTdND/9qldQ1E3dYjBVXOnbhiUQL594bkS5VoEX7D4fZ5UMVZa5pGiz+
husnoRUS9rH1cSeq7aHJu9hSCMuMdvRpuoo0CwLB+7HtzJvAO2M01hcEkUYa6Qdj
njTzP0ZjnoenJmqF9SYmVqAI/VPa9mNQ1OJ+HQ3qh5i6w+FoWlVqEdXjZGrWijub
TqyN33i1Y26t7Os/x8I9fUeNx37y/7Kama8LTdtv9GhWiMVBg2IuVf27HCMYofrQ
m2uCGe61IhtsnhsYaYupmljl+6qgdiuCiS9BAsoIGtqTnu8lnKcGyGz6YnRszN+U
1bNE4w+UFpXWJF8ogpYcghJ06aW/LhjZnQSx3VliLdW8eOJzou41yWmiuL3ZY8eW
KAlD+7eYKS6N6fEJCeNO2VX2lcKtDfaOX+lqGIVyexKayMfpi+0frNzt/92YCpF5
3jkeS77vMMVqKIUiIp1OCGv3XsFpIr6Bt2c2throYPDoQL3zvq6vvG40BKeRQ4tT
Y+5vTc8MeNn3LdzTl9pusxTcKifrJq7f5FIsL2CpAX8uQ+Qz+XWsYQQ5PvyUDtOz
```

```
nU/MRZaP6HnqY42bzI9ZlKgXi9IE3MXIwoET9YyzFjkIDvat7SlB4uJCpeIqp/KM
OIrTMb7paGLYmBU6YqxNBkDWItNG7NeZzyhh/R/Qqb4vJaf4S+ZqD1RZXokCHAQQ
AQIABgUCZ5E2rwAKCRB90Jej2tf1/CdnD/46It+RNoE00TesZK5n2bijH5Eljw0E
4/UpMi1SV6t2zY7lIm7TcKNn18tynJNFqB6YXXOwSbBG/fbN2E9RaoUCZw23TmAv
amuHwrfsDqsHb7zzPF0bISYjqEDLQJj/gtEugUc6XY1dEpFSlWJIOvgryG04cFXI
uD2KY87ya4s1R+sEVAJ14K4RlUCiMmzJdR0NJNYJOwBi1gkLEp6jG86ttiG2U7fY
pE2ibV+c0GeIFq8PIzqqENsn9KBuRH5EcbdBwfnsj2XfM4aR3ZtRIdWXkKkdP9Rs
yU5dTF/Y7XPId5h8/gp00+DMlXFBinQ1jE7A7eDYviEFd1ba8P7dIom3Q3gzKiWu
KTGpnykShs5NvpQmvGUF6JqDHI4RK9s3kLqsNyZkhenJfRBrJ/45fQAuP4CRedkF
7PSfX0Xp7kDnKuyK6wEUEfXXrqmuLGDmigTXblO5qgdyMwkOLjiY9znBZbHoKs76
VplOoNgGnN19i3nuMcPf2npFICJv7kTIyn5Fh7pjWDCahl3U/PwoLjrrlEzpyStU
oXSZrK3kiAADEdSODXJl8KYU0Pb27JbRr1ZbWnxb+O39TOhtssstulkR0v+IDGDQ
rQE1b12sKgcNFSzInzWrNGu4S06WN8DYzlrTZ9aSHj+37ZqpXAevi8WOFXKPV3PA
E6+O8RI2451Dcg==
=aDkv
-----END PGP PUBLIC KEY BLOCK-----
```

2. Importieren Sie den öffentlichen Schlüssel in Ihren Schlüsselbund und notieren Sie den zurückgegebenen Schlüsselwert.

```
rpm --import amazon-ssm-agent.gpg
```

3. Überprüfen Sie den Fingerprint Es wird empfohlen, GPG zu verwenden, um den Fingerprint zu überprüfen, auch wenn Sie RPM verwenden, um das Installer-Paket zu überprüfen.

```
rpm -qa gpg-pubkey --qf '%{Description}' | gpg --with-fingerprint | grep -A 1
 "ssm-agent-signer@amazon.com"
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück:

```
pub  4096R/D0052E5D 2025-01-22 SSM Agent <ssm-agent-signer@amazon.com>
      Key fingerprint = 4855 A9E6 8332 16D6 A77D  8FE4 51A8 E050 D005 2E5D
```

Der Fingerabdruck sollte wie folgt aussehen.

```
4855 A9E6 8332 16D6 A77D 8FE4 51A8 E050 D005 2E5D
```

Wenn die Fingerabdruck-Zeichenfolge nicht übereinstimmt, installieren Sie den Agent nicht. Kontakt AWS -Support.

4. Überprüfen Sie die Installer-Paketsignatur. Achten Sie darauf, den durch die *agent-download-filename* Werte zu ersetzen, die Sie beim Herunterladen des Agenten angegeben haben, wie in der Tabelle weiter oben in diesem Thema aufgeführt.

   ```
   rpm --checksig agent-download-filename
   ```

   Zum Beispiel für die x86_64-Architektur auf Amazon Linux 2:

   ```
   rpm --checksig amazon-ssm-agent.rpm
   ```

   Dieser Befehl gibt etwa die folgende Ausgabe zurück.

   ```
   amazon-ssm-agent.rpm: rsa sha1 md5 OK
   ```

   Wenn pgp in der Ausgabe fehlt und Sie den öffentlichen Schlüssel importiert haben, ist der Agent nicht signiert. Wenn die Ausgabe die Phrase `NOT OK (MISSING KEYS: (MD5) key-id)` enthält, überprüfen Sie, ob Sie das Verfahren korrekt durchgeführt haben, und überprüfen Sie, ob Sie SSM Agent Version 3.1.1141.0 oder höher heruntergeladen haben. Wenn Sie weiterhin diese Antwort erhalten, wenden Sie sich an den Agenten Support und installieren Sie ihn nicht.

Überprüfen des SSM Agent-Paket auf einem Linux-Server (v3.3.1611.0 und älter)

Bevor Sie beginnen

Die Verfahren für GPG und RPM in diesem Abschnitt gelten für SSM Agent-Version 3.3.1611.0 und davor. Wir empfehlen immer die neueste Agent-Version zu verwenden. Weitere Informationen finden Sie unter Überprüfen des SSM Agent-Pakets auf einem Linux-Server (v3.3.1802.0 und höher). Wenn Sie jedoch einen Grund haben, weiterhin die Agent-Version 3.3.1611.0 oder früher zu verwenden, befolgen Sie stattdessen die folgenden Anweisungen.

GPG

Überprüfen Sie das SSM Agent-Paket auf einem Linux-Server (v3.3.1611.0 und älter) wie folgt

1. Kopieren Sie den folgenden öffentlichen Schlüssel und speichern Sie ihn in einer Datei mit dem Namen `amazon-ssm-agent.gpg`.

> **Important**
>
> Der unten gezeigte öffentliche Schlüssel ist am 17.02.2025 (17. Februar 2025) abgelaufen und funktioniert für Version 3.3.1611.0 und frühere Versionen bis 3.2.1542.0 und nur, wenn er zuvor zur Überprüfung der Agentensignatur verwendet wurde. Systems Manager wird in diesem Thema einen neuen öffentlichen Schlüssel veröffentlichen, bevor der alte abläuft. Wir empfehlen Ihnen, den RSS-Feed für dieses Thema zu abonnieren, um eine Benachrichtigung zu erhalten, wenn der neue Schlüssel verfügbar ist.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.22 (GNU/Linux)

mQENBGTtIoIBCAD2M1aoGIE0FXynAHM/jtuvdAVVaX3Q4ZejTqrX+Jq8ElAMhxyO
GzHu2CDtCYxtVxXK3unptLVt2kGgJwNbhYC393jDeZx5dCda4Nk2YXX1UK3P461i
axuuXRzMYvfM4RZn+7bJTu635tA07q9Xm6MGD4TCTvsjBfViOxbrxOg5ozWbJdSw
fSR8MwUrRfmFpAefRlYfCEuZ8FHywa9U6jLeWt2O/kqrZliJOAGjGzXtB7EZkqKb
faCCxikjjvhF1awdEqSK4DQorC/OvQc4I5kP5y2CJbtXvXO73QH2yE75JMDIIx9x
rOsIRUoSfK3UrWaOVuAnEEn5ueKzZNqGG1J1ABEBAAG0J1NTTSBBZ2VudCA8c3Nt
LWFnZW50LXNpZ25lckBhbWF6b24uY29tPokBPwQTAQIAKQUCZO0iggIbLwUJAsaY
gAcLCQgHAwIBBhUIAgkKCwQWAgMBAh4BAheAAAoJELwfSVyX3QTt+icH/A//tJsW
I+7Ay8FGJh8dJPNy++HIBjVSFdGNJFWNbw1Z8uZcazHEcUCH3FhW4CLQLTZ3OVPz
qvFwzDtRDVIN/Y9EGDhLMFvimrE+/z4olWsJ5DANf6BnX8I5UNIcRt5d8SWH1BEJ
2FWIBZFgKyTDI6XzRC5x4ahtgpOVAGeeKDehs+wh6Ga4W0/K4GsviP1Kyr+Ic2br
NAIq0q0IHyN1q9zam3Y0+jKwEuNmTj+Bjyzshyv/X8S0JWWoXJhkexkOvWeBYNNt
5wI4QcSteyfIzp6KlQF8q11Hzz9D9WaPfcBEYyhq7vLEARobkbQMBzpkmaZua241
0RaWG50HRvrgm4aJAhwEEAECAAYFAmTtIoMACgkQfdCXo9rX9fwwqBAAzkTgYJ38
sWgxpn7Ux/81F2BWR1sVkmP79i++fXyJlKI8xtcJFQZhzeUos69KBUCy7mgx5bYU
P7NA5o9DUbwz/QS0i1Cqm4+jtFlX0MXe4FikXcqfDPnnzN8mVB2H+fa43iHR1PuH
GgUWuNdxzSoIYRmLZXWmeN5YXPcmixlhLzcE2TOQn1mOKcu2fKdLtBQ8KiEkmjiu
naoLxnUcyk1zMhaha+LzEkQdOyasix0ggylN2ViWVnlmfy0niuXDxW0qZWPdLStF
OODiX3iqGmkH3rDfy6nvxxBR4GIs+MGD72fpWzzrINDgkGI2i2t1+0AX/mps3aTy
+ftlgrim8stYWB58XXDAb0vad06sNye5/zDzfr0I9HupJrTzFhaYJQjWPaSlINto
LDJnBXohiUIPRYRcy/k012oFHDWZHT3H6CyjK9UD5UlxA9H7dsJurANs6FOVRe+7
34uJyxDZ/W7zLG4AVG0zxibrUSoaJxwcOjVPVsQAlrwG/GTs7tcAccsJqbJ1Py/w
9AgJl8VU2qc8POsHNXk348gjP7C8PDnGMpZFzr9f5INctRushpiv7onX+aWJVX7T
n2uX/TP3LCyH/MsrNJrJOQnMYFRLQitciP0E+F+eA3v9CY6mDuyb8JSx5HuGGUsG
S4bKBOcA8vimEpwPoT8CE7fdsZ3Qkwdu+pw=
=zr5w
```

```
-----END PGP PUBLIC KEY BLOCK-----
```

2. Importieren Sie den öffentlichen Schlüssel in Ihren Schlüsselbund und notieren Sie den zurückgegebenen Schlüsselwert.

```
gpg --import amazon-ssm-agent.gpg
```

3. Überprüfen Sie den Fingerprint Achten Sie darauf, den Wert *key-value* durch den Wert aus dem vorherigen Schritt zu ersetzen. Es wird empfohlen, GPG zu verwenden, um den Fingerprint zu überprüfen, auch wenn Sie RPM verwenden, um das Installer-Paket zu überprüfen.

```
gpg --fingerprint key-value
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück.

```
pub   2048R/97DD04ED 2023-08-28 [expired: 2025-02-17]
      Key fingerprint = DE92 C7DA 3E56 E923 31D6 2A36 BC1F 495C 97DD 04ED
uid                  SSM Agent <ssm-agent-signer@amazon.com>
```

Der Fingerabdruck sollte wie folgt aussehen.

```
DE92 C7DA 3E56 E923 31D6 2A36 BC1F 495C 97DD 04ED
```

Wenn die Fingerabdruck-Zeichenfolge nicht übereinstimmt, installieren Sie den Agent nicht. Kontakt AWS -Support.

4. Laden Sie die Signaturdatei entsprechend der Architektur und dem Betriebssystem Ihrer Instance herunter.
5. Überprüfen Sie die Installer-Paketsignatur. Achten Sie darauf, das *signature-filename* und *agent-download-filename* durch die Werte zu ersetzen, die Sie beim Herunterladen der Signaturdatei und des Agenten angegeben haben, wie in der Tabelle weiter oben in diesem Thema aufgeführt.

```
gpg --verify signature-filename agent-download-filename
```

Zum Beispiel für die x86_64-Architektur auf Amazon Linux 2:

```
gpg --verify amazon-ssm-agent.rpm.sig amazon-ssm-agent.rpm
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück:

```
gpg: Signature made Fri 10 Jan 2025 01:54:18 AM UTC using RSA key ID 97DD04ED
gpg: Good signature from "SSM Agent <ssm-agent-signer@amazon.com>"
gpg: Note: This key has expired!
Primary key fingerprint: DE92 C7DA 3E56 E923 31D6  2A36 BC1F 495C 97DD 04ED
```

Wenn die Ausgabe die Bezeichnung `BAD signature`enthält, überprüfen Sie, ob Sie das Verfahren korrekt durchgeführt haben. Wenn Sie weiterhin diese Antwort erhalten, wenden Sie sich an den Agenten Support und installieren Sie ihn nicht. Die Vertrauens-Warnmeldung bedeutet nicht, dass die Signatur ungültig ist, sondern nur, dass Sie den öffentlichen Schlüssel nicht überprüft haben. Beachten Sie die Warnung zu vertrauenswürdigen Inhalten. Wenn die Ausgabe die Phrase `Can't check signature: No public key` enthält, überprüfen Sie, ob Sie SSM Agent Version 3.1.1141.0 oder höher heruntergeladen haben.

RPM

Überprüfen Sie das SSM Agent-Paket auf einem Linux-Server (v3.3.1611.0 und älter) wie folgt

1. Kopieren Sie den folgenden Schlüssel und speichern Sie ihn in einer Datei mit dem Namen `amazon-ssm-agent.gpg`.

> **Important**
>
> Der unten gezeigte öffentliche Schlüssel ist am 17.02.2025 (17. Februar 2025) abgelaufen und funktioniert für Version 3.3.1611.0 und frühere Versionen bis 3.2.1542.0 und nur, wenn er zuvor zur Überprüfung der Agentensignatur verwendet wurde. Systems Manager wird in diesem Thema einen neuen öffentlichen Schlüssel veröffentlichen, bevor der alte abläuft. Wir empfehlen Ihnen, den RSS-Feed für dieses Thema zu abonnieren, um eine Benachrichtigung zu erhalten, wenn der neue Schlüssel verfügbar ist.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.22 (GNU/Linux)

mQENBGTtIoIBCAD2M1aoGIE0FXynAHM/jtuvdAVVaX3Q4ZejTqrX+Jq8ElAMhxyO
```

```
GzHu2CDtCYxtVxXK3unptLVt2kGgJwNbhYC393jDeZx5dCda4Nk2YXX1UK3P461i
axuuXRzMYvfM4RZn+7bJTu635tA07q9Xm6MGD4TCTvsjBfViOxbrxOg5ozWbJdSw
fSR8MwUrRfmFpAefRlYfCEuZ8FHywa9U6jLeWt2O/kqrZliJOAGjGzXtB7EZkqKb
faCCxikjjvhF1awdEqSK4DQorC/OvQc4I5kP5y2CJbtXvXO73QH2yE75JMDIIx9x
rOsIRUoSfK3UrWaOVuAnEEn5ueKzZNqGG1J1ABEBAAG0J1NTTSBBZ2VudCA8c3Nt
LWFnZW50LXNpZ25lckBhbWF6b24uY29tPokBPwQTAQIAKQUCZO0iggIbLwUJAsaY
gAcLCQgHAwIBBhUIAgkKCwQWAgMBAh4BAheAAAoJELwfSVyX3QTt+icH/A//tJsW
I+7Ay8FGJh8dJPNy++HIBjVSFdGNJFWNbw1Z8uZcazHEcUCH3FhW4CLQLTZ3OVPz
qvFwzDtRDVIN/Y9EGDhLMFvimrE+/z4olWsJ5DANf6BnX8I5UNIcRt5d8SWH1BEJ
2FWIBZFgKyTDI6XzRC5x4ahtgpOVAGeeKDehs+wh6Ga4W0/K4GsviP1Kyr+Ic2br
NAIq0q0IHyN1q9zam3Y0+jKwEuNmTj+Bjyzshyv/X8S0JWWoXJhkexkOvWeBYNNt
5wI4QcSteyfIzp6KlQF8q11Hzz9D9WaPfcBEYyhq7vLEARobkbQMBzpkmaZua241
0RaWG50HRvrgm4aJAhwEEAECAAYFAmTtIoMACgkQfdCXo9rX9fwwqBAAzkTgYJ38
sWgxpn7Ux/81F2BWR1sVkmP79i++fXyJlKI8xtcJFQZhzeUos69KBUCy7mgx5bYU
P7NA5o9DUbwz/QS0i1Cqm4+jtFlX0MXe4FikXcqfDPnnzN8mVB2H+fa43iHR1PuH
GgUWuNdxzSoIYRmLZXWmeN5YXPcmixlhLzcE2TOQn1mOKcu2fKdLtBQ8KiEkmjiu
naoLxnUcyk1zMhaha+LzEkQdOyasix0ggylN2ViWVnlmfy0niuXDxW0qZWPdLStF
OODiX3iqGmkH3rDfy6nvxxBR4GIs+MGD72fpWzzrINDgkGI2i2t1+0AX/mps3aTy
+ftlgrim8stYWB58XXDAb0vad06sNye5/zDzfr0I9HupJrTzFhaYJQjWPaSlINto
LDJnBXohiUIPRYRcy/k012oFHDWZHT3H6CyjK9UD5UlxA9H7dsJurANs6FOVRe+7
34uJyxDZ/W7zLG4AVG0zxibrUSoaJxwcOjVPVsQAlrwG/GTs7tcAccsJqbJ1Py/w
9AgJl8VU2qc8POsHNXk348gjP7C8PDnGMpZFzr9f5INctRushpiv7onX+aWJVX7T
n2uX/TP3LCyH/MsrNJrJOQnMYFRLQitciP0E+F+eA3v9CY6mDuyb8JSx5HuGGUsG
S4bKBOcA8vimEpwPoT8CE7fdsZ3Qkwdu+pw=
=zr5w
-----END PGP PUBLIC KEY BLOCK-----
```

2. Importieren Sie den öffentlichen Schlüssel in Ihren Schlüsselbund und notieren Sie den zurückgegebenen Schlüsselwert.

```
rpm --import amazon-ssm-agent.gpg
```

3. Überprüfen Sie den Fingerprint Es wird empfohlen, GPG zu verwenden, um den Fingerprint zu überprüfen, auch wenn Sie RPM verwenden, um das Installer-Paket zu überprüfen.

```
rpm -qa gpg-pubkey --qf '%{Description}' | gpg --with-fingerprint | grep -A 1
 "ssm-agent-signer@amazon.com"
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück:

```
pub  2048R/97DD04ED 2023-08-28 SSM Agent <ssm-agent-signer@amazon.com>
      Key fingerprint = DE92 C7DA 3E56 E923 31D6 2A36 BC1F 495C 97DD 04ED
```

Der Fingerabdruck sollte wie folgt aussehen.

```
DE92 C7DA 3E56 E923 31D6 2A36 BC1F 495C 97DD 04ED
```

Wenn die Fingerabdruck-Zeichenfolge nicht übereinstimmt, installieren Sie den Agent nicht. Kontakt AWS -Support.

4. Überprüfen Sie die Installer-Paketsignatur. Achten Sie darauf, den durch die *agent-download-filename* Werte zu ersetzen, die Sie beim Herunterladen des Agenten angegeben haben, wie in der Tabelle weiter oben in diesem Thema aufgeführt.

```
rpm --checksig agent-download-filename
```

Zum Beispiel für die x86_64-Architektur auf Amazon Linux 2:

```
rpm --checksig amazon-ssm-agent.rpm
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück.

```
amazon-ssm-agent.rpm: rsa sha1 md5 OK
```

Wenn pgp in der Ausgabe fehlt und Sie den öffentlichen Schlüssel importiert haben, ist der Agent nicht signiert. Wenn die Ausgabe die Phrase `NOT OK (MISSING KEYS: (MD5) key-id)` enthält, überprüfen Sie, ob Sie das Verfahren korrekt durchgeführt haben, und überprüfen Sie, ob Sie SSM Agent Version 3.1.1141.0 oder höher heruntergeladen haben. Wenn Sie weiterhin diese Antwort erhalten, wenden Sie sich an den Agenten Support und installieren Sie ihn nicht.

Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux

Lesen Sie die folgenden Informationen, bevor Sie AWS Systems Manager Agent (SSM Agent) manuell auf einem Amazon Elastic Compute Cloud (Amazon EC2) Linux-Betriebssystem installieren.

⚠ Installation auf anderen Maschinentypen

Die Verfahren in diesem Abschnitt wurden speziell für EC2 Amazon-Instances entwickelt. Verwenden Sie für lokale Server, virtuelle Maschinen oder andere EC2 Umgebungen das `ssm-setup-cli` Tool wie unter So installieren Sie es SSM Agent auf Hybrid-Linux-Knoten beschrieben.

EC2 Die Verwendung von Installationsverfahren auf EC2 Nichtsystemen kann potenziell zu Sicherheitslücken führen. Das `ssm-setup-cli` Tool bietet zusätzliche Sicherheitsvorkehrungen für Computer, die keine EC2 Computer sind.

Wir empfehlen dringend, die Verwendung von Betriebssystemversionen zu vermeiden, die End-of-Life (EOL) erreicht haben. Betriebssystemanbieter, darunter auch, bieten AWS in der Regel keine Sicherheitspatches oder andere Updates für Versionen an, die EOL erreicht haben. Die weitere Verwendung eines EOL-Systems erhöht das Risiko, dass Upgrades, einschließlich Sicherheitsupdates, und andere Betriebsprobleme nicht durchgeführt werden können, erheblich. AWS testet die Systems Manager Manager-Funktionalität nicht auf Betriebssystemversionen, die EOL erreicht haben.

SSM AgentInstallationsdatei URLs

Sie können auf die Installationsdateien zugreifenSSM Agent, die in jeder kommerziellen Version gespeichert sind AWS-Region. Wir stellen auch Installationsdateien in einem global verfügbaren Amazon Simple Storage Service (Amazon S3)-Bucket bereit, den Sie als Alternative oder Sicherungsquelle für Dateien verwenden können.

Wenn Sie den Agenten manuell auf einer oder zwei Instances installieren, können Sie die Befehle in der von uns bereitgestellten Schnellinstallation verwenden, um Zeit zu sparen. Die in diesen Verfahren bereitgestellten Befehle können auch als Skripte über Benutzerdaten an EC2 Amazon-Instances übergeben werden.

Wenn Sie ein Skript oder eine Vorlage für die Installation des Agenten auf mehreren Instances erstellen, empfehlen wir Ihnen, die Installationsdateien in oder in der Nähe einer AWS-Region an Ihrem geografischen Standort zu verwenden. Bei Masseninstallationen kann dies die Geschwindigkeit Ihrer Downloads erhöhen und die Latenz verringern. In diesen Fällen empfehlen wir die Verwendung der Verfahren im Abschnitt Create custom installation commands (Erstellen benutzerdefinierter Installationsbefehle) in den Installationsthemen.

Amazon Machine Images mit dem Agenten vorinstalliert

SSM Agentist auf einigen Amazon Machine Images (AMIs) vorinstalliert, die von AWS bereitgestellt werden. Weitere Informationen finden Sie unter Finden Sie AMIs mit dem vorinstallierten SSM Agent.

Den Agenten auf dem neuesten Stand halten

Wenn Systems Manager neue Tools hinzugefügt oder Aktualisierungen an den vorhandenen Tools vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Tools und Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent in Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die SSM Agent-Versionshinweise-Seite in GitHub, um Benachrichtigungen über SSM Agent-Updates zu erhalten.

Auswahl Ihres Betriebssystems

Um das Verfahren zur manuellen Installation von SSM Agent unter dem angegebenen Betriebssystem anzuzeigen wählen sie einen Link aus der folgenden Liste aus:

> **Note**
>
> Eine Liste der unterstützten Versionen der folgenden Betriebssysteme finden Sie unter Unterstützte Betriebssysteme für Systems Manager.

- AlmaLinux
- Amazon Linux 2 und Amazon Linux 2023
- Debian Server
- Oracle Linux
- Red Hat Enterprise Linux
- Rocky Linux
- Ubuntu Server

Deinstallieren des SSM Agent von Linux-Instances

Verwenden Sie den Paketmanager für Ihr Betriebssystem, um SSM Agent von Linux-Instances zu deinstallieren. Je nach Betriebssystem sieht der Deinstallationsbefehl wie folgt aus:

```
sudo dpkg -r amazon-ssm-agent
```

Manuelle Installation von SSM Agent auf AlmaLinux-Instances

Verwenden Sie die Informationen in diesem Abschnitt, um Sie bei der manuellen Installation oder Neuinstallation SSM Agent auf einer AlmaLinux Instanz zu unterstützen.

Bevor Sie beginnen

Beachten Sie vor SSM Agent der Installation auf einer AlmaLinux Instance Folgendes:

- Stellen Sie sicher, dass Python 3 auf Ihrer AlmaLinux Instanz installiert ist. Dies ist erforderlich, damit SSM Agent richtig funktioniert.
- Wichtige Informationen für die Installation von SSM Agent in allen Linux-Betriebssystemen finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux.

Themen

- Schnellinstallationsbefehle für SSM Agent on AlmaLinux
- Erstellen Sie benutzerdefinierte Agenteninstallationsbefehle für AlmaLinux Ihre Region

Schnellinstallationsbefehle für SSM Agent on AlmaLinux

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

Bevor Sie beginnen

Beachten Sie vor SSM Agent der Installation auf einer AlmaLinux Instanz Folgendes:

- Stellen Sie sicher, dass Python 3 auf Ihrer AlmaLinux Instanz installiert ist. Dies ist erforderlich, damit SSM Agent richtig funktioniert.

Zur Installation SSM Agent auf AlmaLinux

1. Stellen Sie mit Ihrer bevorzugten Methode, AlmaLinux z. B. SSH, eine Connect zu Ihrer Instance her.
2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

> **Note**
>
> Obwohl URLs die folgenden Befehle ein `ec2-downloads-windows` Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für AlmaLinux.

x86_64-Instances

```
sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
```

ARM64-Instances

```
sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_arm64/amazon-ssm-agent.rpm
```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

```
sudo systemctl status amazon-ssm-agent
```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
   Active: active (running) since Tue 2025-04-19 16:40:41 UTC; 9s ago
 Main PID: 4898 (amazon-ssm-agen)
    Tasks: 14 (limit: 4821)
   Memory: 34.6M
   CGroup: /system.slice/amazon-ssm-agent.service
           ##4898 /usr/bin/amazon-ssm-agent
           ##4954 /usr/bin/ssm-agent-worker
            --truncated--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
   Active: inactive (dead) since Tue 2025-04-19 16:42:05 UTC; 2s ago
            --truncated--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen Sie benutzerdefinierte Agenteninstallationsbefehle für AlmaLinux Ihre Region

Wenn Sie SSM Agent in mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallationsbefehle für SSM Agent on AlmaLinux weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie den folgenden Befehl *`region`* durch Ihre eigenen Informationen. Eine Liste der unterstützten *`region`* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende Beispiel an.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

ARM64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende Beispiel an.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Manuelles Installieren von SSM Agent auf Instances von Amazon Linux 2 und Amazon Linux 2023

In den meisten Fällen ist AWS Systems Manager Agent Amazon Machine Images (AMIsSSM Agent) für Amazon Linux 2 und Amazon Linux 2023, die von AWS bereitgestellt werden, standardmäßig vorinstalliert. Weitere Informationen finden Sie unter Finden Sie AMIs mit dem vorinstallierten SSM Agent.

Für den Fall, dass SSM Agent auf einer neuen Instance von Amazon Linux 2 oder Amazon Linux 2023 nicht vorinstalliert ist, oder wenn Sie den Agenten manuell neu installieren müssen, verwenden Sie die Informationen auf dieser Seite, um Ihnen zu helfen.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer Instance von Amazon Linux 2 oder Amazon Linux 2023 beachten Sie bitte Folgendes:

- Wichtige Informationen für die Installation von SSM Agent in allen Linux-Betriebssystemen finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux.
- Wenn Sie einen `yum`-Befehl verwenden, um SSM Agent auf einem verwalteten Knoten zu aktualisieren, nachdem der Agent mit dem SSM-Dokument `AWS-UpdateSSMAgent` installiert oder aktualisiert wurde, sehen Sie möglicherweise die folgende Meldung: `"Warning: RPMDB altered outside of yum."` Diese Meldung wird erwartet und kann ignoriert werden.

Themen

- Schnellinstallations-Befehle für SSM Agent auf Amazon Linux 2 oder Amazon Linux 2023
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für Amazon Linux 2 oder Amazon Linux 2023 in Ihrer Region

Schnellinstallations-Befehle für SSM Agent auf Amazon Linux 2 oder Amazon Linux 2023

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

So installieren Sie SSM Agent schnell mit Befehlen zum Kopieren und Einfügen auf Amazon Linux 2 oder Amazon Linux 2023

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung zu Ihrer Amazon-Linux-2- oder Amazon-Linux-2023-Instance her.
2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

   > **Note**
   >
   > Obwohl URLs die folgenden Befehle ein `ec2-downloads-windows` Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für Amazon Linux 2 und Amazon Linux 2023.

   x86_64

   ```
   sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_amd64/amazon-ssm-agent.rpm
   ```

   ARM64

   ```
   sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_arm64/amazon-ssm-agent.rpm
   ```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

   ```
   sudo systemctl status amazon-ssm-agent
   ```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
amazon-ssm-agent.service - amazon-ssm-agent
Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: enabled)
Active: active (running) since Wed 2021-10-20 19:09:29 UTC; 4min 6s ago
            --truncated--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
amazon-ssm-agent.service - amazon-ssm-agent
Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: enabled)
Active: inactive (dead) since Wed 2021-10-20 22:16:41 UTC; 18s ago
            --truncated--
```

Um den Agenten in diesen Fällen zu aktivieren, führen Sie den folgenden Befehl aus.

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für Amazon Linux 2 oder Amazon Linux 2023 in Ihrer Region

Wenn Sie SSM Agent in mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent auf Amazon Linux 2 oder Amazon Linux 2023 weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie den Befehl im folgenden Befehl *region* durch Ihre eigenen Informationen. Eine Liste der unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende Beispiel an.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

ARM64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende Beispiel an.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Manuelle Installation von SSM Agent auf CentOS Stream-Instances

Die Amazon Machine Images (AMIs) CentOS Stream dafür, die von bereitgestellt werden, sind standardmäßig AWS nicht mit AWS Systems Manager Agent (SSM Agent) vorinstalliert. Eine Liste von AWS -verwalteten AMIs, auf denen der Agent möglicherweise vorinstalliert ist, finden Sie unter Finden Sie AMIs mit dem vorinstallierten SSM Agent.

Verwenden Sie die Informationen in diesem Abschnitt, um SSM Agent manuell auf einer CentOS Stream-Instance zu installieren oder neu zu installieren.

Bevor Sie beginnen

Vor der Installation von SSM Agent in einer CentOS Stream-Instance beachten Sie bitte Folgendes:

- Wichtige Informationen, die für die Installation von SSM Agent in allen Linux-Betriebssystemen gelten, finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux.

Themen

- Schnellinstallations-Befehle für SSM Agent in CentOS Stream
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für CentOS Stream in Ihrer Region

Schnellinstallations-Befehle für SSM Agent in CentOS Stream

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

Installieren Sie den SSM Agent auf CentOS Stream wie folgt

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer CentOS Stream-Instance her.
2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

   > **Note**
   >
   > Obwohl URLs die folgenden Befehle ein `ec2-downloads-windows` Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien fürCentOS Stream.

   x86_64-Instances

   ```
   sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_amd64/amazon-ssm-agent.rpm
   ```

   ARM64-Instances

   ```
   sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_arm64/amazon-ssm-agent.rpm
   ```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

```
sudo systemctl status amazon-ssm-agent
```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
   Active: active (running) since Tue 2025-04-19 16:40:41 UTC; 9s ago
 Main PID: 4898 (amazon-ssm-agen)
    Tasks: 14 (limit: 4821)
   Memory: 34.6M
   CGroup: /system.slice/amazon-ssm-agent.service
           ##4898 /usr/bin/amazon-ssm-agent
           ##4954 /usr/bin/ssm-agent-worker
            --truncated--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
   Active: inactive (dead) since Tue 2025-04-19 16:42:05 UTC; 2s ago
            --truncated
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für CentOS Stream in Ihrer Region

Wenn Sie SSM Agent in mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent in CentOS Stream weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie den Befehl im folgenden Befehl *region* durch Ihre eigenen Informationen. Eine Liste der unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende Beispiel an.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

ARM64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende Beispiel an.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Manuelle Installation von SSM Agent auf Debian Server-Instances

Die Amazon Machine Images (AMIs) Debian Server dafür, die von bereitgestellt werden, sind standardmäßig AWS nicht mit AWS Systems Manager Agent (SSM Agent) vorinstalliert. Eine Liste von AWS -verwalteten AMIs, auf denen der Agent möglicherweise vorinstalliert ist, finden Sie unter Finden Sie AMIs mit dem vorinstallierten SSM Agent.

Verwenden Sie die Informationen in diesem Abschnitt, um SSM Agent manuell auf einer Debian Server-Instance zu installieren oder neu zu installieren.

Bevor Sie beginnen

Vor der Installation von SSM Agent in einer Debian Server-Instance beachten Sie bitte Folgendes:

- Wichtige Informationen, die für die Installation von SSM Agent in allen Linux-Betriebssystemen gelten, finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux.

Themen

- Schnellinstallations-Befehle für SSM Agent in Debian Server
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für Debian Server in Ihrer Region

Schnellinstallations-Befehle für SSM Agent in Debian Server

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

Installieren Sie den SSM Agent auf Debian Server wie folgt

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer Debian Server-Instance her.
2. Führen Sie den folgenden Befehl aus, um ein temporäres Verzeichnis auf der Instance zu erstellen.

   ```
   mkdir /tmp/ssm
   ```

3. Führen Sie den folgenden Befehl aus, um in das temporäre Verzeichnis zu wechseln.

   ```
   cd /tmp/ssm
   ```

4. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

> **Note**
>
> Obwohl URLs die folgenden Befehle ein `ec2-downloads-windows` Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien fürDebian Server. Für Debian Server 8 wird nur die x86_64-Architektur unterstützt.

x86_64-Instances

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/
debian_amd64/amazon-ssm-agent.deb
```

ARM64-Instances

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/
debian_arm64/amazon-ssm-agent.deb
```

5. Führen Sie den folgenden Befehl aus.

```
sudo dpkg -i amazon-ssm-agent.deb
```

6. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

```
sudo systemctl status amazon-ssm-agent
```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/lib/systemd/system/amazon-ssm-agent.service; enabled; vendor
   Active: active (running) since Tue 2025-04-19 16:25:03 UTC; 4s ago
 Main PID: 628 (amazon-ssm-agen)
   CGroup: /system.slice/amazon-ssm-agent.service
           ##628 /usr/bin/amazon-ssm-agent
           ##650 /usr/bin/ssm-agent-worker
            --truncated--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/lib/systemd/system/amazon-ssm-agent.service; enabled; vendor
   Active: inactive (dead) since Tue 2025-04-19 16:26:30 UTC; 5s ago
 Main PID: 628 (code=exited, status=0/SUCCESS)
            --truncated--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für Debian Server in Ihrer Region

Wenn Sie SSM Agent in mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent in Debian Server weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie den Befehl im folgenden Befehl *`region`* durch Ihre eigenen Informationen. Eine Liste der unterstützten *`region`* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_amd64/amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

Sehen Sie sich das folgende Beispiel an.

```
wget https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/debian_amd64/amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

ARM64

```
wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_arm64/amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

Sehen Sie sich das folgende Beispiel an.

```
wget https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/debian_arm64/amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

Manuelle Installation von SSM Agent auf Oracle Linux-Instances

Die Amazon Machine Images (AMIs) dafürOracle Linux, die von bereitgestellt werden, sind standardmäßig AWS nicht mit vorinstalliertem AWS Systems Manager Agent (SSM Agent) ausgestattet. Eine Liste von AWS -verwalteten AMIs, auf denen der Agent möglicherweise vorinstalliert ist, finden Sie unter Finden Sie AMIs mit dem vorinstallierten SSM Agent.

Verwenden Sie die Informationen in diesem Abschnitt, um SSM Agent manuell auf einer Oracle Linux-Instance zu installieren oder neu zu installieren.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer Oracle Linux-Instance beachten Sie bitte Folgendes:

- Wichtige Informationen für die Installation von SSM Agent in allen Linux-Betriebssystemen finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux.
- Wenn Sie einen `yum`-Befehl verwenden, um SSM Agent auf einem verwalteten Knoten zu aktualisieren, nachdem der Agent mit dem SSM-Dokument `AWS-UpdateSSMAgent` installiert oder aktualisiert wurde, sehen Sie möglicherweise die folgende Meldung: `"Warning: RPMDB altered outside of yum."` Diese Meldung wird erwartet und kann ignoriert werden.

Themen

- Schnellinstallations-Befehle für SSM Agent in Oracle Linux
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für Oracle Linux in Ihrer Region

Schnellinstallations-Befehle für SSM Agent in Oracle Linux

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

So installieren Sie SSM Agent schnell mit Befehlen zum Kopieren und Einfügen auf Oracle Linux

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer Oracle Linux-Instance her.
2. Kopieren Sie den folgenden Befehl und führen Sie ihn auf der Instance aus.

> **Note**
>
> Obwohl die URL im folgenden Befehl ein `ec2-downloads-windows`-Verzeichnis enthält, sind dies die richtigen globalen Installationsdateien für Oracle Linux.

x86_64

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

```
sudo systemctl status amazon-ssm-agent
```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
amazon-ssm-agent.service - amazon-ssm-agent
Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: enabled)
Active: active (running) since Wed 2021-10-20 19:09:29 UTC; 4min 6s ago
            --truncated--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
amazon-ssm-agent.service - amazon-ssm-agent
Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: enabled)
Active: inactive (dead) since Wed 2021-10-20 22:16:41 UTC; 18s ago
            --truncated--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für Oracle Linux in Ihrer Region

Wenn Sie SSM Agent in mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent in Oracle Linux weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie den Befehl im folgenden Befehl *region* durch Ihre eigenen Informationen. Eine Liste der unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende Beispiel an.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Manuelle Installation von SSM Agent auf Red Hat Enterprise Linux-Instances

Die von bereitgestellten Amazon Machine Images Red Hat Enterprise Linux (AMIsRHEL) for () sind standardmäßig AWS nicht mit AWS Systems Manager Agent (SSM Agent) vorinstalliert. Eine Liste der AWS verwalteten Programme, AMIs auf denen der Agent möglicherweise vorinstalliert ist, finden Sie unter. Finden Sie AMIs mit dem vorinstallierten SSM Agent

Verwenden Sie die Informationen in diesem Abschnitt, um SSM Agent manuell auf einer RHEL-Instance zu installieren oder neu zu installieren.

Bevor Sie beginnen

Vor der Installation von SSM Agent in einer RHEL-Instance beachten Sie bitte Folgendes:

- Wichtige Informationen, die für die Installation von SSM Agent in allen Linux-Betriebssystemen gelten, finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux.

- Wenn Sie einen `yum`-Befehl verwenden, um SSM Agent auf einem verwalteten Knoten zu aktualisieren, nachdem der Agent mit dem SSM-Dokument `AWS-UpdateSSMAgent` installiert oder aktualisiert wurde, sehen Sie möglicherweise die folgende Meldung: `"Warning: RPMDB altered outside of yum."` Diese Meldung wird erwartet und kann ignoriert werden.

Themen

- Installieren von SSM Agent unter RHEL 8.x, 9.x und 10.x
- So installieren Sie SSM Agent auf RHEL 7.x

Installieren von SSM Agent unter RHEL 8.x, 9.x und 10.x

Bei den von bereitgestellten Amazon Machine Images (AMIs) für RHEL 8 und 9 ist AWS Systems Manager Agent (SSM Agent) AWS nicht standardmäßig vorinstalliert. Verwenden Sie die Informationen auf dieser Seite, um den Agenten auf RHEL-8- und 9-Instances zu installieren oder neu zu installieren.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer RHEL-Instance (Version 8, 9 oder 10) beachten Sie bitte Folgendes:

- Auf der RHEL-Instance (Version 8, 9 oder 10) muss Python 2 oder Python 3 installiert sein. Dies ist erforderlich, damit SSM Agent richtig funktioniert.

Themen

- Schnellinstallations-Befehle für SSM Agent unter RHEL 8, 9 oder 10
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für RHEL 8, 9 und 10 in Ihrer Region

Schnellinstallations-Befehle für SSM Agent unter RHEL 8, 9 oder 10

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

Installieren Sie SSM Agent unter RHEL 8.x, 9.x oder 10.x wie folgt:

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer RHEL-Instance (Version 8, 9 oder 10) her.

2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

> **Note**
>
> Obwohl URLs die folgenden Befehle ein `ec2-downloads-windows` Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für RHEL 8, 9 und 10.

x86_64-Instances

```
sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
```

ARM64-Instances

```
sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_arm64/amazon-ssm-agent.rpm
```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

```
sudo systemctl status amazon-ssm-agent
```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
   Active: active (running) since Tue 2025-04-19 16:40:41 UTC; 9s ago
 Main PID: 4898 (amazon-ssm-agen)
    Tasks: 14 (limit: 4821)
   Memory: 34.6M
   CGroup: /system.slice/amazon-ssm-agent.service
           ##4898 /usr/bin/amazon-ssm-agent
           ##4954 /usr/bin/ssm-agent-worker
            --truncated--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
   Active: inactive (dead) since Tue 2025-04-19 16:42:05 UTC; 2s ago
            --truncated--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für RHEL 8, 9 und 10 in Ihrer Region

Wenn Sie SSM Agent in mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent unter RHEL 8, 9 oder 10 weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie den Befehl im folgenden Befehl *`region`* durch Ihre eigenen Informationen. Eine Liste der unterstützten *`region`* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende Beispiel an.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

ARM64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende Beispiel an.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

So installieren Sie SSM Agent auf RHEL 7.x

Bei den von bereitgestellten Amazon Machine Images (AMIs) für RHEL 7 ist AWS Systems Manager Agent (SSM Agent) AWS nicht standardmäßig vorinstalliert. Verwenden Sie die Informationen auf dieser Seite, um den Agenten auf RHEL-7-Instances zu installieren oder neu zu installieren.

Themen

- Schnellinstallations-Befehle für SSM Agent auf RHEL 7
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für RHEL 7 in Ihrer Region

Schnellinstallations-Befehle für SSM Agent auf RHEL 7

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

So installieren Sie SSM Agent auf RHEL 7.x

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer RHEL-7-Instance her.

2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

> **Note**
>
> Obwohl URLs die folgenden Befehle ein `ec2-downloads-windows` Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für RHEL 7.

x86_64-Instances

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
```

ARM64-Instances

```
sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_arm64/amazon-ssm-agent.rpm
```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

```
sudo systemctl status amazon-ssm-agent
```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: disabled)
   Active: active (running) since Tue 2025-04-19 16:47:36 UTC; 22s ago
 Main PID: 1342 (amazon-ssm-agen)
   CGroup: /system.slice/amazon-ssm-agent.service
           ##1342 /usr/bin/amazon-ssm-agent
           ##1362 /usr/bin/ssm-agent-worker
            --truncated--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendor
 preset: disabled)
   Active: inactive (dead) since Tue 2025-04-19 16:48:56 UTC; 5s ago
  Process: 1342 ExecStart=/usr/bin/amazon-ssm-agent (code=exited, status=0/SUCCESS)
 Main PID: 1342 (code=exited, status=0/SUCCESS)
           --truncated--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für RHEL 7 in Ihrer Region

Wenn Sie SSM Agent in mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent auf RHEL 7 weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie den Befehl im folgenden Befehl *`region`* durch Ihre eigenen Informationen. Eine Liste der unterstützten *`region`* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende Beispiel an.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

ARM64

```
sudo yum install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende Beispiel an.

```
sudo yum install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Manuelle Installation von SSM Agent auf Rocky Linux-Instances

Die Amazon Machine Images (AMIs) Rocky Linux dafür, die von bereitgestellt werden, sind standardmäßig AWS nicht mit AWS Systems Manager Agent (SSM Agent) vorinstalliert. Eine Liste von AWS -verwalteten AMIs, auf denen der Agent möglicherweise vorinstalliert ist, finden Sie unter Finden Sie AMIs mit dem vorinstallierten SSM Agent.

Verwenden Sie die Informationen in diesem Abschnitt, um SSM Agent manuell auf einer Rocky Linux-Instance zu installieren oder neu zu installieren.

Bevor Sie beginnen

Vor der Installation von SSM Agent in einer Rocky Linux-Instance beachten Sie bitte Folgendes:

- Wichtige Informationen, die für die Installation von SSM Agent in allen Linux-Betriebssystemen gelten, finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux.

Themen

- Schnellinstallations-Befehle für SSM Agent in Rocky Linux
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für Rocky Linux in Ihrer Region

Schnellinstallations-Befehle für SSM Agent in Rocky Linux

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

Bevor Sie beginnen

Vor der Installation von SSM Agent in einer Rocky Linux-Instance beachten Sie bitte Folgendes:

- Stellen Sie sicher, dass Python 2 oder Python 3 auf Ihrer Rocky Linux-Instance installiert ist. Dies ist erforderlich, damit SSM Agent richtig funktioniert.

Installieren Sie den SSM Agent auf Rocky Linux wie folgt

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer Rocky Linux-Instance her.
2. Kopieren Sie den Befehl für die Architektur Ihrer Instance und führen Sie ihn auf der Instance aus.

   > **Note**
   >
   > Obwohl URLs die folgenden Befehle ein `ec2-downloads-windows` Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien fürRocky Linux.

   x86_64-Instances

   ```
   sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_amd64/amazon-ssm-agent.rpm
   ```

   ARM64-Instances

   ```
   sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
   latest/linux_arm64/amazon-ssm-agent.rpm
   ```

3. (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

```
sudo systemctl status amazon-ssm-agent
```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
   Active: active (running) since Tue 2025-04-19 16:40:41 UTC; 9s ago
 Main PID: 4898 (amazon-ssm-agen)
    Tasks: 14 (limit: 4821)
   Memory: 34.6M
   CGroup: /system.slice/amazon-ssm-agent.service
           ##4898 /usr/bin/amazon-ssm-agent
           ##4954 /usr/bin/ssm-agent-worker
            --truncated--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/etc/systemd/system/amazon-ssm-agent.service; enabled; vendo>
   Active: inactive (dead) since Tue 2025-04-19 16:42:05 UTC; 2s ago
            --truncated--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für Rocky Linux in Ihrer Region

Wenn Sie SSM Agent in mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent in Rocky Linux weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie den Befehl im folgenden Befehl *`region`* durch Ihre eigenen Informationen. Eine Liste der unterstützten *`region`* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_amd64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende Beispiel an.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_amd64/amazon-ssm-agent.rpm
```

ARM64

```
sudo dnf install -y https://s3.region.amazonaws.com/amazon-ssm-region/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende Beispiel an.

```
sudo dnf install -y https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
linux_arm64/amazon-ssm-agent.rpm
```

Manuelle Installation von SSM Agent auf SUSE Linux Enterprise Server-Instances

In den meisten Fällen sind die von bereitgestellten Amazon Machine Images SUSE Linux Enterprise Server (AMIsSLES) für () AWS standardmäßig mit AWS Systems Manager Agent (SSM Agent)

vorinstalliert. Weitere Informationen finden Sie unter Finden Sie AMIs mit dem vorinstallierten SSM Agent.

Für den Fall, dass SSM Agent auf einer neuen SLES-Instance nicht vorinstalliert ist, oder wenn Sie den Agenten manuell neu installieren müssen, verwenden Sie die Informationen auf dieser Seite, um Ihnen zu helfen.

Bevor Sie beginnen

Vor der Installation von SSM Agent in einer SLES-Instance beachten Sie bitte Folgendes:

- Wichtige Informationen, die für die Installation von SSM Agent in allen Linux-Betriebssystemen gelten, finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux.

Themen

- Schnellinstallations-Befehle für SSM Agent in SLES
- Erstellen von benutzerdefinierten Agent-Installationsbefehlen für SLES in Ihrer Region

Schnellinstallations-Befehle für SSM Agent in SLES

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

So installieren Sie SSM Agent schnell mit Befehlen zum Kopieren und Einfügen auf SLES

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer SLES-Instance her.
2. Option 1: Benutzen Sie einen `zypper`-Befehl:

   - Führen Sie den folgenden Befehl aus:

     ```
     sudo zypper install amazon-ssm-agent
     ```

   - Geben Sie `y` als Reaktion auf alle Eingabeaufforderungen ein.

   Option 2: Benutzen Sie einen `rpm`-Befehl.

   - Erstellen Sie ein temporäres Verzeichnis auf der Instance.

```
mkdir /tmp/ssm
```

- Ändern Sie das temporäre Verzeichnis.

```
cd /tmp/ssm
```

- Führen Sie nacheinander die folgenden Befehle aus, um das SSM Agent-Installationsprogramm herunterzuladen und es auszuführen.

> **Note**
>
> Obwohl URLs die folgenden Befehle ein `ec2-downloads-windows` Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien fürSLES.

x86_64-Instances:

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_amd64/
amazon-ssm-agent.rpm
```

ARM64-Instances:

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_arm64/
amazon-ssm-agent.rpm
```

- Führen Sie den folgenden Befehl aus.

```
sudo rpm --install amazon-ssm-agent.rpm
```

- (Empfohlen) Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der Agent ausgeführt wird.

```
sudo systemctl status amazon-ssm-agent
```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# amazon-ssm-agent.service - amazon-ssm-agent
```

```
 Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; enabled;
 vendor preset: disabled)
 Active: active (running) since Mon 2025-02-21 23:13:28 UTC; 7s ago
 Main PID: 2102 (amazon-ssm-agen)
 Tasks: 15 (limit: 512)
 CGroup: /system.slice/amazon-ssm-agent.service
 ##2102 /usr/sbin/amazon-ssm-agent
 ##2107 /usr/sbin/ssm-agent-worker
           --truncated--
```

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

```
# amazon-ssm-agent.service - amazon-ssm-agent
   Loaded: loaded (/usr/lib/systemd/system/amazon-ssm-agent.service; disabled;
 vendor preset: disabled)
   Active: inactive (dead)
           --truncated--
```

Um den Agent in diesen Fällen zu aktivieren, führen Sie die folgenden Befehle aus.

```
sudo systemctl enable amazon-ssm-agent
```

```
sudo systemctl start amazon-ssm-agent
```

Erstellen von benutzerdefinierten Agent-Installationsbefehlen für SLES in Ihrer Region

Wenn Sie SSM Agent in mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent in SLES weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie den Befehl im folgenden Befehl *region* durch Ihre eigenen Informationen. Eine Liste der unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

x86_64

```
wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/linux_amd64/amazon-ssm-agent.rpm
```

```
sudo rpm --install amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende Beispiel an.

```
wget https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/linux_amd64/amazon-ssm-agent.rpm
```

```
sudo rpm --install amazon-ssm-agent.rpm
```

ARM64

```
wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/linux_arm64/amazon-ssm-agent.rpm
```

```
sudo rpm --install amazon-ssm-agent.rpm
```

Sehen Sie sich das folgende Beispiel an.

```
wget https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/linux_arm64/amazon-ssm-agent.rpm
```

```
sudo rpm --install amazon-ssm-agent.rpm
```

Manuelle Installation von SSM Agent auf Ubuntu Server-Instances

> **Important**
>
> Stellen Sie vor der Installation von SSM Agent auf einer 64-Bit-Version von Ubuntu Server sicher, dass Sie die richtigen Installationstools verwenden. Beginnend mit Amazon Machine Images (AMIs), die mit 20180627 identifiziert werden, ist SSM Agent unter Verwendung von Snap-Paketen auf Version 16.04 vorinstalliert. Auf Instances, die aus früheren AMIs erstellt wurden, muss SSM Agent mit Deb-Installationsprogramm-Paketen installiert werden. Weitere Informationen finden Sie unter Bestimmen der korrekten SSM Agent-Version zur Installation auf 64-Bit-Instances von Ubuntu Server 16.04.

In den meisten Fällen sind die von AWS bereitgestellten Amazon Machine Images (AMIs) für Ubuntu Server standardmäßig mit AWS Systems Manager Agent (SSM Agent) vorinstalliert. Weitere Informationen finden Sie unter Finden Sie AMIs mit dem vorinstallierten SSM Agent.

Für den Fall, dass SSM Agent auf einer neuen Ubuntu Server-Instance nicht vorinstalliert ist, oder wenn Sie den Agenten manuell neu installieren müssen, verwenden Sie die Informationen in diesem Abschnitt, um Ihnen zu helfen.

Bevor Sie beginnen

Vor der Installation von SSM Agent auf einer Ubuntu Server-Instance beachten Sie bitte Folgendes:

- Wichtige Informationen für die Installation von SSM Agent in allen Linux-Betriebssystemen finden Sie unter Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux.

Themen

- Installieren von SSM Agent unter Ubuntu Server 16.04 LTS 64-Bit (Snap), 18.04, 20.04, 22.04 LTS, 23.10, 24.04 LTS, 24.0 und 25.04
- Installieren von SSM Agent unter Ubuntu Server 16.04 64-Bit (deb)
- Installieren von SSM Agent unter Ubuntu Server 16.04 32-Bit

- Bestimmen der korrekten SSM Agent-Version zur Installation auf 64-Bit-Instances von Ubuntu Server 16.04

Installieren von SSM Agent unter Ubuntu Server 16.04 LTS 64-Bit (Snap), 18.04, 20.04, 22.04 LTS, 23.10, 24.04 LTS, 24.0 und 25.04

Bevor Sie beginnen

Vor der Installation von SSM Agent unter Ubuntu Server 16.04 LTS 64-Bit (Snap), 18.04, 20.04, 22.04 LTS, 23.10, 24.04 LTS, 24.0 und 25.04 beachten Sie bitte Folgendes:

Installation von Version 16.04 durch Snaps oder Deb-Installationsprogramme

Auf Ubuntu Server 16.04, wird der SSM Agent, abhängig von der Version des 16.04-AMI, entweder mit Snaps oder mit deb-Installationspaketen installiert.

Speicherorte von SSM Agent-Installationsprogramm-Dateien

Unter Ubuntu Server 16.04 LTS 64-Bit (Snap), 18.04, 20.04, 22.04 LTS, 23.10, 24.04 LTS, 24.0 und 25.04 werden SSM Agent-Installationsprogramm-Dateien, einschließlich Agentenbinär- und Konfigurationsdateien, im folgenden Verzeichnis gespeichert: `/snap/amazon-ssm-agent/current/`. Wenn Sie Änderungen an einer Konfigurationsdatei in diesem Verzeichnis vornehmen, müssen Sie dies Datei aus dem Verzeichnis `/snap` in das Verzeichnis `/etc/amazon/ssm/` kopieren. Protokoll- und Bibliotheksdateien wurden nicht geändert (`/var/lib/amazon/ssm`, `/var/log/amazon/ssm`).

Verwenden des Snap-`candidate`-Kanals

Der Kandidat-Kanal im Snap Store enthält die neueste Version von SSM Agent (einschließlich aller neuesten Fehlerbehebungen); nicht den stabilen Kanal. Weitere Informationen zu den Unterschieden zwischen den Kanälen Candidate und Stable finden Sie unter Risk-levels unter https://snapcraft. io/docs/channels.

Wenn Sie SSM Agent-Versionsinformationen auf dem Kandidatenkanal verfolgen, führen Sie auf Ihren Ubuntu Server-Instances der Versionen 20.04, 18.04 und 16.04 LTS 64-Bit den folgenden Befehl aus.

```
sudo snap switch --channel=candidate amazon-ssm-agent
```

Für Version 18.04 und höher empfohlene Snaps

Unter Ubuntu Server 18.04, 20.04, 22.04 LTS, 23.10, 24.04 LTS, 24.0 und 25.04 empfehlen wir, nur Snaps zu verwenden. Stellen Sie außerdem sicher, dass nur eine Instance des Agenten auf Ihren Instances installiert ist und ausgeführt wird. Wenn Sie SSM Agent ohne Snaps verwenden möchten, deinstallieren Sie den SSM Agent. Installieren Sie dann SSM Agent als Debian-Paket gemäß den Anweisungen zum Installieren von SSM Agent unter Ubuntu Server 16.04 (deb). Stellen Sie vor der Installation sicher, dass Sie keine Snaps installiert haben, die sich mit der Liste der Pakete überschneiden, die Sie als Debian-Pakete verwalten möchten.

`Maximum timeout exceeded`-Fehlermeldung

Aufgrund eines bekannten Problems mit Snap sehen Sie bei `snap`-Befehlen möglicherweise einen `Maximum timeout exceeded`-Fehler. Wenn Sie diese Fehlermeldung erhalten, führen Sie die folgenden Befehle nacheinander aus, um den Agenten zu starten, zu stoppen und seinen Status zu überprüfen:

```
sudo systemctl start snap.amazon-ssm-agent.amazon-ssm-agent.service
```

```
sudo systemctl stop snap.amazon-ssm-agent.amazon-ssm-agent.service
```

```
sudo systemctl status snap.amazon-ssm-agent.amazon-ssm-agent.service
```

Installieren Sie SSM Agent unter Ubuntu Server 16.04 LTS 64-Bit (Snap), 18.04, 20.04, 22.04 LTS, 23.10, 24.04 LTS, 24.0 und 25.04 (mit dem Snap-Paket) wie folgt:

1. SSM Agent wird standardmäßig in Ubuntu Server-AMIs (Versionen 16.04 LTS 64-Bit (Snap), 18.04, 20.04, 22.04 LTS, 23.10, 24.04 LTS, 24.0 und 25.04) mit der Kennung `20180627` oder höher erstellt.

   Sie können das folgende Skript verwenden, wenn Sie SSM Agent auf einem On-Premises-Server installieren oder wenn Sie den Agenten neu installieren möchten. Sie müssen keine URL für den Download angeben, da der `snap`-Befehl den Agenten automatisch aus dem Snap App Store unter https://snapcraft.io herunterlädt.

   ```
   sudo snap install amazon-ssm-agent --classic
   ```

2. Führen Sie den folgenden Befehl aus, um sicherzustellen, dass der SSM Agent ausgeführt wird.

```
sudo snap list amazon-ssm-agent
```

3. Führen Sie den folgenden Befehl aus, um den Service zu starten, wenn der vorherige Befehl die Meldung `amazon-ssm-agent is stopped`, `inactive` oder `disabled` zurückgibt.

```
sudo snap start amazon-ssm-agent
```

4. Prüfen Sie den Status des Agents.

```
sudo snap services amazon-ssm-agent
```

Installieren von SSM Agent unter Ubuntu Server 16.04 64-Bit (deb)

> **Important**
>
> Stellen Sie vor der Installation von SSM Agent auf einer 64-Bit-Version von Ubuntu Server sicher, dass Sie die richtigen Installationstools verwenden. Beginnend mit Amazon Machine Images (AMIs), die mit 20180627 identifiziert wurden, SSM Agent ist in Version 16.04 mit Snap-Paketen vorinstalliert. Auf Instances, die auf einer früheren Version erstellt wurden AMIs, SSM Agent muss es mithilfe von Deb-Installationspaketen installiert werden. Weitere Informationen finden Sie unter Bestimmen der korrekten SSM Agent-Version zur Installation auf 64-Bit-Instances von Ubuntu Server 16.04. Wenn SSM Agent in Verbindung mit einem Snap auf Ihrer Instance installiert ist und Sie SSM Agent mithilfe eines deb-Installationsprogramm-Pakets installieren oder aktualisieren, schlagen die Installation oder SSM Agent-Vorgänge möglicherweise fehl.

In den meisten Fällen ist auf den Amazon Machine Images (AMIs) Ubuntu Server 16.04, die von AWS bereitgestellt werden, AWS Systems Manager Agent (SSM Agent) standardmäßig vorinstalliert. Weitere Informationen finden Sie unter Finden Sie AMIs mit dem vorinstallierten SSM Agent.

Wenn SSM Agent auf einer neuen Ubuntu Server-Instance (Version 16.04) vor Version 20180627 nicht vorinstalliert ist oder wenn Sie den Agenten manuell neu installieren müssen, verwenden Sie die Informationen auf dieser Seite.

Schnellinstallations-Befehle für SSM Agent unter Ubuntu Server 16.04 (deb)

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

Installieren Sie SSM Agent mit Kurzbefehlen zum Kopieren und Einfügen unter Ubuntu Server 16.04 64-Bit (deb) wie folgt:

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer Ubuntu Server-Instance her.
2. Führen Sie den folgenden Befehl aus, um ein temporäres Verzeichnis auf der Instance zu erstellen.

   ```
   mkdir /tmp/ssm
   ```

3. Ändern Sie das temporäre Verzeichnis.

   ```
   cd /tmp/ssm
   ```

4. Führen Sie die folgenden Befehle aus.

   > **Note**
   >
   > Obwohl URLs die folgenden Befehle ein `ec2-downloads-windows` Verzeichnis enthalten, sind dies die richtigen globalen Installationsdateien für Ubuntu Server 16.04 64-Bit.

   ```
   wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/debian_amd64/
   amazon-ssm-agent.deb
   ```

   ```
   sudo dpkg -i amazon-ssm-agent.deb
   ```

5. (Empfohlen) Führen Sie den folgenden Befehl aus, um festzustellen, ob SSM Agent ausgeführt wird.

   Ubuntu Server 16.04

   ```
   sudo systemctl status amazon-ssm-agent
   ```

In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird.

In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

6. Führen Sie den folgenden Befehl aus, um den Service zu starten, wenn der vorherige Befehl die Meldung `amazon-ssm-agent is stopped`, `inactive` oder `disabled` zurückgibt.

   Ubuntu Server 16.04:

   ```
   sudo systemctl enable amazon-ssm-agent
   ```

Erstellen benutzerdefinierter Installationsbefehle für SSM Agent unter Ubuntu Server 16.04 64-Bit (deb) in Ihrer Region

Wenn Sie SSM Agent in mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent unter Ubuntu Server 16.04 (deb) weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie den Befehl im folgenden Befehl *`region`* durch Ihre eigenen Informationen. Eine Liste der unterstützten *`region`* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

```
wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_amd64/amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

Sehen Sie sich das folgende Beispiel an.

```
wget https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/debian_amd64/
amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

Installieren von SSM Agent unter Ubuntu Server 16.04 32-Bit

In den meisten Fällen ist auf den Amazon Machine Images (AMIs) Ubuntu Server 16.04, die von AWS bereitgestellt werden, AWS Systems Manager Agent (SSM Agent) standardmäßig vorinstalliert. Weitere Informationen finden Sie unter Finden Sie AMIs mit dem vorinstallierten SSM Agent.

Wenn SSM Agent auf einer neuen Ubuntu Server-Instance (Version 16.04) nicht vorinstalliert ist oder wenn Sie den Agenten manuell neu installieren müssen, verwenden Sie die Informationen auf dieser Seite.

Schnellinstallations-Befehle für SSM Agent unter Ubuntu Server 16.04 32-Bit (deb)

Gehen Sie wie folgt vor, um SSM Agent manuell auf einer einzelnen Instance zu installieren. Dieses Verfahren verwendet global verfügbare Installationsdateien.

Installieren Sie SSM Agent mit Kurzbefehlen zum Kopieren und Einfügen unter Ubuntu Server 16.04 32-Bit (deb) wie folgt:

1. Stellen Sie mithilfe Ihrer bevorzugten Methode, z. B. SSH, eine Verbindung mit Ihrer Ubuntu Server-Instance her.
2. Führen Sie den folgenden Befehl aus, um ein temporäres Verzeichnis auf der Instance zu erstellen.

   ```
   mkdir /tmp/ssm
   ```

3. Ändern Sie das temporäre Verzeichnis.

   ```
   cd /tmp/ssm
   ```

4. Führen Sie die folgenden Befehle aus.

> **Note**
>
> Obwohl URLs in den folgenden Befehlen ein `ec2-downloads-windows`-Verzeichnis enthalten, ist dies die richtige globale Installationsdatei für Ubuntu Server 16.04 32-Bit.

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/debian_386/
amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

5. (Empfohlen) Führen Sie den folgenden Befehl aus, um festzustellen, ob SSM Agent ausgeführt wird.

   Ubuntu Server 16.04

   ```
   sudo systemctl status amazon-ssm-agent
   ```

   In den meisten Fällen meldet der Befehl, dass der Agent ausgeführt wird.

   In seltenen Fällen meldet der Befehl, dass der Agent installiert ist, aber nicht ausgeführt wird, wie im folgenden Beispiel gezeigt.

6. Führen Sie den folgenden Befehl aus, um den Service zu starten, wenn der vorherige Befehl die Meldung `amazon-ssm-agent is stopped`, `inactive` oder `disabled` zurückgibt.

   Ubuntu Server 16.04:

   ```
   sudo systemctl enable amazon-ssm-agent
   ```

Erstellen benutzerdefinierter Installationsbefehle für SSM Agent unter Ubuntu Server 16.04 32-Bit (deb) in Ihrer Region

Wenn Sie SSM Agent in mehreren Instances installieren, die ein Skript oder eine Vorlage verwenden, empfehlen wir die Verwendung von Installationsdateien, die in der AWS-Region gespeichert wurden, in der Sie arbeiten.

Für die folgenden Befehle stellen wir Beispiele bereit, die einen öffentlich zugänglichen S3-Bucket in der Region USA Ost (Ohio) (`us-east-2`) verwenden.

> **Tip**
>
> Sie können auch eine globale URL im Verfahren Schnellinstallations-Befehle für SSM Agent unter Ubuntu Server 16.04 32-Bit (deb) weiter oben in diesem Thema durch eine benutzerdefinierte regionale URL ersetzen, die Sie erstellen.

Ersetzen Sie den Befehl im folgenden Befehl durch Ihre *region* eigenen Informationen. Eine Liste der unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

```
wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/debian_386/amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

Sehen Sie sich das folgende Beispiel an.

```
wget https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/debian_386/amazon-ssm-agent.deb
```

```
sudo dpkg -i amazon-ssm-agent.deb
```

Bestimmen der korrekten SSM Agent-Version zur Installation auf 64-Bit-Instances von Ubuntu Server 16.04

> **Important**
>
> Stellen Sie vor der Installation von SSM Agent auf einer 64-Bit-Version von Ubuntu Server sicher, dass Sie die richtigen Installationstools verwenden. Beginnend mit Amazon Machine Images (AMIs), die mit 20180627 identifiziert werden, ist SSM Agent unter Verwendung von Snap-Paketen auf Version 16.04 vorinstalliert. Auf Instances, die aus früheren AMIs erstellt wurden, muss SSM Agent mit Deb-Installationsprogramm-Paketen installiert werden. Weitere Informationen finden Sie unter Bestimmen der korrekten SSM Agent-Version zur Installation auf 64-Bit-Instances von Ubuntu Server 16.04

Beachten Sie, dass bei Instances mit mehr als einer Installation des SSM Agent (z. B. eine mit einem Snap und eine mit einem deb-Installationsprogramm installierte Instance) Ihre Agenten-Operationen nicht ordnungsgemäß ausgeführt werden.

Sie können das AMI-ID-Quell-Erstellungsdatum für eine Instance mit einem der folgenden Verfahren verifizieren. Diese Verfahren gelten nur für AWS-verwaltete AMIs.

Überprüfen eines AMI-ID-Quell-Erstellungsdatums (Konsole)

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im linken Navigationsbereich die Option Instances aus.
3. Wählen Sie eine Instance aus.
4. Suchen Sie auf der Registerkarte Details nach einer YYYYMMDD-Kennung in dem Wert im Feld AMI-Name. Beispiel: `ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20180627`.

Überprüfen eines AMI-ID-Quell-Erstellungsdatums (AWS CLI)

- Führen Sie den folgenden Befehl aus.

```
aws ec2 describe-images --image-ids ami-id
```

*ami-id*repräsentiert die ID eines von AWS bereitgestellten AMI, wie beispielsweise `ami-07c8bc5c1ce9598c3`.

Bei Erfolg gibt der Befehl Informationen wie die folgenden zurück, in denen Sie die `CreationDate`- und `Name`-Felder auf Informationen prüfen können.

```
{
    "Images": [
        {
            "Architecture": "x86_64",
            "CreationDate": "2020-07-24T20:40:27.000Z",
            "ImageId": "ami-07c8bc5c1ce9598c3",
-- truncated --
            "ImageOwnerAlias": "amazon",
            "Name": "amzn2-ami-hvm-2.0.20200722.0-x86_64-gp2",
            "RootDeviceName": "/dev/xvda",
```

```
            "RootDeviceType": "ebs",
            "SriovNetSupport": "simple",
            "VirtualizationType": "hvm"
        }
    ]
}
```

Konfigurieren Sie SSM Agent, um einen Proxy in Linux-Knoten zu verwenden

Sie können AWS Systems Manager Agent (SSM Agent) für die Kommunikation über einen HTTP-Proxy konfigurieren, indem Sie eine Override-Konfigurationsdatei erstellen und der Datei `http_proxyhttps_proxy`,, und `no_proxy` Einstellungen hinzufügen. Eine Override-Datei behält auch die Proxy-Einstellungen bei, wenn Sie neuere oder ältere Versionen von SSM Agent installieren. Dieser Abschnitt enthält Vorgänge zum Erstellen einer Override-Datei für die Umgebungen upstart und systemd. Wenn Sie Session Manager verwenden möchten, beachten Sie, dass HTTPS-Proxyserver nicht unterstützt werden.

Themen

- Konfigurieren des SSM Agent zur Nutzung eines Proxys (Upstart)
- Konfigurieren des SSM Agent zur Nutzung eines Proxys (systemd)

Konfigurieren des SSM Agent zur Nutzung eines Proxys (Upstart)

Gehen Sie folgendermaßen vor, um eine Override-Konfigurationsdatei für eine `upstart`-Umgebung zu erstellen.

Konfigurieren des SSM Agent zur Nutzung eines Proxys (Upstart)

1. Stellen Sie eine Verbindung mit der verwalteten Instance her, auf der Sie den SSM Agent installiert haben.
2. Öffnen Sie einen einfachen Editor wie VIM und geben Sie je nachdem, ob Sie einen HTTP-Proxy-Server oder HTTPS-Proxy-Server verwenden, eine der folgenden Konfigurationen an:

   Für einen HTTP-Proxy-Server:

   ```
   env http_proxy=http://hostname:port
   ```

```
env https_proxy=http://hostname:port
env no_proxy=IP address for instance metadata services (IMDS)
```

Bei einem HTTPS-Proxy-Server:

```
env http_proxy=http://hostname:port
env https_proxy=https://hostname:port
env no_proxy=IP address for instance metadata services (IMDS)
```

> **Important**
>
> Fügen Sie die `no_proxy`-Einstellung der Datei hinzu und geben Sie die IP-Adresse an. Die IP-Adresse für `no_proxy` ist der IMDS-Endpunkt (Instance Metadata Services) für Systems Manager. Wenn Sie nichts angeben`no_proxy`, übernehmen Aufrufe von Systems Manager die Identität des Proxydienstes (falls IMDSv1 Fallback aktiviert ist), oder Aufrufe von Systems Manager schlagen fehl (falls IMDSv2 erzwungen).
>
> - Geben Sie für IPv4 an. `no_proxy=169.254.169.254`
> - Für IPv6, geben Sie an`no_proxy=[fd00:ec2::254]`. Die IPv6 Adresse des Instanz-Metadatendienstes ist mit IMDSv2 Befehlen kompatibel. Auf die IPv6 Adresse kann nur auf Instanzen zugegriffen werden, die auf dem AWS Nitro-System basieren. Weitere Informationen finden Sie unter So funktioniert Instance Metadata Service Version 2 im EC2 Amazon-Benutzerhandbuch.

3. Speichern Sie die Datei unter dem Namen `amazon-ssm-agent.override` am folgenden Speicherort: `/etc/init/`
4. Mit den folgenden Befehlen wird der SSM Agent beendet und neu gestartet.

```
sudo service stop amazon-ssm-agent
sudo service start amazon-ssm-agent
```

> **Note**
>
> Weitere Informationen zum Arbeiten mit `.override`-Dateien in Upstart-Umgebungen finden Sie unter init: Upstart-init-Daemon-Auftragskonfiguration.

Konfigurieren des SSM Agent zur Nutzung eines Proxys (systemd)

Gehen Sie wie folgt vor, um SSM Agent für die Verwendung eines Proxy in einer `systemd`-Umgebung zu konfigurieren.

> **Note**
>
> Einige der Schritte in diesem Verfahren enthalten explizite Anweisungen für Ubuntu Server-Instances, wenn SSM Agent unter Verwendung von Snap installiert wurden.

1. Stellen Sie eine Verbindung mit der Instance her, auf der Sie den SSM Agent installiert haben.
2. Führen Sie abhängig von der Art des Betriebssystems einen der folgenden Befehle aus.

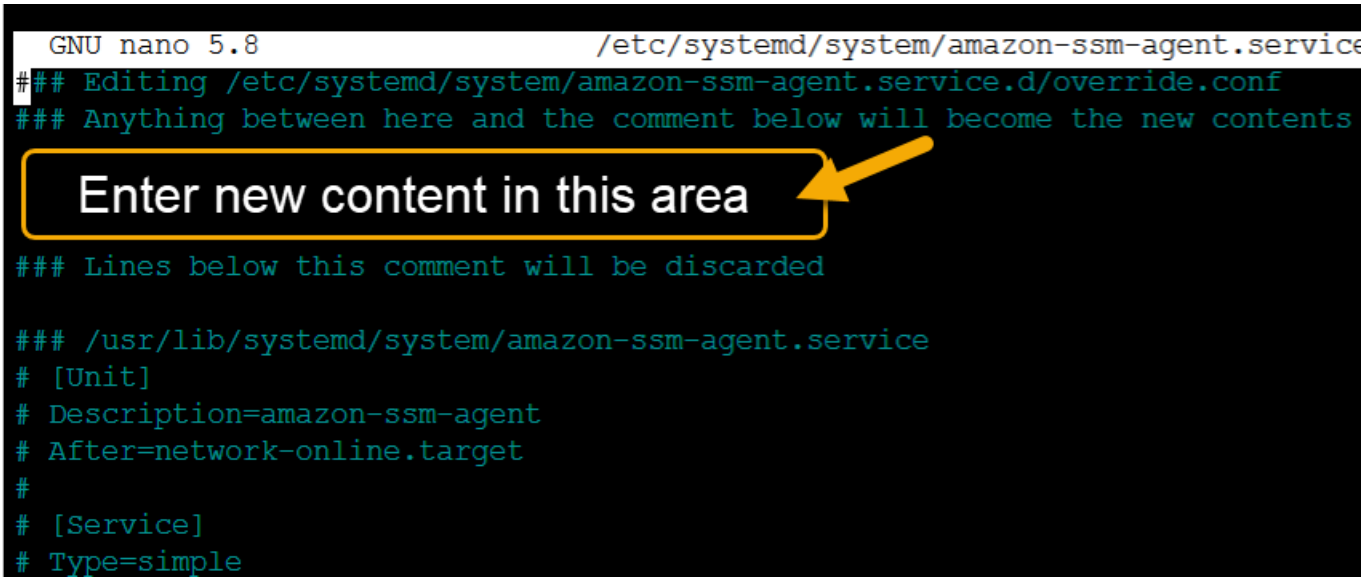
   - Auf Ubuntu Server-Instances, wenn SSM Agent unter Verwendung von Snap installiert wurde:

     ```
     sudo systemctl edit snap.amazon-ssm-agent.amazon-ssm-agent
     ```

     Auf anderen Betriebssystemen:

     ```
     sudo systemctl edit amazon-ssm-agent
     ```

3. Öffnen Sie einen einfachen Editor wie VIM und geben Sie je nachdem, ob Sie einen HTTP-Proxy-Server oder HTTPS-Proxy-Server verwenden, eine der folgenden Konfigurationen an:



   Für einen HTTP-Proxy-Server:

```
[Service]
Environment="http_proxy=http://hostname:port"
Environment="https_proxy=http://hostname:port"
Environment="no_proxy=IP address for instance metadata services (IMDS)"
```

Bei einem HTTPS-Proxy-Server:

```
[Service]
Environment="http_proxy=http://hostname:port"
Environment="https_proxy=https://hostname:port"
Environment="no_proxy=IP address for instance metadata services (IMDS)"
```

> **Important**
>
> Fügen Sie die `no_proxy`-Einstellung der Datei hinzu und geben Sie die IP-Adresse an. Die IP-Adresse für `no_proxy` ist der IMDS-Endpunkt (Instance Metadata Services) für Systems Manager. Wenn Sie nichts angeben`no_proxy`, übernehmen Aufrufe von Systems Manager die Identität des Proxydienstes (falls IMDSv1 Fallback aktiviert ist), oder Aufrufe von Systems Manager schlagen fehl (falls IMDSv2 erzwungen).
>
> - Geben Sie für IPv4 an. `no_proxy=169.254.169.254`
> - Für IPv6, geben Sie an`no_proxy=[fd00:ec2::254]`. Die IPv6 Adresse des Instanz-Metadatendienstes ist mit IMDSv2 Befehlen kompatibel. Auf die IPv6 Adresse kann nur auf Instanzen zugegriffen werden, die auf dem AWS Nitro-System basieren. Weitere Informationen finden Sie unter So funktioniert Instance Metadata Service Version 2 im EC2 Amazon-Benutzerhandbuch.

4. Speichern Sie Ihre Änderungen. Abhängig vom Typ des Betriebssystems erstellt das System automatisch eine der folgenden Dateien.

   - Auf Ubuntu Server-Instances, wenn SSM Agent unter Verwendung von Snap installiert wurde:

     `/etc/systemd/system/snap.amazon-ssm-agent.amazon-ssm-agent.service.d/override.conf`

   - Auf Instances von Amazon Linux 2, Amazon Linux 2023 und RHEL:

     `/etc/systemd/system/amazon-ssm-agent.service.d/override.conf`

- Auf anderen Betriebssystemen

  `/etc/systemd/system/amazon-ssm-agent.service.d/`**`amazon-ssm-agent.override`**

5. Starten Sie SSM Agent abhängig von der Art des Betriebssystems über einen der folgenden Befehle neu.

   - Auf Ubuntu Server-Instances, die unter Verwendung von Snap installiert wurden:

     ```
     sudo systemctl daemon-reload && sudo systemctl restart snap.amazon-ssm-agent.amazon-ssm-agent
     ```

   - Auf anderen Betriebssystemen:

     ```
     sudo systemctl daemon-reload && sudo systemctl restart amazon-ssm-agent
     ```

> **Note**
>
> Weitere Informationen zum Arbeiten mit `.override`-Dateien in systemd-Umgebungen finden Sie unter Modifying Existing Unit Files im Systemadministrator-Handbuch für Red Hat Enterprise Linux 7.

Arbeiten mit SSM Agent EC2 On-Instances für macOS

AWS Systems Manager (SSM Agent) verarbeitet Systems Manager Manager-Anfragen und konfiguriert Ihren Computer wie in der Anfrage angegeben. Sie können den SSM Agent für macOS auf folgende Weise installieren, konfigurieren oder deinstallieren.

> **Note**
>
> SSM Agent ist standardmäßig auf Amazon Machine Images (AMIs) für macOS vorinstalliert. Sie müssen nicht SSM Agent auf einer Amazon Elastic Compute Cloud (Amazon EC2) - Instance installieren, macOS sofern Sie sie nicht deinstalliert haben.

Der Quellcode für SSM Agent ist in GitHub verfügbar, damit Sie den Agenten entsprechend Ihren Anforderungen anpassen können. Wir möchten Sie bitten, uns eventuelle Änderungswünsche mitzuteilen. AWS bietet jedoch keine Unterstützung für die Ausführung modifizierter Kopien dieser Software.

> **Note**
>
> Details zu den unterschiedlichen SSM Agent-Versionen finden Sie in den Versionshinweisen.

Bevor Sie SSM Agent manuell auf einem macOS-Betriebssystem installieren, lesen Sie die folgenden Informationen.

- SSM Agentist standardmäßig auf den EC2 Instanzen installiert und Amazon Machine Images wird von Systems Manager unterstützt. Weitere Informationen finden Sie auf der Liste unterstützter Betriebssysteme für macOS.

  SSM Agentmuss nicht manuell auf macOS EC2 Instanzen installiert werden, es sei denn, es wurde deinstalliert.
- EC2 Instanzen für macOS werden nicht in allen unterstützt AWS-Regionen. Eine Liste der Regionen, in denen x86-basierte und EC2 M1-Instances unterstützt macOS werden, finden Sie unter macOSWorkloads im Amazon. EC2 FAQs
- Wenn Systems Manager neue Tools hinzugefügt oder Aktualisierungen an den vorhandenen Tools vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Tools und Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent in Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die SSM Agent-Versionshinweise-Seite in GitHub, um Benachrichtigungen über SSM Agent-Updates zu erhalten.

Themen

- Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für macOS

Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für macOS

Stellen Sie eine Verbindung mit Ihrer macOS-Instance her und folgen Sie den Schritten, um den AWS Systems Manager -Agent (SSM Agent) zu installieren. Führen Sie diese Schritte mit Systems Manager auf jeder Instance aus, auf der Befehle ausgeführt werden. Die in diesem Verfahren bereitgestellten Befehle können auch als Skripte über Benutzerdaten an EC2 Amazon-Instances übergeben werden.

> **Important**
>
> Wir empfehlen dringend, die Verwendung von Betriebssystemversionen zu vermeiden, die End-of-Life (EOL) erreicht haben. Betriebssystemanbieter, darunter auch, bieten AWS in der Regel keine Sicherheitspatches oder andere Updates für Versionen an, die EOL erreicht haben. Die weitere Verwendung eines EOL-Systems erhöht das Risiko, dass Upgrades, einschließlich Sicherheitsupdates, und andere Betriebsprobleme nicht durchgeführt werden können, erheblich. AWS testet die Systems Manager Manager-Funktionalität nicht auf Betriebssystemversionen, die EOL erreicht haben.

Bevor Sie beginnen

Installieren Sie `wget` mithilfe von Homebrew.

Installieren Sie den SSM Agent auf macOS wie folgt

1. Laden Sie die Agent-Installationsdatei für x86_64-Instances mit dem folgenden Befehl herunter.

   Ersetzen Sie den Befehl im folgenden Befehl *`region`* durch Ihre eigenen Informationen. Eine Liste der unterstützten *`region`* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

   ```
   sudo wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/darwin_amd64/
   amazon-ssm-agent.pkg
   ```

   Verwenden Sie für Apple silicon-Instances den folgenden Befehl.

```
sudo wget https://s3.region.amazonaws.com/amazon-ssm-region/latest/darwin_arm64/
amazon-ssm-agent.pkg
```

Ein Beispiel.

```
sudo wget https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/latest/
darwin_amd64/amazon-ssm-agent.pkg
```

2. Verwenden Sie den folgenden Befehl, um das SSM Agent-Installationsprogramm auszuführen.

   x86_64:

   ```
   sudo installer -pkg amazon-ssm-agent.pkg -target /
   ```

3. Prüfen Sie den Status des Agents.

   Um zu überprüfen, ob der SSM Agent ausgeführt wird, prüfen Sie das Agent-Protokoll unter `/var/log/amazon/ssm/amazon-ssm-agent.log`.

4. Führen Sie den folgenden Befehl aus, um den Dienst zu starten, falls aus dem Agent-Protokoll hervorgeht, dass "gestoppt amazon-ssm-agent ist“.

   ```
   sudo launchctl load -w /Library/LaunchDaemons/com.amazon.aws.ssm.plist && sudo
    launchctl start com.amazon.aws.ssm
   ```

> **Important**
>
> Wenn Systems Manager neue Tools hinzugefügt oder Aktualisierungen an den vorhandenen Tools vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Tools und Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent in Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die SSM Agent-Versionshinweise-Seite in GitHub, um Benachrichtigungen über SSM Agent-Updates zu erhalten.

Deinstallieren des SSM Agent von macOS-Instances

macOS unterstützt die Deinstallation von PKG-Dateien nicht nativ. Um AWS Systems Manager-Agent (SSM Agent) aus einer Amazon Elastic Compute Cloud (Amazon EC2)-Instance für macOS zu deinstallieren, können Sie das AWS-verwaltete Skript aus dem folgenden Speicherort verwenden.

https://github.com/aws/amazon-ssm-agent/blob/mainline/Tools/src/update/darwin/uninstall.sh

Arbeiten mit SSM Agent EC2 On-Instances für Windows Server

AWS Systems Manager Agent (SSM Agent) ist standardmäßig auf den Amazon Machine Images (AMIs) vorinstalliert, die von AWS bereitgestellt werden. Windows Server Support wird für die folgenden Betriebssystemversionen bereitgestellt.

- Windows Server 2012 R2-AMIs, die im November 2016 oder später veröffentlicht wurden
- Windows Server 2016, 2019, 2022 (ohne Nano-Versionen) und 2025

Support-Hinweise für frühere Versionen

Windows ServerAMIsvor November 2016 veröffentlichte Versionen verwenden den EC2 Config-Service, um Anfragen zu verarbeiten und Instanzen zu konfigurieren.

Sofern Sie keinen bestimmten Grund dafür haben, den EC2 Config-Service oder eine frühere Version von zur Verarbeitung von SSM Agent Systems Manager-Anfragen zu verwenden, empfehlen wir Ihnen, die neueste Version von SSM Agent herunterzuladen und auf jeder Ihrer Amazon Elastic Compute Cloud (Amazon EC2) -Instances oder EC2 Nicht-Maschinen, die für Systems Manager in einer Hybrid- und Multi-Cloud-Umgebung konfiguriert sind, zu installieren.

Aktualisieren von SSM Agent

Wenn Systems Manager neue Tools hinzugefügt oder Aktualisierungen an den vorhandenen Tools vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Tools und Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent in Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die SSM Agent-Versionshinweise-Seite in GitHub, um Benachrichtigungen über SSM Agent-Updates zu erhalten.

Details zu den unterschiedlichen SSM Agent-Versionen finden Sie in den Versionshinweisen.

Themen

- Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Windows Server
- Konfigurieren des SSM Agent zur Nutzung eines Proxys für Windows Server-Instances

Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Windows Server

AWS Systems Manager Agent (SSM Agent) ist standardmäßig auf dem folgenden Amazon Machine Images (AMIs) von Amazon Windows Server bereitgestellten Gerät vorinstalliert:

- Windows Server 2012 R2-AMIs, die im November 2016 oder später veröffentlicht wurden
- Windows Server 2016, 2019, 2022 (ohne Nano-Versionen) und 2025

> **Important**
>
> Wir empfehlen dringend, die Verwendung von Betriebssystemversionen zu vermeiden, die End-of-Life (EOL) erreicht haben. Betriebssystemanbieter, darunter auch, bieten AWS in der Regel keine Sicherheitspatches oder andere Updates für Versionen an, die EOL erreicht haben. Die weitere Verwendung eines EOL-Systems erhöht das Risiko, dass Upgrades, einschließlich Sicherheitsupdates, und andere Betriebsprobleme nicht durchgeführt werden können, erheblich. AWS testet die Systems Manager Manager-Funktionalität nicht auf Betriebssystemversionen, die EOL erreicht haben.

Installieren Sie SSM Agent auf EC2 Instanzen für Windows Server

Falls erforderlich, können Sie die neueste Version von manuell herunterladen und SSM Agent auf Ihrer Amazon Elastic Compute Cloud (Amazon EC2) -Instance installieren, Windows Server indem Sie das folgende Verfahren verwenden. Die in diesem Verfahren bereitgestellten Befehle können auch als Skripte über Benutzerdaten an EC2 Amazon-Instances übergeben werden.

SSM Agenterfordert Windows PowerShell 3.0 oder höher, um bestimmte AWS Systems Manager Dokumente (SSM-Dokumente) auf Windows Server Instanzen auszuführen (z. B. das ältere `AWS-ApplyPatchBaseline` Dokument). Vergewissern Sie sich, dass Ihre Windows Server-Instances auf Windows Management Framework 3.0 oder höher ausgeführt werden. Dieses Framework umfasst Windows PowerShell. Weitere Informationen finden Sie unter Windows Management Framework 3.0

> **Installation auf anderen Maschinentypen**
>
> Dieses Verfahren in diesem Thema bezieht sich speziell auf die Installation oder Neuinstallation SSM Agent auf einer EC2 Instanz von. Windows Server Verwenden Sie das `ssm-setup-cli` Tool für lokale Server, virtuelle Maschinen oder andere EC2 Umgebungen, wie unter beschrieben. the section called “Installation von SSM Agent auf hybriden Windows Server-Knoten”
>
> EC2 Die Verwendung von Installationsverfahren auf EC2 Nichtsystemen kann potenziell zu Sicherheitslücken führen. Das `ssm-setup-cli` Tool bietet zusätzliche Sicherheitsvorkehrungen für Computer, die keine EC2 Computer sind.

Um die neueste Version von EC2 On-Instances manuell SSM Agent zu installieren für Windows Server

1. Stellen Sie mithilfe von Remote Desktop oder Windows eine Connect zu Ihrer Instance her PowerShell. Weitere Informationen finden Sie unter Connect zu Ihrer Instance herstellen im EC2 Amazon-Benutzerhandbuch.
2. Laden Sie die neueste Version des SSM Agent auf Ihre Instance herunter. Sie können entweder PowerShell Befehle oder einen direkten Download-Link verwenden.

   > **Note**
   >
   > URLs In diesem Schritt können Sie SSM Agent von jedem herunterladen AWS-Region. Wenn Sie den Agenten aus einer bestimmten Region herunterladen möchten, verwenden Sie stattdessen eine regionsspezifische URL: `https://amazon-ssm-region.s3.region.amazonaws.com/latest/windows_amd64/AmazonSSMAgentSetup.exe`
   >
   > *region*steht für den Bezeichner einer Region AWS Systems Manager, die von AWS-Region unterstützt wird, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

PowerShell

Führen Sie die folgenden drei PowerShell Befehle der Reihe nach aus. Mit diesen Befehlen können Sie SSM Agent herunterladen, ohne die erweiterten Sicherheitseinstellungen von Internet Explorer (IE) anzupassen, und dann den Agent installieren und die Installationsdatei entfernen.

64-bit

```
[System.Net.ServicePointManager]::SecurityProtocol = 'TLS12'
$progressPreference = 'silentlyContinue'
Invoke-WebRequest `
    https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/
windows_amd64/AmazonSSMAgentSetup.exe `
    -OutFile $env:USERPROFILE\Desktop\SSMAgent_latest.exe
```

32-bit

```
[System.Net.ServicePointManager]::SecurityProtocol = 'TLS12'
$progressPreference = 'silentlyContinue'
Invoke-WebRequest `
    https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/
windows_386/AmazonSSMAgentSetup.exe `
    -OutFile $env:USERPROFILE\Desktop\SSMAgent_latest.exe
```

```
Start-Process `
    -FilePath $env:USERPROFILE\Desktop\SSMAgent_latest.exe `
    -ArgumentList "/S" `
    -Wait
```

```
rm -Force $env:USERPROFILE\Desktop\SSMAgent_latest.exe
```

Direkter Download

Laden Sie über den folgenden Link die neueste Version von SSM Agent auf Ihre Instance herunter. Wenn Sie möchten, aktualisieren Sie diese URL mit einer AWS-Region-spezifischen URL.

https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/windows_amd64/AmazonSSMAgentSetup.exe

Führen Sie die heruntergeladene `AmazonSSMAgentSetup.exe`-Datei aus, um SSM Agent zu installieren.

3. Starten oder starten Sie neu, SSM Agent indem Sie den folgenden Befehl einsenden PowerShell:

```
Restart-Service AmazonSSMAgent
```

> **Note**
>
> Um SSM Agent von einer Windows Server-Instance zu deinstallieren, öffnen Sie die Systemsteuerung, Programme. Wählen Sie die Option Uninstall a program (Programm deinstallieren). Öffnen Sie das Kontextmenü (Rechtsklick) für Amazon SSM Agent und wählen Sie Uninstall (Deinstallieren) aus.

Konfigurieren des SSM Agent zur Nutzung eines Proxys für Windows Server-Instances

Die Informationen in diesem Thema gelten für Windows Server-Instances, die im oder nach dem November 2016 erstellt wurden, die nicht die Nano-Installationsoption verwenden. Wenn Sie Session Manager verwenden möchten, beachten Sie, dass HTTPS-Proxyserver nicht unterstützt werden.

Bevor Sie beginnen

Bevor Sie SSM Agent für die Verwendung eines Proxys konfigurieren, beachten Sie die folgenden wichtigen Informationen.

Im folgenden Verfahren führen Sie einen Befehl aus, um die Verwendung eines Proxys SSM Agent zu konfigurieren. Der Befehl enthält eine `no_proxy`-Einstellung mit einer IP-Adresse. Die IP-Adresse ist der IMDS-Endpunkt (Instance Metadata Services) für Systems Manager. Wenn Sie nichts angeben`no_proxy`, übernehmen Aufrufe von Systems Manager die Identität des Proxydienstes (falls IMDSv1 Fallback aktiviert ist), oder Aufrufe von Systems Manager schlagen fehl (falls IMDSv2 erzwungen).

- Geben Sie für IPv4 an. `no_proxy=169.254.169.254`
- Für IPv6, geben Sie an`no_proxy=[fd00:ec2::254]`. Die IPv6 Adresse des Instanz-Metadatendienstes ist mit IMDSv2 Befehlen kompatibel. Auf die IPv6 Adresse kann nur auf Instanzen zugegriffen werden, die auf dem AWS Nitro-System basieren. Weitere Informationen finden Sie unter So funktioniert Instance Metadata Service Version 2 im EC2 Amazon-Benutzerhandbuch.

So konfigurieren Sie den SSM Agent zur Nutzung eines Proxys

1. Stellen Sie über Remote Desktop oder Windows PowerShell eine Verbindung zu der Instance her, die Sie für die Verwendung eines Proxys konfigurieren möchten.
2. Führen Sie den folgenden Befehlsblock in aus PowerShell. Ersetzen Sie *`hostname`* und *`port`* durch die Informationen zu Ihrem Proxy.

```
$serviceKey = "HKLM:\SYSTEM\CurrentControlSet\Services\AmazonSSMAgent"
$keyInfo = (Get-Item -Path $serviceKey).GetValue("Environment")
$proxyVariables = @("http_proxy=hostname:port", "https_proxy=hostname:port",
 "no_proxy=IP address for instance metadata services (IMDS)")

if ($keyInfo -eq $null) {
    New-ItemProperty -Path $serviceKey -Name Environment -Value $proxyVariables -
PropertyType MultiString -Force
} else {
    Set-ItemProperty -Path $serviceKey -Name Environment -Value $proxyVariables
}

Restart-Service AmazonSSMAgent
```

Nachdem Sie den vorherigen Befehl ausgeführt haben, können Sie die SSM Agent-Protokolle überprüfen, um zu bestätigen, dass die Proxy-Einstellungen angewendet wurden. Einträge in den Protokollen ähneln den folgenden. Weitere Informationen über SSM Agent-Protokolle finden Sie unter Anzeigen von SSM Agent-Protokollen.

```
2020-02-24 15:31:54 INFO Getting IE proxy configuration for current user: The operation
 completed successfully.
2020-02-24 15:31:54 INFO Getting WinHTTP proxy default configuration: The operation
 completed successfully.
2020-02-24 15:31:54 INFO Proxy environment variables:
2020-02-24 15:31:54 INFO http_proxy: hostname:port
```

```
2020-02-24 15:31:54 INFO https_proxy: hostname:port
2020-02-24 15:31:54 INFO no_proxy: IP address for instance metadata services (IMDS)
2020-02-24 15:31:54 INFO Starting Agent: amazon-ssm-agent - v2.3.871.0
2020-02-24 15:31:54 INFO OS: windows, Arch: amd64
```

So setzen die Proxy-Konfiguration des SSM Agent zurück

1. Stellen Sie mithilfe von Remote Desktop oder Windows eine Verbindung zu der Instanz her PowerShell, um sie zu konfigurieren.
2. Wenn Sie über Remote Desktop eine Verbindung hergestellt haben, starten Sie PowerShell als Administrator.
3. Führen Sie den folgenden Befehlsblock in aus PowerShell.

   ```
   Remove-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\AmazonSSMAgent -
   Name Environment
   Restart-Service AmazonSSMAgent
   ```

Priorität der SSM Agent-Proxy-Einstellung

Beim Konfigurieren von Proxyeinstellungen für die SSM Agent unter Windows Server-Instances muss beachtet werden, dass diese Einstellungen ausgewertet und auf die Agentenkonfiguration angewendet werden, wenn der SSM Agent gestartet wird. Wie Sie Ihre Proxy-Einstellungen für eine Windows Server-Instance konfigurieren, kann bestimmen, ob andere Einstellungen Ihre beabsichtigten Einstellungen möglicherweise ersetzen. Der Agent verwendet die ersten Proxyeinstellungen, die er findet.

> **Important**
>
> SSM Agent kommuniziert mit dem HTTPS-Protokoll. Aus diesem Grund müssen Sie die `HTTPS proxy`-Parameter mithilfe einer der folgenden Einstellungsoptionen konfigurieren.

SSM Agent-Proxy-Einstellungen werden in der folgenden Reihenfolge ausgewertet.

1. `AmazonSSMAgent`-Registrierungseinstellungen (`HKLM:\SYSTEM\CurrentControlSet\Services\AmazonSSMAgent`)
2. `System`-Umgebungsvariablen (`http_proxy`, `https_proxy`, `no_proxy`)

3. Umgebungsvariablen des `LocalSystem`-Benutzerkontos `http_proxy`, `https_proxy`, `no_proxy`)
4. Browsereinstellungen (`HTTP`, `secure`, `exceptions`)
5. `WinHTTP`-Proxy-Einstellungen (`http=`, `https=`, `bypass-list=`)

SSM Agent-Proxy-Einstellungen und Systems Manager-Services

Wenn Sie den für SSM Agent die Verwendung eines Proxys konfiguriert haben und AWS Systems Manager Tools wie Run Command und verwenden Patch Manager PowerShell , die während der Ausführung auf Windows Server Instanzen den Windows Update-Client verwenden, konfigurieren Sie zusätzliche Proxyeinstellungen. Andernfalls schlägt der Vorgang möglicherweise fehl, da die vom PowerShell und vom Windows Update-Client verwendeten Proxyeinstellungen nicht von der SSM Agent Proxykonfiguration übernommen werden.

Konfigurieren Sie für Run Command `WinINet`-Proxy-Einstellungen auf Ihren Windows Server-Instances. Die `[System.Net.WebRequest]`-Befehle werden pro Sitzung bereitgestellt.
Um diese Konfigurationen auf nachfolgende Netzwerkbefehle anzuwenden, die in ausgeführt werdenRun Command, müssen diese Befehle vor anderen PowerShell Befehlen in derselben `aws:runPowershellScript` Plugin-Eingabe stehen.

Die folgenden PowerShell Befehle geben die aktuellen `WinINet` Proxyeinstellungen zurück und wenden Ihre Proxyeinstellungen auf an`WinINet`.

```
[System.Net.WebRequest]::DefaultWebProxy

$proxyServer = "http://hostname:port"
$proxyBypass = "169.254.169.254"
$WebProxy = New-Object System.Net.WebProxy($proxyServer,$true,$proxyBypass)

[System.Net.WebRequest]::DefaultWebProxy = $WebProxy
```

Sie müssen für Patch Manager systemweite Proxy-Einstellungen konfigurieren, damit der Windows Update-Client nach Updates suchen und herunterladen kann. Es wird empfohlen, dass Sie Run Command verwenden, um die folgenden Befehle auszuführen, da sie auf dem SYSTEM-Konto ausgeführt werden und die Einstellungen systemweit gelten. Mit den folgenden `netsh`-Befehlen werden die aktuellen Proxy-Einstellungen zurückgegeben und die Proxy-Einstellungen werden auf das lokale System angewendet.

```
netsh winhttp show proxy

netsh winhttp set proxy proxy-server="hostname:port" bypass-list="169.254.169.254"
```

Weitere Informationen zur Verwendung von Run Command finden Sie unter AWS Systems Manager Run Command.

Prüfen des SSM Agent-Status und Starten des Agenten

In diesem Thema werden die Befehle aufgeführt, mit denen überprüft werden kann, ob AWS Systems Manager Agent (SSM Agent) auf jedem unterstützten Betriebssystem ausgeführt wird. Es enthält auch die Befehle, mit denen der Agent gestartet wird, wenn er nicht ausgeführt wird.

Betriebssystem	Befehl zum Überprüfen des SSM Agent-Status	Befehl zum Starten von SSM Agent
Amazon Linux 2 und Amazon Linux 2023	`sudo systemctl status amazon-ssm-agent`	`sudo systemctl enable amazon-ssm-agent` `sudo systemctl start amazon-ssm-agent`
Debian Server11 und 12	`sudo systemctl status amazon-ssm-agent`	`sudo systemctl enable amazon-ssm-agent` `sudo systemctl start amazon-ssm-agent`
macOS	Überprüfen Sie die Agent-Protokolldatei unter `/var/log/amazon/ssm/amazon-ssm-agent.log`	`sudo launchctl load -w /Library/LaunchDaemons/com.amazon.aws.ssm.plist` `sudo launchctl start com.amazon.aws.ssm`
Oracle Linux	`sudo systemctl status amazon-ssm-agent`	`sudo systemctl enable amazon-ssm-agent`

Betriebssystem	Befehl zum Überprüfen des SSM Agent-Status	Befehl zum Starten von SSM Agent
		`sudo systemctl start amazon-ssm-agent`
Red Hat Enterprise Linux (RHEL) 7.x, 8.x, 9.x, und 10.x	`sudo systemctl status amazon-ssm-agent`	`sudo systemctl enable amazon-ssm-agent` `sudo systemctl start amazon-ssm-agent`
SUSE Linux Enterprise Server (SLES)	`sudo systemctl status amazon-ssm-agent`	`sudo systemctl enable amazon-ssm-agent` `sudo systemctl start amazon-ssm-agent`
Ubuntu Server 16.04 (32 Bit)	`sudo status amazon-ssm-agent`	`sudo start amazon-ssm-agent`
Ubuntu Server 16.04-64-Bit-Instances (deb-Installations programmpaket)	`sudo systemctl status amazon-ssm-agent`	`sudo systemctl enable amazon-ssm-agent` `sudo systemctl start amazon-ssm-agent`
Ubuntu Server 16.04 LTS, 18.04, 20.04, 22.04 LTS, 23.10, 24.04 LTS, 24.0 und 25.04	`sudo systemctl status snap.amazon-ssm-agent.amazon-ssm-agent.service`	`sudo snap start amazon-ssm-agent`
Windows Server	Führen Sie es aus in PowerShell: `Get-Service AmazonSSMAgent`	Im PowerShell Administratormodus ausführen: `Start-Service AmazonSSMAgent`

Weitere Informationen

- Arbeiten mit SSM Agent auf EC2-Instances für Linux
- Arbeiten mit SSM Agent EC2 On-Instances für Windows Server
- Überprüfen der SSM Agent-Versionsnummer

Überprüfen der SSM Agent-Versionsnummer

Bestimmte AWS Systems Manager Funktionen erfordern die Installation einer Mindestversion von Systems Manager Agent (SSM Agent) auf Ihren verwalteten Knoten. Sie können die aktuell installierte SSM Agent-Version auf Ihren verwalteten Knoten über die Systems-Manager-Konsole abrufen oder sich bei Ihren verwalteten Knoten anmelden.

> **Note**
>
> Einzelheiten zu früheren Versionen finden Sie in den SSM AgentVersionshinweisen unter GitHub.

In den folgenden Verfahren wird beschrieben, wie Sie die aktuell installierte SSM Agent-Version auf Ihren verwalteten Knoten abrufen.

So ermitteln Sie die Versionsnummer von SSM Agent auf einem verwalteten Knoten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Notieren Sie sich in SSM Agent-Version die Nummer der Agent-Version.

So rufen Sie die aktuell installierte SSM Agent-Version innerhalb des Betriebssystems ab

Wählen Sie aus den folgenden Registerkarten aus, um die aktuell installierte SSM Agent-Version innerhalb des Betriebssystems abzurufen.

Amazon Linux 2 and Amazon Linux 2023

> **Note**
>
> Dieser Befehl variiert je nach Paketmanager für Ihr Betriebssystem.

1. Melden Sie sich bei Ihrem verwalteten Knoten an.
2. Führen Sie den folgenden Befehl aus.

```
yum info amazon-ssm-agent
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück.

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Installed Packages
Name        : amazon-ssm-agent
Arch        : x86_64
Version     : 3.0.655.0
```

Debian Server

1. Melden Sie sich bei Ihrem verwalteten Knoten an.
2. Führen Sie den folgenden Befehl aus.

```
apt list amazon-ssm-agent
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück.

```
apt list amazon-ssm-agent
Listing... Done
amazon-ssm-agent/now 3.0.655.0-1 amd64 [installed,local]

3.0.655.0 is the version of SSM agent
```

macOS

1. Melden Sie sich bei Ihrem verwalteten Knoten an.
2. Führen Sie den folgenden Befehl aus.

```
pkgutil --pkg-info com.amazon.aws.ssm
```

RHEL

1. Melden Sie sich bei Ihrem verwalteten Knoten an.
2. Führen Sie den folgenden Befehl auf RHEL 7, 8, 9 und 10 aus.

```
yum info amazon-ssm-agent
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück.

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Installed Packages
Name        : amazon-ssm-agent
Arch        : x86_64
Version     : 3.0.655.0
```

Führen Sie den folgenden Befehl für das DNF-Paketdienstprogramm aus.

```
dnf info amazon-ssm-agent
```

SLES

1. Melden Sie sich bei Ihrem verwalteten Knoten an.
2. Führen Sie den folgenden Befehl auf SLES 15.x aus.

```
zypper info amazon-ssm-agent
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück.

```
Loading repository data...
Reading installed packages...
```

```
Information for package amazon-ssm-agent:
----------------------------------------
Repository : @System
Name : amazon-ssm-agent
Version : 3.0.655.0-1
```

Ubuntu Server

> **Note**
>
> Ob Ihre Ubuntu Server 16.04-Instance Deb- oder Snap-Pakete verwendet, finden Sie unter Manuelle Installation von SSM Agent auf Ubuntu Server-Instances heraus.

1. Melden Sie sich bei Ihrem verwalteten Knoten an.
2. Führen Sie den folgenden Befehl auf Ubuntu Server 16.04 64-Bit (mit dem deb-Installationsprogrammpaket) aus.

   ```
   apt list amazon-ssm-agent
   ```

   Dieser Befehl gibt etwa die folgende Ausgabe zurück.

   ```
   apt list amazon-ssm-agent
   Listing... Done
   amazon-ssm-agent/now 3.0.655.0-1 amd64 [installed,local]

   3.0.655.0 is the version of SSM agent
   ```

   Führen Sie den folgenden Befehl für Ubuntu Server 16.04 LTS 64-Bit (Snap), 18.04, 20.04, 22.04 LTS, 23.10, 24.04 LTS, 24.0 und 25.04 aus.

   ```
   sudo snap list amazon-ssm-agent
   ```

   Dieser Befehl gibt etwa die folgende Ausgabe zurück.

   ```
   snap list amazon-ssm-agent
   Name Version Rev Tracking Publisher Notes
   amazon-ssm-agent 3.0.529.0 3552 latest/stable/… aws# classic-
   ```

```
3.0.529.0 is the version of SSM agent
```

Windows

1. Melden Sie sich bei Ihrem verwalteten Knoten an.
2. Führen Sie den folgenden PowerShell Befehl aus.

   ```
   & "C:\Program Files\Amazon\SSM\amazon-ssm-agent.exe" -version
   ```

   Dieser Befehl gibt etwa die folgende Ausgabe zurück.

   ```
   SSM Agent version: 3.1.804.0
   ```

Wir empfehlen, die neueste SSM Agent-Version zu verwenden, damit Sie von neuen oder aktualisierten Funktionen profitieren können. Um sicherzustellen, dass auf Ihren verwalteten Instanzen immer die neueste up-to-date Version von ausgeführt wirdSSM Agent, können Sie den Aktualisierungsprozess von automatisierenSSM Agent. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent.

Anzeigen von SSM Agent-Protokollen

AWS Systems Manager Agent (SSM Agent) schreibt Informationen über Ausführungen, Befehle, geplante Aktionen, Fehler und den Integritätsstatus in Protokolldateien auf jedem verwalteten Knoten. Sie können Protokolldateien anzeigen, indem Sie manuell eine Verbindung zu einem verwalteten Knoten herstellen, oder Sie können Protokolle automatisch an Amazon CloudWatch Logs senden. Weitere Informationen zum Senden von Protokollen an CloudWatch Logs finden Sie unterEinloggen und Überwachen AWS Systems Manager.

Sie können SSM Agent-Protokolle auf verwalteten Knoten in den folgenden Speicherorten anzeigen.

Linux and macOS

```
/var/log/amazon/ssm/
```

Windows

```
%PROGRAMDATA%\Amazon\SSM\Logs\
```

Für Linux-verwaltete Knoten werden die Dateien SSM Agent, `stderr` und `stdout` in das folgende Verzeichnis geschrieben: `/var/lib/amazon/ssm/`.

Für Windows-verwaltete Knoten werden die Dateien SSM Agent, `stderr` und `stdout` in das folgende Verzeichnis geschrieben: `%PROGRAMDATA%\Amazon\SSM\InstanceData\`.

Informationen zur Aktivierung der SSM Agent-Debug-Protokollierung finden Sie unter Zulassen von SSM Agent-Debug-Protokollierung.

Weitere Informationen über die `cihub/seelog`-Konfiguration finden Sie in der Seelog Wiki auf GitHub. Beispiele für `cihub/seelog` Konfigurationen finden Sie im cihub/seelong examples-Repository auf GitHub.

Zulassen von SSM Agent-Debug-Protokollierung

Mit den folgenden Verfahren können Sie die SSM Agent-Debug-Protokollierung auf Ihren verwalteten Knoten erlauben.

Linux and macOS

So erlauben Sie die SSM Agent-Debug-Protokollierung auf Linux- und macOS-verwalteten Knoten

1. Verwenden Sie Session Manager entweder ein Tool in AWS Systems Manager, um eine Verbindung zu dem verwalteten Knoten herzustellen, für den Sie die Debug-Protokollierung zulassen möchten, oder melden Sie sich am verwalteten Knoten an. Weitere Informationen finden Sie unter Arbeiten mit Session Manager.
2. Suchen Sie die Datei seelog.xml.template.

   Linux:

   Bei den meisten von Linux verwalteten Knotentypen befindet sich die Datei im Verzeichnis `/etc/amazon/ssm/seelog.xml.template`.

   Unter Ubuntu Server 20.04, 18.04 und 16.04 LTS befindet sich die Datei im Verzeichnis `/snap/amazon-ssm-agent/current/seelog.xml.template`. Kopieren Sie diese Datei aus dem `/snap/amazon-ssm-agent/current/`-Verzeichnis in das `/etc/amazon/ssm/`-Verzeichnis, bevor Sie Änderungen vornehmen.

   macOS:

Bei macOS-Instance-Typen befindet sich die Datei im Verzeichnis `/opt/aws/ssm/seelog.xml.template`.

3. Ändern Sie den Dateinamen von `seelog.xml.template` in `seelog.xml`.

> **Note**
>
> Unter Ubuntu Server 20.04, 18.04 und 16.04 LTS muss die Datei `seelog.xml` im Verzeichnis `/etc/amazon/ssm/` erstellt werden. Sie können dieses Verzeichnis und die Datei mit den folgenden Befehlen erstellen.
>
> ```
> sudo mkdir -p /etc/amazon/ssm
> ```
>
> ```
> sudo cp -p /snap/amazon-ssm-agent/current/seelog.xml.template /etc/amazon/ssm/seelog.xml
> ```

4. Bearbeiten Sie die Datei `seelog.xml`, um das Standardverhalten für die Protokollierung zu ändern. Ändern Sie den Wert für Mindeststufe von Info in Debuggen wie im folgenden Beispiel gezeigt.

   ```
   <seelog type="adaptive" mininterval="2000000" maxinterval="100000000" critmsgcount="500" minlevel="debug">
   ```

5. (Optional) Starten Sie SSM Agent mit dem folgenden Befehl neu.

   Linux:

   ```
   sudo service amazon-ssm-agent restart
   ```

   macOS:

   ```
   sudo /opt/aws/ssm/bin/amazon-ssm-agent restart
   ```

Windows

So erlauben Sie die SSM Agent-Debug-Protokollierung auf Windows Server-verwalteten Knoten

1. Verwenden Sie entweder Session Manager, um eine Verbindung mit dem verwalteten Knoten herzustellen, für den Sie die Debug-Protokollierung aktivieren möchten, oder melden Sie sich bei dem verwalteten Knoten an. Weitere Informationen finden Sie unter Arbeiten mit Session Manager.
2. Erstellen Sie eine Kopie der Datei seelog.xml.template. Ändern Sie den Namen der Kopie auf seelog.xml. Die Datei befindet sich im folgenden Verzeichnis.

   ```
   %PROGRAMFILES%\Amazon\SSM\seelog.xml.template
   ```

3. Bearbeiten Sie die Datei `seelog.xml`, um das Standardverhalten für die Protokollierung zu ändern. Ändern Sie den Wert für Mindeststufe von Info in Debuggen wie im folgenden Beispiel gezeigt.

   ```
   <seelog type="adaptive" mininterval="2000000" maxinterval="100000000"
   critmsgcount="500" minlevel="debug">
   ```

4. Suchen Sie den folgenden Eintrag.

   ```
   filename="{{LOCALAPPDATA}}\Amazon\SSM\Logs\{{EXECUTABLENAME}}.log"
   ```

   Ändern Sie diesen Eintrag, sodass der folgende Pfad verwendet wird.

   ```
   filename="C:\ProgramData\Amazon\SSM\Logs\{{EXECUTABLENAME}}.log"
   ```

5. Suchen Sie den folgenden Eintrag.

   ```
   filename="{{LOCALAPPDATA}}\Amazon\SSM\Logs\errors.log"
   ```

   Ändern Sie diesen Eintrag, sodass der folgende Pfad verwendet wird.

   ```
   filename="C:\ProgramData\Amazon\SSM\Logs\errors.log"
   ```

6. Starten Sie SSM Agent mit dem folgenden PowerShell Befehl im Administratormodus neu.

   ```
   Restart-Service AmazonSSMAgent
   ```

Einschränken des Zugriffs auf Befehle auf Stammebene durch SSM Agent

AWS Systems Manager-Agent (SSM Agent) läuft auf Instances der Amazon Elastic Compute Cloud (Amazon EC2) und anderen Maschinentypen in Hybrid- und Multi-Cloud-Umgebungen mit Root-Berechtigungen (Linux) oder SYSTEM-Berechtigungen (Windows Server). Da dies die höchste Stufe der Systemzugriffsberechtigungen ist, verfügt jede vertrauenswürdige Entität, der die Berechtigung zum Senden von Befehlen an den SSM Agent erteilt wurde, über Root- oder SYSTEM-Berechtigungen. (In AWS wird eine vertrauenswürdige Entität, die Aktionen ausführen und auf Ressourcen in AWS zugreifen kann, als Prinzipal bezeichnet. Ein Prinzipal kann ein Root-Benutzer des AWS-Kontos, Benutzer oder eine Rolle sein.)

Dieses Zugriffslevel ist erforderlich, damit ein Prinzipal autorisierte Systems Manager-Befehle an den SSM Agent senden kann, es ermöglicht einem Prinzipal aber auch, bösartigen Code auszuführen, indem potenzielle Schwachstellen in SSM Agent ausgenutzt werden.

Insbesondere sollten die Berechtigungen zum Ausführen der Befehle SendCommand und StartSession sorgfältig beschränkt werden. Eine guter erster Schritt ist es, Berechtigungen für jeden Befehl nur für bestimmte Prinzipale in Ihrer Organisation zu gewähren. Allerdings empfehlen wir, Ihren Sicherheitsstatus weiter zu verbessern, indem Sie einschränken, auf welchen verwalteten Knoten ein Prinzipal diese Befehle ausführen kann. Dies kann in der IAM-Richtlinie erfolgen, die dem Prinzipal zugeordnet ist. In die IAM-Richtlinie können Sie eine Bedingung einfügen, mit der der Benutzer nur Befehle auf verwalteten Knoten ausführen kann, die mit spezifischen Tags oder Kombinationen von Tags markiert sind.

Nehmen wir an, Sie haben zwei Serverflotten, eine für Tests und eine für die Produktion. In der IAM-Richtlinie, die für nachrangige Ingenieure gilt, geben Sie an, dass sie Befehle nur auf Instances ausführen können, die mit `ssm:resourceTag/testServer` gekennzeichnet sind. Aber für eine kleinere Gruppe von leitenden Ingenieuren, die alle Instances zugreifen können sollten, gewähren Sie Zugriff auf Instances, die mit `ssm:resourceTag/testServer` und `ssm:resourceTag/productionServer` gekennzeichnet sind.

Mit diesem Ansatz wird nachrangigen Ingenieuren, die versuchen, einen Befehl auf einer Produktions-Instance auszuführen, der Zugriff verweigert, da ihre zugewiesenen IAM-Richtlinie keinen expliziten Zugriff auf Instances zulässt, die mit `ssm:resourceTag/productionServer` gekennzeichnet sind.

Weitere Informationen und Beispiele finden Sie in den folgenden Themen:

- Den Zugriff von Run Command anhand von Tags beschränken
- Beschränkung des Sitzungszugriffs auf Instance-Tags

Automatisieren von Updates für SSM Agent

AWS veröffentlicht eine neue Version von AWS Systems Manager Agent (SSM Agent), wenn wir Systems Manager Manager-Tools hinzufügen oder aktualisieren. Wenn Ihre verwalteten Knoten eine ältere Version des Agents verwenden, können Sie weder die neuen Tools nutzen noch von den aktualisierten Tools profitieren. Aus diesen Gründen empfehlen wir, die Aktualisierung von SSM Agent auf Ihren verwalteten Knoten mit einer der folgenden Methoden zu automatisieren.

Agent-Updates auf dem Bottlerocket-Betriebssystem

SSM Agent auf dem Bottlerocket-Betriebssystem kann nicht mit dem Befehlsdokument `AWS-UpdateSSMAgent` von Systems Manager aktualisiert werden. Updates werden im Bottlerocket-Control-Container verwaltet. Weitere Informationen finden Sie unter Bottlerocket Control Container und Bottlerocket-Update-Infrastruktur auf GitHub.

macOS-Versionsanforderung

Wenn auf einer Instance macOS-Version 11.0 (Big Sur) oder höher ausgeführt wird, muss die Instance über die SSM Agent-Version 3.1.941.0 oder höher verfügen, um das AWS-UpdateSSMAgent-Dokument auszuführen. Wenn auf der Instance eine Version von SSM Agent ausgeführt wird, die vor 3.1.941.0 veröffentlicht wurde, aktualisieren Sie Ihr SSM Agent, um die AWS-UpdateSSMAgent auszuführen, indem Sie die `brew update`- und `brew upgrade amazon-ssm-agent`-Befehle ausführen.

Methode	Details
Automatisierte Aktualisierung auf allen verwalteten Knoten mit einem Klick (Empfohlen)	Sie können alle verwalteten Knoten in Ihrem so konfigurieren AWS-Konto , dass automatisch nach neuen Versionen von gesucht und diese heruntergeladen SSM Agent werden. Wählen Sie dazu auf der Registerkarte Einstellungen in Fleet Manager die Option Automatische Aktualisierung von SSM Agent, wie nachfolgend in diesem Thema beschrieben.

Methode	Details
Globale oder selektive Aktualisierung	Sie können ein Tool in verwenden State Manager AWS Systems Manager, um eine Zuordnung zu erstellen, die automatisch heruntergeladen und SSM Agent auf Ihren verwalteten Knoten installiert wird. Wenn Sie die Unterbrechung Ihrer Workloads begrenzen möchten, können Sie ein Systems Manager-Wartungsfenster erstellen, um die Installation in festgelegten Zeiträumen durchzuführen. Bei beiden Methoden können Sie entweder eine globale Aktualisierungs-Konfiguration für alle Ihre verwalteten Knoten erstellen oder auswählen, welche Instances aktualisiert werden. Informationen zum Erstellen einer State Manager-Zuordnung finden Sie unter Exemplarische Vorgehensweise: Automatisches Update SSM Agent mit AWS CLI. Weitere Informationen zum Erstellen eines Wartungsfensters, finden Sie unter Tutorial: Erstellen Sie ein Wartungsfenster zum Patchen über die Konsole.
Globale oder selektive Aktualisierung für neue Umgebungen	Wenn Sie mit Systems Manager beginnen, empfehlen wir Ihnen, den Update Systems Manager (SSM) -Agenten alle zwei Wochen zu verwendenQuick Setup, ein Tool in AWS Systems Manager. Quick Setupermöglicht es Ihnen, entweder eine globale Update-Konfiguration für alle Ihre verwalteten Knoten zu erstellen oder selektiv auszuwählen, welche verwalteten Knoten aktualisiert werden. Weitere Informationen finden Sie unter Amazon-EC2-Host-Verwaltung mit Quick Setup einrichten.

Wenn Sie es vorziehen, Ihre verwalteten Knoten manuell zu aktualisierenSSM Agent, können Sie Benachrichtigungen abonnieren, die AWS veröffentlicht werden, wenn eine neue Version des Agenten veröffentlicht wird. Weitere Informationen finden Sie unter Abonnieren von SSM Agent-Benachrichtigungen. Nachdem Sie Benachrichtigungen abonniert haben, können Sie Run Command verwenden, um einen oder mehrere verwalteten Knoten manuell mit der neuesten Version zu aktualisieren. Weitere Informationen finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command.

Automatische Aktualisierung von SSM Agent

Sie können Systems Manager so konfigurieren, dass er SSM Agent automatisch auf allen Linux-basierten und Windows-basierten verwalteten Knoten in Ihrem AWS-Konto aktualisiert. Wenn Sie diese Option aktivieren, sucht Systems Manager automatisch alle zwei Wochen nach einer neuen Version des Agenten. Wenn es eine neue Version gibt, aktualisiert Systems Manager den Agent mit Hilfe des SSM-Dokuments `AWS-UpdateSSMAgent` automatisch auf die neueste freigegebene Version. Wir empfehlen Ihnen, diese Option zu wählen, um sicherzustellen, dass auf Ihren verwalteten Knoten immer die neueste up-to-date Version von ausgeführt wirdSSM Agent.

> **Note**
>
> Wenn Sie einen `yum`-Befehl verwenden, um SSM Agent auf einem verwalteten Knoten zu aktualisieren, nachdem der Agent mit dem SSM-Dokument `AWS-UpdateSSMAgent` installiert oder aktualisiert wurde, sehen Sie möglicherweise die folgende Meldung: `"Warning: RPMDB altered outside of yum."` Diese Meldung wird erwartet und kann ignoriert werden.

So aktualisieren Sie SSM Agent automatisch

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Fleet Manager aus.
3. Wählen Sie die Registerkarte Einstellungen.
4. Wählen Sie im Bereich Automatische Aktualisierung des Agenten die Option Automatische Aktualisierung von SSM Agent.

Um zu ändern, auf welche Version von SSM Agent Ihre Flotte aktualisiert wird, wählen Sie unter Agent auto update (Automatische Agent-Aktualisierung) auf der Registerkarte Settings (Einstellungen) Edit (Bearbeiten). Geben Sie dann bei Version unter Parameter die Versionsnummer von SSM Agent ein, auf die Sie aktualisieren möchten. Ist hierfür nichts angegeben, wird der Agent auf die neueste Version aktualisiert.

Um die automatische Bereitstellung aktualisierter Versionen von SSM Agent auf alle verwalteten Knoten in Ihrem Konto zu beenden, wählen Sie unter Agent auto update (Automatische Agent-Aktualisierung) auf der Registerkarte Settings (Einstellungen) Delete (Löschen). Diese Aktion löscht die State Manager-Zuordnung gelöscht, durch die SSM Agent auf Ihren verwalteten Knoten automatisch aktualisiert wird.

Abonnieren von SSM Agent-Benachrichtigungen

Amazon Simple Notification Service (Amazon SNS) kann Sie benachrichtigen, wenn neue Versionen von AWS Systems Manager-Agent (SSM Agent) veröffentlicht werden. Führen Sie die folgenden Schritte durch, um diese Benachrichtigungen zu abonnieren.

> **Tip**
>
> Sie können zudem über die Seite SSM Agent-Versionshinweise auf GitHub auch Benachrichtigungen abonnieren.

So abonnieren Sie SSM Agent-Benachrichtigungen

1. Öffnen Sie die Amazon SNS-Konsole unter https://console.aws.amazon.com/sns/v3/home.
2. Wählen Sie über die Regionsauswahl in der Navigationsleiste USA Ost (Nord-Virginia) aus, falls diese Option nicht bereits ausgewählt ist. Sie müssen diese AWS-Region auswählen, da die Amazon SNS-Benachrichtigungen für SSM Agent, die Sie abonnieren möchten, nur von dieser Region generiert werden.
3. Wählen Sie im Navigationsbereich Subscriptions aus.
4. Wählen Sie Create subscription.
5. Führen Sie unter Create subscription (Abonnement erstellen) die folgenden Schritte aus:

   a. Verwenden Sie für Topic ARN den folgenden Amazon-Ressourcennamen (ARN):

      ```
      arn:aws:sns:us-east-1:720620558202:SSM-Agent-Update
      ```

b. Wählen Sie für Protokoll `Email` oder `SMS` aus.

c. Geben Sie für Endpoint (Endpunkt) je nachdem, ob Sie im vorherigen Schritt `Email` oder `SMS` gewählt haben, eine E-Mail-Adresse oder eine Vorwahl und Nummer ein, um Benachrichtigungen zu erhalten.

d. Wählen Sie Create subscription (Abonnement erstellen) aus.

6. Wenn Sie `Email` auswählen, erhalten Sie eine Nachricht, in der Sie aufgefordert werden, Ihr Abonnement zu bestätigen. Öffnen Sie die Nachricht und befolgen Sie die Anweisungen, um Ihr Abonnement abzuschließen.

Sobald eine neue Version von SSM Agent veröffentlicht wird, senden wir den Abonnenten Benachrichtigungen. Wenn Sie diese Benachrichtigungen nicht mehr erhalten möchten, führen Sie die folgenden Schritte aus, um sich abzumelden.

So melden Sie sich von den SSM Agent-Benachrichtigungen ab

1. Öffnen Sie die Amazon SNS-Konsole.
2. Wählen Sie im Navigationsbereich Subscriptions aus.
3. Wählen Sie das Abonnement und dann Delete (Löschen) aus. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Delete (Löschen) aus.

Fehlerbehebung für SSM Agent

Wenn Sie Probleme bei der Ausführung von Vorgängen auf Ihren verwalteten Knoten haben, liegt möglicherweise ein Problem mit AWS Systems Manager Agent (SSM Agent) vor. Verwenden Sie die folgenden Informationen, um die SSM Agent-Protokolldateien anzuzeigen und Fehler für den Agent zu beheben. Wenn Ihr Agent nicht reagiert oder seltener kommuniziert, finden Sie weitere Informationen unter Grundlegendes zum SSM Agent-Ruhezustand.

Themen

- SSM Agent ist veraltet
- Probleme mithilfe von SSM Agent-Protokolldateien beheben
- Agent-Protokolldateien werden nicht gedreht (Windows)
- Keine Verbindung mit SSM-Endpunkten möglich
- Ihre VPC-Konfiguration überprüfen

- Ihre VPC-DNS-bezogenen Attribute überprüfen
- Die Eingangsregeln für Endpunkt-Sicherheitsgruppen überprüfen
- Verwenden Sie ssm-cli, um die Verfügbarkeit von verwalteten Knoten zu überprüfen

SSM Agent ist veraltet

Wenn Systems Manager neue Tools hinzugefügt oder Aktualisierungen an den vorhandenen Tools vorgenommen werden, wird eine neue Version von SSM Agent veröffentlicht. Wenn Sie nicht die neueste Version des Agenten verwenden, kann dies dazu führen, dass der verwaltete Knoten nicht die zahlreichen Tools und Features von Systems Manager verwendet. Aus diesem Grund empfehlen wir, dass Sie den Prozess zur Aktualisierung von SSM Agent in Ihren Maschinen automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Abonnieren Sie die SSM Agent-Versionshinweise-Seite in GitHub, um Benachrichtigungen über SSM Agent-Updates zu erhalten.

Probleme mithilfe von SSM Agent-Protokolldateien beheben

SSM Agent protokolliert Informationen in den folgenden Dateien. Die Informationen in diesen Dateien können Ihnen auch bei der Problembehandlung behilflich sein. Weitere Informationen zu SSM Agent-Protokolldateien, einschließlich der Aktivierung der Debug-Protokollierung, finden Sie unter Anzeigen von SSM Agent-Protokollen.

> **Note**
>
> Wenn Sie diese Protokolle mithilfe von Windows File Explorer anzeigen möchten, überprüfen Sie, dass Sie die Anzeige von ausgeblendeten Dateien und Systemdateien unter „Folder Options“ (Ordneroptionen) aktiviert ist.

Unter Windows

- `%PROGRAMDATA%\Amazon\SSM\Logs\amazon-ssm-agent.log`
- `%PROGRAMDATA%\Amazon\SSM\Logs\errors.log`

Unter Linux und macOS

- `/var/log/amazon/ssm/amazon-ssm-agent.log`

- `/var/log/amazon/ssm/errors.log`

Für Linux-verwaltete Knoten finden Sie möglicherweise weitere Informationen in der `messages`-Datei, die in das folgende Verzeichnis geschrieben ist: `/var/log`.

Weitere Informationen zur Fehlerbehebung mithilfe von Agentenprotokollen finden Sie unter Wie verwende ich SSM Agent-Protokolle zur Behebung von Problemen mit SSM Agent in meiner verwalteten Instance? im AWS re:POST Knowledge Center.

Agent-Protokolldateien werden nicht gedreht (Windows)

Wenn Sie die datumsbasierte Drehung der Protokolldatei in der Datei seelog.xml (auf von Windows Server verwalteten Knoten) angeben und die Protokolle nicht gedreht werden, geben Sie den `fullname=true`-Parameter an. Hier finden Sie ein Beispiel für eine seelog.xml -Konfigurationsdatei mit angegebenem `fullname=true`-Parameter.

```
<seelog type="adaptive" mininterval="2000000" maxinterval="100000000"
 critmsgcount="500" minlevel="debug">
   <exceptions>
      <exception filepattern="test*" minlevel="error" />
   </exceptions>
   <outputs formatid="fmtinfo">
      <console formatid="fmtinfo" />
      <rollingfile type="date" datepattern="200601021504" maxrolls="4" filename="C:
\ProgramData\Amazon\SSM\Logs\amazon-ssm-agent.log" fullname=true />
      <filter levels="error,critical" formatid="fmterror">
         <rollingfile type="date" datepattern="200601021504" maxrolls="4" filename="C:
\ProgramData\Amazon\SSM\Logs\errors.log" fullname=true />
      </filter>
   </outputs>
   <formats>
      <format id="fmterror" format="%Date %Time %LEVEL [%FuncShort @ %File.%Line] %Msg
%n" />
      <format id="fmtdebug" format="%Date %Time %LEVEL [%FuncShort @ %File.%Line] %Msg
%n" />
      <format id="fmtinfo" format="%Date %Time %LEVEL %Msg%n" />
   </formats>
</seelog>
```

Keine Verbindung mit SSM-Endpunkten möglich

SSM Agent muss ausgehenden HTTPS-Verkehr (Port 443) zu den folgenden Endpunkten zulassen:

- `ssm.region.amazonaws.com`
- `ssmmessages.region.amazonaws.com`

*region*steht für die Kennung einer Region, die von AWS-Region unterstützt wird AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

> **Note**
>
> Vor 2024 war `ec2messages.region.amazonaws.com` ebenfalls erforderlich. Bei Produkten, die vor 2024 AWS-Regionen gestartet wurden, `ssmmessages.region.amazonaws.com` ist das Zulassen von Datenverkehr weiterhin erforderlich, aber optional. `ec2messages.region.amazonaws.com` Für Regionen, die im Jahr 2024 oder später gestartet werden, ist das Zulassen von Datenverkehr zu `ssmmessages.region.amazonaws.com` erforderlich, aber `ec2messages.region.amazonaws.com`-Endpunkte werden für diese Regionen nicht unterstützt.

SSM Agentfunktioniert nicht, wenn es nicht wie beschrieben mit den vorherigen Endpunkten kommunizieren kann, selbst wenn Sie AWS provided Amazon Machine Images (AMIs) wie Amazon Linux 2 oder Amazon Linux 2023 verwenden. Ihre Netzwerkkonfiguration muss über einen offenen Internetzugang verfügen, oder Sie müssen benutzerdefinierte Virtual Private Cloud (VPC)-Endpunkte konfiguriert haben. Wenn Sie keinen benutzerdefinierten VPC-Endpunkt erstellen möchten, überprüfen Sie Ihre Internet-Gateways oder NAT-Gateways. Weitere Informationen dazu, wie Sie VPC-Endpunkte verwalten, finden Sie unter Verbessern Sie die Sicherheit von EC2 Instanzen mithilfe von VPC-Endpunkten für Systems Manager.

Ihre VPC-Konfiguration überprüfen

Wenn Sie eine Virtual Private Cloud (VPC) verwenden, müssen Ihre VPC-Endpoints zur Verwaltung von EC2 Instanzen mit Systems Manager ordnungsgemäß für`ssm.region.amazonaws.com`, konfiguriert sein und in einigen Fällen weiter oben in diesem

Thema unterssmmessages.*region*.amazonaws.com, erklärt werden. Keine Verbindung mit SSM-Endpunkten möglich ec2messages.*region*.amazonaws.com

> **Note**
>
> Die Alternative zur Verwendung eines VPC-Endpunkts ist das Aktivieren von ausgehendem Internetzugriff auf Ihren verwalteten Instances. In diesem Fall müssen die verwalteten Instances auch ausgehenden HTTPS-Datenverkehr (Port 443) zu den folgenden Endpunkten zulassen:
>
> - ssm.*region*.amazonaws.com
> - ssmmessages.*region*.amazonaws.com
> - ec2messages.*region*.amazonaws.com
>
> SSM Agent initiiert alle Verbindungen zum Systems Manager-Service in der Cloud. Aus diesem Grund müssen Sie Ihre Firewall nicht so konfigurieren, dass eingehender Datenverkehr zu Ihren Instances für Systems Manager zugelassen wird. Weitere Informationen über Anrufe an diese Endpunkte finden Sie unter Referenz: ec2messages, ssmmessages und andere API-Operationen.

Gehen Sie wie folgt vor, um Probleme mit Ihren VPC-Endpunkten zu beheben:

- Stellen Sie sicher, dass VPC-Endpunkte auf VPC-Ebene enthalten sind. Wenn der VPC-Endpunkt mit einem bestimmten Servicenamen nicht auf der VPC gefunden wird, überprüfen Sie zunächst, ob die DNS-Unterstützung auf VPC-Ebene aktiviert ist. Erstellen Sie als Nächstes einen neuen VPC-Endpunkt und ordnen Sie ihn jeweils einem Subnetz in jeder Availability Zone zu.
- Stellen Sie sicher, dass ein privater DNS-Name auf VPC-Endpunktebene aktiviert ist. Private DNS-Namen sind standardmäßig aktiviert, wurden aber möglicherweise irgendwann manuell deaktiviert.
- Stellen Sie sicher, dass die vorhandenen VPC-Endpunkte dem richtigen Subnetz zugeordnet sind. Stellen Sie außerdem sicher, dass die VPCE bereits einem Subnetz in dieser Availability Zone zugeordnet ist.

Weitere Informationen finden Sie unter den folgenden Themen:

- Zugriff auf AWS-Service über einen Schnittstellen-VPC-Endpunkt im AWS PrivateLink -Handbuch

- Ordnen Sie in der Anleitung einen privaten DNS-Namen zu AWS PrivateLink
- Verbessern Sie die Sicherheit von EC2 Instanzen mithilfe von VPC-Endpunkten für Systems Manager

Ihre VPC-DNS-bezogenen Attribute überprüfen

Wenn Sie eine Virtual Private Cloud (VPC) verwenden, müssen Sie im Rahmen der Überprüfung Ihrer VPC-Konfiguration sicherstellen, dass die Attribute `enableDnsSupport` und `enableDnsHostnames` aktiviert sind.

Sie können diese Attribute mit der Amazon EC2 Modify VPCAttribute API-Aktion oder dem AWS CLI Befehl aktivieren modify-vpc-attribute.

Informationen zur Aktivierung dieser Attribute in der Amazon-VPC-Konsole finden Sie unter Anzeigen und Aktualisieren von DNS-Attributen für Ihre VPC im Amazon-VPC-Benutzerhandbuch.

> **Note**
>
> Die Alternative zur Verwendung eines VPC-Endpunkts ist das Aktivieren von ausgehendem Internetzugriff auf Ihren verwalteten Instances. In diesem Fall müssen die verwalteten Instances auch ausgehenden HTTPS-Datenverkehr (Port 443) zu den folgenden Endpunkten zulassen:
>
> - `ssm.region.amazonaws.com`
> - `ssmmessages.region.amazonaws.com`
> - `ec2messages.region.amazonaws.com`
>
> SSM Agent initiiert alle Verbindungen zum Systems Manager-Service in der Cloud. Aus diesem Grund müssen Sie Ihre Firewall nicht so konfigurieren, dass eingehender Datenverkehr zu Ihren Instances für Systems Manager zugelassen wird. Weitere Informationen über Anrufe an diese Endpunkte finden Sie unter Referenz: ec2messages, ssmmessages und andere API-Operationen.

Die Eingangsregeln für Endpunkt-Sicherheitsgruppen überprüfen

Stellen Sie sicher, dass alle VPC-Endpunkte, die Sie konfiguriert haben (`ssm`, `ssmmessages` und `ec2messages`), eine Eingangsregel für ihre Sicherheitsgruppen enthalten, um Datenverkehr über Port 443 zuzulassen. Bei Bedarf können Sie in der VPC eine neue Sicherheitsgruppe mit einer Eingangsregel erstellen, um Datenverkehr auf Port 443 für den Classless Inter-Domain Routing (CIDR)-Block für die VPC zuzulassen. Nachdem Sie die Sicherheitsgruppe erstellt haben, fügen Sie sie jedem VPC-Endpunkt hinzu.

Weitere Informationen finden Sie unter den folgenden Themen:

- Wie erstelle ich VPC-Endpoints, sodass ich Systems Manager verwenden kann, um private EC2 Instances ohne Internetzugang zu verwalten? auf re:POST AWS
- VPC-CIDR-Blöcke im Amazon VPC-Benutzerhandbuch

Verwenden Sie **`ssm-cli`**, um die Verfügbarkeit von verwalteten Knoten zu überprüfen

Ab SSM Agent-Version 3.1.501.0 können Sie mit `ssm-cli` feststellen, ob ein verwalteter Knoten die primären Voraussetzungen erfüllt, um von Systems Manager verwaltet zu werden und in den Listen der verwalteten Knoten in Fleet Manager zu erscheinen. Die `ssm-cli` ist ein eigenständiges Befehlszeilentool, das in der SSM Agent-Installation enthalten ist. Es sind vorkonfigurierte Befehle enthalten, die die erforderlichen Informationen sammeln, um Ihnen bei der Diagnose zu helfen, warum eine EC2 Amazon-Instance oder ein EC2 Amazon-Computer, von dem Sie bestätigt haben, dass sie läuft, nicht in Ihren Listen der verwalteten Knoten in Systems Manager enthalten ist. Diese Befehle werden ausgeführt, wenn Sie die `get-diagnostics`-Option angeben.

Weitere Informationen finden Sie unter Problembehandlung bei der Verfügbarkeit von verwalteten Knoten mit `ssm-cli`.

Sicherheit in AWS Systems Manager

Cloud-Sicherheit genießt bei Amazon Web Services höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS-Services in der läuft AWS Cloud. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für geltenAWS Systems Manager, finden Sie AWS-Services unter Umfang nach Compliance-ProgrammenAWS-Services und unter .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS-Service , was Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, die Anforderungen Ihres Unternehmens und die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung anwenden können AWS Systems Manager. Die folgenden Themen veranschaulichen, wie Sie Systems Manager zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie erfahren auch, wie Sie andere verwenden können AWS-Services , die Ihnen bei der Überwachung und Sicherung Ihrer Systems Manager Ressourcen helfen.

Topics

- Datenschutz in AWS Systems Manager
- Datenperimeter in AWS Systems Manager
- Identity and Access Management für AWS Systems Manager
- Verwenden von serviceverknüpften Rollen für Systems Manager
- Protokollieren und Überwachen in AWS Systems Manager
- Compliance-Validierung für AWS Systems Manager
- Ausfallsicherheit in AWS Systems Manager
- Sicherheit der Infrastruktur in AWS Systems Manager
- Konfigurations- und Schwachstellenanalyse in AWS Systems Manager

- Bewährte Methoden für die Sicherheit für Systems Manager

Datenschutz in AWS Systems Manager

Dieser Datenschutz bezieht sich auf Daten bei der Übertragung (wenn sie zu oder von Systems Manager geschickt werden), ebenso wie im Ruhezustand (die in AWS-Rechenzentren gespeichert sind).

Das Modell der geteilten Verantwortung von AWS gilt für den Datenschutz in AWS Systems Manager. Wie in diesem Modell beschrieben, ist AWS verantwortlich für den Schutz der globalen Infrastruktur, in der die gesamte AWS Cloud ausgeführt wird. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS-Modell der geteilten Verantwortung und in der DSGVO im AWS-Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, AWS-Konto-Anmeldeinformationen zu schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS für die Kommunikation mit AWS-Ressourcen. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit AWS CloudTrail ein. Informationen zur Verwendung von CloudTrail-Trails zur Erfassung von AWS-Aktivitäten finden Sie unter Arbeiten mit CloudTrail-Trails im AWS CloudTrail-Benutzerhandbuch.
- Verwenden Sie AWS-Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder über eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit der Systems Manager oder anderen AWS-Services über die Konsole, API, AWS CLI oder AWS-SDKs arbeiten. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Datenverschlüsselung

Verschlüsselung im Ruhezustand

Parameter Store-Parameter

Zu den Parametertypen, die Sie in Parameter Store (einem Tool in AWS Systems Manager) erstellen können, gehören `String`, `StringList` und `SecureString`.

Alle Parameter, unabhängig von ihrem Typ, werden bei der Übertragung und im Ruhezustand verschlüsselt. Bei der Übertragung werden Parameter mithilfe von Transport Layer Security (TLS) verschlüsselt, um eine sichere HTTPS-Verbindung für API-Anfragen herzustellen. Im Ruhezustand werden sie mit einem AWS-eigener Schlüssel in AWS Key Management Service (AWS KMS) verschlüsselt. Weitere Informationen zur AWS-eigener Schlüssel-Verschlüsselung finden Sie im AWS Key Management Service-Entwicklerhandbuch unter AWS-eigene Schlüssel.

Der Typ `SecureString` bietet zusätzliche Verschlüsselungsoptionen und wird für alle sensiblen Daten empfohlen. Sie können aus den folgenden AWS KMS-Schlüsseltypen wählen, um den Wert eines `SecureString`-Parameters zu verschlüsseln und zu entschlüsseln:

- Der Von AWS verwalteter Schlüssel Ihres Kontos.
- Ein kundenseitig verwalteter Schlüssel (CMK), den Sie in Ihrem Konto erstellt haben
- Ein CMK in einem anderen AWS-Konto, der für Sie freigegeben wurde

Weitere Informationen zur AWS KMS-Verschlüsselung finden Sie im AWS Key Management Service-Entwicklerhandbuch.

Inhalt in S3-Buckets

Als Teil Ihrer Systems Manager-Vorgänge können Sie Daten in einen oder mehrere Amazon Simple Storage Service (Amazon S3)-Buckets hochladen oder speichern.

Informationen zur Verschlüsselung von S3-Buckets finden Sie unter Daten durch Verschlüsselung schützen und Datenschutz in Amazon S3 im Benutzerhandbuch zu Amazon Simple Storage Service.

Die folgenden Datentypen können Sie als Teil Ihrer Systems Manager-Aktivitäten hochladen oder in S3 Buckets speichern lassen:

- Die Ausgabe von Befehlen in Run Command, einem Tool in AWS Systems Manager
- Pakete in Distributor, einem Tool in AWS Systems Manager
- Patching-Protokolle in Patch Manager, einem Tool in AWS Systems Manager
- Patch Manager Patch-Überschreibungslisten
- Skripte oder Ansible Playbooks, die in einem Runbook-Workflow in Automation ausgeführt werden sollen, einem Tool in AWS Systems Manager
- Chef InSpec-Profile für die Verwendung mit Scans in Compliance, einem Tool in AWS Systems Manager
- AWS CloudTrail-Protokolle
- Sitzungsverlaufsprotokolle in Session Manager, einem Tool in AWS Systems Manager
- Berichte aus Explorer, einem Tool in AWS Systems Manager
- OpsData von OpsCenter, einem Tool in AWS Systems Manager
- AWS CloudFormation-Vorlagen für die Verwendung mit Automation-Workflows
- Compliance-Daten aus einem Resource Data Sync-Scan
- Ausgabe von Anforderungen zum Erstellen oder Bearbeiten von Zuordnungen in State Manager, einem Tool in AWS Systems Manager, auf verwalteten Knoten
- Benutzerdefinierte Systems Manager-Dokumente (SSM-Dokumente), die Sie mit dem AWS-verwalteten SSM-Dokument `AWS-RunDocument` ausführen können

CloudWatch Logs-Protokollgruppen

Als Teil Ihrer Systems Manager-Operationen könnten Sie sich dafür entscheiden, Daten in eine oder mehrere Amazon CloudWatch Logs-Protokollgruppen zu streamen.

Informationen zur Verschlüsselung von CloudWatch-Logs-Protokollgruppen finden Sie unter Verschlüsseln von Protokolldaten in CloudWatch Logs mit AWS Key Management Service im Benutzerhandbuch zu Amazon CloudWatch Logs.

Im Folgenden sind die Datentypen aufgeführt, die Sie möglicherweise als Teil Ihrer Systems Manager-Aktivitäten in eine CloudWatch Logs-Protokollgruppe gestreamt haben.

- Die Ausgabe der Run Command-Befehle
- Ausgabe von Skripten, die mit der `aws:executeScript`-Aktion in einem Automation-Runbooks ausgeführt werden
- Session Manager-Sitzungsverlaufsprotokolle
- Protokolle vom SSM Agent auf Ihren verwalteten Nodes

Verschlüsselung während der Übertragung.

Wir empfehlen, dass Sie ein Verschlüsselungsprotokoll wie Transport Layer Security (TLS) verwenden, um sensible Daten bei der Übertragung zwischen den Clients und Ihren Knoten zu verschlüsseln.

Systems Manager bietet die folgende Unterstützung für die Verschlüsselung Ihrer Daten während der Übertragung.

Verbindungen zu Systems Manager API-Endpunkten

Systems Manager-API-Endpunkte unterstützen ausschließlich sichere Verbindungen über HTTPS. Wenn Sie Systems Manager-Ressourcen mit der AWS-Managementkonsole, dem AWS SDK oder der Systems Manager-API verwalten, wird die gesamte Kommunikation mit Transport Layer Security (TLS) verschlüsselt. Eine vollständige Liste der API-Endpunkte finden Sie unter AWS-Service-Endpunkte im Allgemeine Amazon Web Services-Referenz.

Verwaltete Instances

AWS bietet sichere und private Konnektivität zwischen Amazon Elastic Compute Cloud (Amazon EC2)-Instances. Darüber hinaus wird Datenverkehr zwischen unterstützen Instances in einer Virtual Private Cloud (VPC) oder in per Peering verbundenen VPCs automatisch mithilfe von AEAD-Algorithmen mit 256-Bit-Verschlüsselung verschlüsselt. Das Verschlüsselungsfeature verwendet die Offload-Möglichkeiten der zugrunde liegenden Hardware ohne Auswirkungen auf die Netzwerkleistung. Unterstützte Instances: C5n, G4, I3en, M5dn, M5n, P3dn, R5dn und R5n.

Session Manager-Sitzungen

Standardmäßig verwendet Session Manager TLS 1.3 zum Verschlüsseln von Sitzungsdaten, die zwischen lokalen Computern von Benutzern in Ihrem Konto und Ihren EC2-Instances übertragen werden. Sie können die Daten während der Übertragung auch weiter verschlüsseln, indem Sie ein AWS KMS key verwenden, das in AWS KMS erstellt wurde. AWS KMS-Verschlüsselung ist verfügbar für `Standard_Stream`-, `InteractiveCommands`-, und `NonInteractiveCommands`-Sitzungstypen.

Run Command-Zugriff

Standardmäßig wird der Remote-Zugriff auf Ihre Knoten über Run Command mit TLS 1.3 verschlüsselt und Anfragen zum Verbindungsaufbau werden mit SigV4 signiert.

Richtlinie für den Datenverkehr zwischen Netzwerken

Sie können Amazon Virtual Private Cloud (Amazon VPC) verwenden, um Grenzen zwischen Ressourcen in Ihren verwalteten Knoten zu erstellen und den Datenverkehr zwischen ihnen, Ihrem On-Premises-Netzwerk und dem Internet zu steuern. Einzelheiten finden Sie unter Verbessern Sie die Sicherheit von EC2-Instances mithilfe von VPC-Endpunkten für Systems Manager.

Weitere Informationen zur Sicherheit der Amazon Virtual Private Cloud finden Sie unter Datenschutz des Internet-Datenverkehrs in Amazon VPC im Benutzerhandbuch Amazon VPC.

Datenperimeter in AWS Systems Manager

Bei einem Datenperimeter handelt es sich um eine Reihe präventiver Schutzmaßnahmen in Ihrer AWS Umgebung, die sicherstellen, dass nur vertrauenswürdige Identitäten aus den erwarteten Netzwerken und Ressourcen auf Ihre Daten zugreifen können. Wenn Sie Datenperimeterkontrollen implementieren, müssen Sie möglicherweise Ausnahmen für AWS diensteigene Ressourcen einbeziehen, auf die Systems Manager in Ihrem Namen zugreift.

Beispielszenario: SSM-Dokumentkategorien, S3-Bucket

Systems Manager greift auf einen AWS verwalteten S3-Bucket zu, um Informationen zur AWS Systems Manager-Documents Dokumentkategorie abzurufen. Dieser Bucket enthält Metadaten zu Dokumentkategorien, mit deren Hilfe SSM-Dokumente in der Konsole organisiert und klassifiziert werden können.

ARN-Muster für Ressourcen

```
arn:aws:s3:::ssm-document-categories-region
```

Regionale Beispiele:

- `arn:aws:s3:::ssm-document-categories-us-east-1`
- `arn:aws:s3:::ssm-document-categories-us-west-2`

- `arn:aws:s3:::ssm-document-categories-eu-west-1`
- `arn:aws:s3:::ssm-document-categories-ap-northeast-1`

Zugriff

Auf diese Ressource wird zugegriffen, wenn Sie SSM-Dokumente in der Systems Manager Manager-Konsole anzeigen oder wenn Sie APIs diese zum Abrufen von Dokumentmetadaten und Kategorien verwenden.

Gespeicherte Daten

Der Bucket enthält JSON-Dateien mit Definitionen und Metadaten für Dokumentkategorien. Diese Daten sind schreibgeschützt und enthalten keine kundenspezifischen Informationen.

Verwendete Identität

Systems Manager greift im Namen Ihrer Anfragen mithilfe von AWS Dienstanmeldedaten auf diese Ressource zu.

Erforderliche Berechtigungen

`s3:GetObject` für die Inhalte des Buckets.

Überlegungen zur Datenperimeter-Richtlinien

Bei der Implementierung von Datenperimeterkontrollen mithilfe von Service Control Policies (SCPs) oder VPC-Endpunktrichtlinien mit Bedingungen wie`aws:ResourceOrgID`, müssen Sie Ausnahmen für die AWS diensteigenen Ressourcen erstellen, die Systems Manager benötigt.

Wenn Sie beispielsweise ein SCP mit verwenden, `aws:ResourceOrgID` um den Zugriff auf Ressourcen außerhalb Ihrer Organisation einzuschränken, müssten Sie eine Ausnahme für den Bereich SSM-Dokumentkategorien hinzufügen.

Die Richtlinie müsste auf Ressourcen außerhalb Ihrer Organisation zugreifen, aber eine Ausnahme für die entsprechenden S3-Buckets enthalten, damit Systems Manager weiterhin ordnungsgemäß funktionieren kann.

Wenn Sie VPC-Endpunktrichtlinien verwenden, um den S3-Zugriff einzuschränken, müssten Sie ebenfalls sicherstellen, dass die Buckets für SSM-Dokumentkategorien über Ihre VPC-Endpunkte zugänglich sind.

Weitere Informationen

Weitere Informationen zu Datenperimetern finden Sie in den AWS folgenden Themen:

- Datenperimeter aktiviert. AWS
- Richten Sie mithilfe von Datenperimetern im IAM-Benutzerhandbuch Richtlinien für Berechtigungen ein
- Servicespezifische Anleitungen: AWS Systems Manager und Ressourcen, die dem Dienst gehören, befinden sich im Samples-Repository auf AWS GitHub

Identity and Access Management für AWS Systems Manager

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf AWS Ressourcen sicher zu kontrollieren. IAM-Administratoren steuern, wer authentifiziert (angemeldet) und autorisiert (Berechtigungen besitzt) ist, um Systems Manager Ressourcen zu nutzen. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

- Zielgruppe
- Authentifizierung mit Identitäten
- Verwalten des Zugriffs mit Richtlinien
- Funktionsweise von AWS Systems Manager mit IAM
- AWS Systems ManagerBeispiele für identitätsbasierte -Richtlinien
- AWS verwaltete Richtlinien für AWS Systems Manager
- Fehlerbehebung für AWS Systems Manager-Identität und -Zugriff

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von Ihrer Rolle ab:

- Servicebenutzer – Fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Funktionen zugreifen können (siehe Fehlerbehebung für AWS Systems Manager-Identität und -Zugriff).
- Serviceadministrator – Bestimmen Sie den Benutzerzugriff und stellen Sie Berechtigungsanfragen (siehe Funktionsweise von AWS Systems Manager mit IAM).
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe AWS Systems ManagerBeispiele für identitätsbasierte -Richtlinien).

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM-Benutzer authentifizieren oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich als föderierte Identität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung oder Anmeldeinformationen verwenden. Google/Facebook Weitere Informationen zum Anmelden finden Sie unter So melden Sie sich bei Ihrem AWS-Konto an im Benutzerhandbuch für AWS-Anmeldung .

AWS Bietet für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter AWS Signature Version 4 for API requests im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, dem sogenannten AWS-Konto Root-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Tasks that require root user credentials im IAM-Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Verwenden Sie möglichst temporäre Anmeldeinformationen statt IAM-Benutzer mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Erfordern, dass menschliche Benutzer für den Zugriff AWS mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden müssen.

Eine IAM-Gruppe gibt eine Sammlung von IAM-Benutzern an und vereinfacht die Verwaltung von Berechtigungen bei großer Benutzerzahl. Weitere Informationen finden Sie unter Use cases for IAM users im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität mit bestimmten Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie von einer Benutzer-

zu einer IAM-Rolle (Konsole) wechseln AWS CLI oder einen AWS API-Vorgang aufrufen. Weitere Informationen finden Sie unter Methods to assume a role im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für Verbundbenutzerzugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, dienstübergreifenden Zugriff und Anwendungen, die auf Amazon ausgeführt werden. EC2 Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an Identitäten oder Ressourcen anhängen. AWS Eine Richtlinie definiert Berechtigungen, wenn sie mit einer Identität oder Ressource verknüpft sind. AWS bewertet diese Richtlinien, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter Overview of JSON policies im IAM-Benutzerhandbuch.

Mithilfe von Richtlinien legen Administratoren fest, wer auf was Zugriff hat, indem sie definieren, welcher Prinzipal Aktionen mit welchen Ressourcen und unter welchen Bedingungen ausführen kann.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie Rollen hinzu, die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität (Benutzer, Gruppe oder Rolle) anfügen können. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können Inline-Richtlinien (Richtlinien, die direkt in eine einzelne Identität eingebettet sind) oder verwaltete Richtlinien (eigenständige Richtlinien, die mehreren Identitäten zugeordnet sind) sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter Choose between managed policies and inline policies im IAM-Benutzerhandbuch.

Informationen zu AWS verwalteten Richtlinien für Systems Manager finden Sie unterVon AWS Systems Manager verwaltete Richtlinien.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen und Bucket-Richtlinien von Amazon S3. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Bedingungsschlüssel für die Richtlinie

Die Aktionen, die Benutzer und Rollen ausführen können, und die Ressourcen, auf denen sie diese Aktionen ausführen können, können durch bestimmte Bedingungen weiter eingeschränkt werden.

Das Element `Condition` (oder `Condition`-Blockierung) ermöglicht Ihnen in JSON-Richtliniendokumenten die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. `StringEquals` oder `StringNotLike`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen `AND`-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen Operation aus. `OR` Alle Bedingungen müssen erfüllt sein, bevor die Berechtigungen für die Anweisung erteilt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Weitere Informationen finden Sie unter Globale AWS -Bedingungskontextschlüssel im IAM-Benutzerhandbuch.

> **Important**
>
> Wenn Sie Systems Manager Automation verwenden, empfehlen wir, den SourceIp Bedingungsschlüssel aws: nicht in Ihren Richtlinien zu verwenden. Das Verhalten dieses Bedingungsschlüssels hängt von mehreren Faktoren ab, z. B. davon, ob eine IAM-Rolle für die Ausführung des Automation-Runbooks bereitgestellt wird, und von den im Runbook verwendeten Automatisierungsaktionen. Daher kann der Bedingungsschlüssel zu unerwartetem Verhalten führen. Aus diesem Grund raten wir davon ab, dass Sie es verwenden.

Systems Manager unterstützt eine Reihe von eigenen Bedingungsschlüsseln. Weitere Informationen finden Sie unter Bedingungsschlüssel für AWS Systems Manager in der Service-Autorisierungsreferenz. Die Aktionen und Ressourcen, mit denen Sie einen Systems-Manager-spezifischen Bedingungsschlüssel verwenden können, sind unter Ressourcentypen definiert von AWS Systems Manager in der Service Authorization Reference aufgeführt.

Wenn Ihre Richtlinie von einem Serviceprinzipalnamen abhängen muss, der dem Systems-Manager-Service gehört, empfehlen wir Ihnen, anhand des `aws:PrincipalServiceNamesList` mehrwertigen Bedingungsschlüssels und nicht anhand des `aws:PrincipalServiceName`-Bedingungsschlüssels zu prüfen, ob es existiert oder nicht existiert. Der `aws:PrincipalServiceName`-Bedingungsschlüssel enthält nur einen Eintrag aus der Liste der Serviceprinzipalnamen, und es handelt sich möglicherweise nicht immer um den erwarteten Serviceprinzipalnamen. Der folgende `Condition`-Block zeigt die Überprüfung auf das Vorhandensein von `ssm.amazonaws.com`.

```
{
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:PrincipalServiceNamesList": "ssm.amazonaws.com"
        }
    }
}
```

Beispiele für identitätsbasierte Systems-Manager-Richtlinien finden Sie unter AWS Systems ManagerBeispiele für identitätsbasierte -Richtlinien.

Zugriffskontrolllisten (ACLs)

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Dienste, die Unterstützung ACLs bieten. AWS WAF Weitere Informationen finden Sie unter Übersicht über ACLs die Zugriffskontrollliste (ACL) im Amazon Simple Storage Service Developer Guide.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximalen Berechtigungen festgelegt werden können, die durch gängigere Richtlinientypen gewährt werden:

- Berechtigungsgrenzen – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im -IAM-Benutzerhandbuch.
- Richtlinien zur Dienstkontrolle (SCPs) — Geben Sie die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in an AWS Organizations. Weitere Informationen finden Sie unter Service-Kontrollrichtlinien im AWS Organizations -Benutzerhandbuch.
- Richtlinien zur Ressourcenkontrolle (RCPs) — Legen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten fest. Weitere Informationen finden Sie im AWS Organizations Benutzerhandbuch unter Richtlinien zur Ressourcenkontrolle (RCPs).
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn für eine Anfrage mehrere Arten von Richtlinien gelten, sind die sich daraus ergebenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie unter Bewertungslogik für Richtlinien im IAM-Benutzerhandbuch.

Funktionsweise von AWS Systems Manager mit IAM

Bevor Sie AWS Identity and Access Management (IAM) zur Verwaltung des Zugriffs auf verwenden AWS Systems Manager, sollten Sie wissen, mit welchen IAM-Funktionen Sie verwenden können. Systems Manager Einen allgemeinen Überblick über die Funktionsweise von IAM Systems Manager und andere AWS-Services Funktionen finden Sie AWS-Services im IAM-Benutzerhandbuch.

Themen

- Identitätsbasierte Systems Manager-Richtlinien
- Ressourcenbasierte Systems Manager-Richtlinien
- Autorisierung auf der Basis von Systems Manager-Tags
- Systems ManagerIAM-Rollen

Identitätsbasierte Systems Manager-Richtlinien

Mit identitätsbasierten IAM-Richtlinien können Sie festlegen, welche Aktionen und Ressourcen gewährt oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Systems Manager unterstützt bestimmte Aktionen, Ressourcen und Bedingungsschlüssel. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Aktionen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu erteilen.

Richtlinienaktionen in Systems Manager verwenden das folgende Präfix vor der Aktion: `ssm:`. Um jemandem beispielsweise die Berechtigung zu erteilen, einen Systems Manager-Parameter (SSM-Parameter) mit der Systems Manager `PutParameter` API-Operation zu erstellen, nehmen Sie die Aktion `ssm:PutParameter` in seine Richtlinie auf. Richtlinienanweisungen müssen ein `Action`- oder `NotAction`-Element enthalten. Systems Manager definiert seinen eigenen Satz an Aktionen, die Aufgaben beschreiben, die Sie mit diesem Service durchführen können.

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie wie folgt durch Kommata:

```
"Action": [
      "ssm:action1",
      "ssm:action2"
]
```

> **Note**
>
> Die folgenden Tools AWS Systems Manager verwenden unterschiedliche Präfixe vor Aktionen.
>
> - AWS AppConfig verwendet das Präfix `appconfig:` vor Aktionen.
> - Incident Manager verwendet das Präfix `ssm-incidents:` oder `ssm-contacts:` vor Aktionen.
> - Systems Manager GUI Connect verwendet das Präfix `ssm-guiconnect:` vor Aktionen.
> - Quick Setup verwendet das Präfix `ssm-quicksetup:` vor Aktionen.

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "ssm:Describe*"
```

Eine Liste der Systems Manager-Aktionen finden Sie unter Von AWS Systems Manager definierte Aktionen in der Service-Autorisierungs-Referenz.

Ressourcen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf

Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Die Ressource des Systems Manager-Wartungsfensters hat beispielsweise das folgende ARN-Format.

```
arn:aws:ssm:region:account-id:maintenancewindow/window-id
```

Um die mw-0c50858d01EXAMPLE-Wartungsfenster in Ihrer Anweisung in der Region USA Ost (Ohio) anzugeben, verwenden Sie einen ARN ähnlich dem folgenden.

```
"Resource": "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-0c50858d01EXAMPLE"
```

Um alle Wartungsfenster anzugeben, die zu einem bestimmten Konto gehören, verwenden Sie den Platzhalter (*).

```
"Resource": "arn:aws:ssm:region:123456789012:maintenancewindow/*"
```

Bei `Parameter Store` API-Vorgängen können Sie den Zugriff auf alle Parameter auf einer Hierarchieebene bereitstellen oder einschränken, indem Sie hierarchische Namen und AWS Identity and Access Management (IAM-) Richtlinien wie folgt verwenden.

```
"Resource": "arn:aws:ssm:region:123456789012:parameter/Dev/ERP/Oracle/*"
```

Einige Systems Manager-Aktionen, z. B. zum Erstellen von Ressourcen, können auf bestimmten Ressourcen nicht ausgeführt werden. In diesen Fällen müssen Sie den Platzhalter (*) verwenden.

```
"Resource": "*"
```

Manche Systems Manager-API-Operationen akzeptieren mehrere Ressourcen. Um mehrere Ressourcen in einer einzigen Anweisung anzugeben, trennen Sie sie wie folgt ARNs durch Kommas.

```
"Resource": [
      "resource1",
      "resource2"
```

> **Note**
>
> Die meisten AWS-Services behandeln einen Doppelpunkt (:) oder einen Schrägstrich (/) als dasselbe Zeichen in. ARNs Allerdings erfordert Systems Manager eine exakte Übereinstimmung in den Ressourcenmustern und -regeln. Verwenden Sie also die richtigen ARN-Zeichen zum Erstellen von Ereignismustern, sodass sie mit dem ARN der Ressource übereinstimmen.

Die folgende Tabelle beschreibt die ARN-Formate für die von Systems Manager unterstützten Ressourcentypen.

> **Note**
>
> Beachten Sie die folgenden Ausnahmen für ARN-Formate.
>
> - Die folgenden Tools AWS Systems Manager verwenden unterschiedliche Präfixe vor Aktionen.
>   - AWS AppConfig verwendet das Präfix `appconfig:` vor Aktionen.
>   - Incident Manager verwendet das Präfix `ssm-incidents:` oder `ssm-contacts:` vor Aktionen.
>   - Systems Manager GUI Connect verwendet das Präfix `ssm-guiconnect` vor Aktionen.
> - Dokumente und Automatisierungsdefinitionsressourcen, die Amazon gehören, sowie öffentliche Parameter, die sowohl von Amazon als auch von Drittanbietern bereitgestellt werden, enthalten IDs in ihren ARN-Formaten kein Konto. Zum Beispiel:
>   - Das SSM-Dokument `AWS-RunPatchBaseline`:
>
>     ```
>     arn:aws:ssm:us-east-2::document/AWS-RunPatchBaseline
>     ```
>   - Automation-Runbook `AWS-ConfigureMaintenanceWindows`:
>
>     ```
>     arn:aws:ssm:us-east-2::automation-definition/AWS-
>     ConfigureMaintenanceWindows
>     ```
>   - Öffentliche Parameter `/aws/service/bottlerocket/aws-ecs-1-nvidia/x86_64/1.13.4/image_version`:
>
>     ```
>     arn:aws:ssm:us-east-2::parameter/aws/service/bottlerocket/aws-
>     ecs-1-nvidia/x86_64/1.13.4/image_version
>     ```

Weitere Informationen zu diesen drei Ressourcentypen finden Sie in den folgenden Themen:

- Arbeiten mit Dokumenten
- Führen Sie einen automatisierten Vorgang aus, der von Systems Manager Automation unterstützt wird
- Arbeiten mit öffentlichen Parametern in Parameter Store

- Quick Setup verwendet das Präfix `ssm-quicksetup:` vor Aktionen.

Ressourcentyp	ARN-Format
Anwendung (AWS AppConfig)	arn:aws:appconfig: ::application/ *region account-id application-id*
Zuordnung	arn:aws:ssm: :assoziation/ *region account-id association-id*
Automatisierungsausführung	arn:aws:ssm: *region* ::Automationsausführung/ *account-id automation-execution-id*
Automatisierungsdefinition (mit Versions-Subressource)	arn:aws:ssm: ***region*** ::automationsdefinition/: 1 ***account-id automation-definition-id version-id***
Konfigurationsprofil (AWS AppConfig)	arn:aws:appconfig: *region* ::application/ /configurationprofile/ *account-id application-id configurationprofile-id*
Kontakt (Incident Manager)	arn:aws:ssm-contacts: *region* ::kontakt/ *account-id contact-alias*
Bereitstellungsstrategie (AWS AppConfig)	arn:aws:appconfig: ::deploymentstrategy/ *region account-id deploymentstrategy-id*
Dokument	arn:aws:ssm: :document/ *region account-id document-name*
Umgebung (AWS AppConfig)	arn:aws:appconfig: :application/ *region* /environment/ *account-id application-id environment-id*

Ressourcentyp	ARN-Format
Vorfall	arn:aws:ssm-incidents: *region account-id* ::incident-record/ /*response-plan-name incident-id*
Wartungsfenster	arn:aws:ssm: ::Wartungsfenster/ *region account-id window-id*
Verwalteter Knoten	arn:aws:ssm: *region* ::managed-instance/ *account-id managed-node-id*
Bestand an verwalteten Knoten	arn:aws:ssm::*region*:/*account-id* managed-instance-i nventory*managed-node-id*
OpsItem	arn:aws:ssm *regionaccount-id* : :opsitem/ *OpsItem-id*
Parameter	Ein Parameter mit einer Ebene: • arn:aws:ssm: *region account-id* :parameter//*parameter-name* Ein Parameter, der mit einer hierarchischen Struktur benannt ist: • arn:aws:ssm: ***region*** :parameter////***account-id*** /2 ***parameter-name-root level-2 level-3 level-4 level-5***
Patch-Baseline	arn:aws:ssm: *region* :patchbaseline/ *account-id patch-baseline-id*
Response-Plan	arn:aws:ssm-incidents: *region* ::response-plan/ *account-id response-plan-name*
Sitzung	arn:aws:ssm: ::session/ 3 ***region account-id session-id***
Alle Systems Manager-Ressourcen	arn:aws:ssm:*

Ressourcentyp	ARN-Format
Alle Ressourcen gehören den in der angegebenen Liste angegebenen Systems Manager AWS-Konto AWS-Region	arn:aws:ssm::: * *region account-id*

[1] Für Automatisierungsdefinitionen unterstützt Systems Manager eine Second-Level-Ressource (version-ID). In werden diese AWS Ressourcen der zweiten Ebene als Unterressourcen bezeichnet. Wenn Sie eine Versions-Subressource für eine Automatisierungsdefinition-Ressource angeben, können Sie Zugriff auf bestimmte Versionen einer Automatisierungsdefinition erteilen. So können Sie beispielsweise sicherstellen, dass nur die neueste Version einer Automatisierungsdefinition in der Verwaltung Ihrer Knoten verwendet wird.

[2] Zum Organisieren und Verwalten von Parametern können Sie Namen für Parameter mit hierarchischem Aufbau anlegen. Bei dem hierarchischen Aufbau kann ein Parametername einen Pfad enthalten, den Sie mit Schrägstrichen definieren. Der Name einer Parameterressource darf maximal fünfzehn Ebenen umfassen. Wir empfehlen, dass Sie Hierarchien erstellen, die eine vorhandene hierarchische Struktur in Ihrer Umgebung abbilden. Weitere Informationen finden Sie unter Parameter Store-Parameter im Systems Manager erstellen.

[3] In den meisten Fällen wird die Sitzungs-ID mit der ID des Kontobenutzers erstellt, der die Sitzung gestartet hat, an die ein alphanumerisches Suffix angefügt wird. Zum Beispiel:

```
arn:aws:us-east-2:111122223333:session/JohnDoe-1a2b3c4sEXAMPLE
```

Wenn die Benutzer-ID jedoch nicht verfügbar ist, wird der ARN stattdessen auf diese Weise erstellt:

```
arn:aws:us-east-2:111122223333:session/session-1a2b3c4sEXAMPLE
```

Weitere Informationen zum Format von ARNs finden Sie unter Amazon Resource Names (ARNs) in der Allgemeine Amazon Web Services-Referenz.

Eine Liste der Systems Manager Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter Resources Defined by AWS Systems Manager in der Service Authorization Reference. Informationen

zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter Von AWS Systems Manager definierte Aktionen.

Bedingungsschlüssel für Systems Manager

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das `Condition`-Element legt fest, ob Anweisungen auf der Grundlage definierter Kriterien ausgeführt werden. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Eine Liste von Systems Manager-Bedingungsschlüsseln finden Sie unter Bedingungsschlüssel für AWS Systems Manager in der Service-Autorisierungs-Referenz. Informationen dazu, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von AWS Systems Manager definierte Aktionen.

Weitere Informationen zum Verwenden des `ssm:resourceTag/*`-Bedingungsschlüssels finden Sie in den folgenden Themen:

- Einschränken des Zugriffs auf Befehle auf Stammebene durch SSM Agent
- Den Zugriff von Run Command anhand von Tags beschränken
- Beschränkung des Sitzungszugriffs auf Instance-Tags

Weitere Informationen zum Verwenden der Bedingungsschlüssels `ssm:Recursive`, `ssm:Policies` und `ssm:Overwrite` finden Sie unter Zugriff auf Parameter Store-API-Vorgänge verhindern.

Beispiele

Beispiele für identitätsbasierte Systems Manager-Richtlinien finden Sie unter AWS Systems ManagerBeispiele für identitätsbasierte -Richtlinien.

Ressourcenbasierte Systems Manager-Richtlinien

Andere AWS-Services, wie Amazon Simple Storage Service (Amazon S3), unterstützen ressourcenbasierte Berechtigungsrichtlinien. Beispielsweise können Sie einem S3-Bucket eine Berechtigungsrichtlinie zuweisen, um die Zugriffsberechtigungen für diesen Bucket zu verwalten.

Systems Manager unterstützt keine ressourcenbasierten Richtlinien.

Autorisierung auf der Basis von Systems Manager-Tags

Sie können Tags an Systems Manager-Ressourcen anfügen oder Tags in einer Anforderung an Systems Manager übergeben. Um den Zugriff basierend auf Tags zu steuern, stellen Sie Tag-Informationen im Bedingungselement einer Richtlinie unter Verwendung der Bedingungsschlüssel `ssm:resourceTag/key-name`, `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` oder `aws:TagKeys` zur Verfügung. Sie können den folgenden Ressourcentypen beim Erstellen oder Aktualisieren Tags hinzufügen:

- Dokument
- Verwalteter Knoten
- Wartungsfenster
- Parameter
- Patch-Baseline
- OpsItem

Ein Beispiel für eine identitätsbasierte Richtlinie zur Einschränkung des Zugriffs auf eine Ressource auf der Grundlage der Markierungen dieser Ressource finden Sie unter Anzeigen von Systems Manager-Dokumenten basierend auf Tags.

Systems ManagerIAM-Rollen

Eine IAM-Rolle ist eine Entität innerhalb Ihres Unternehmens AWS-Konto , die über bestimmte Berechtigungen verfügt.

Verwenden temporärer Anmeldeinformationen mit Systems Manager

Sie können temporäre Anmeldeinformationen verwenden, um sich über einen Verbund anzumelden, eine IAM-Rolle anzunehmen oder eine kontenübergreifende Rolle anzunehmen. Sie erhalten temporäre Sicherheitsanmeldedaten, indem Sie API-Operationen AWS Security Token Service (AWS STS) wie AssumeRoleoder GetFederationTokenaufrufen.

Systems Manager unterstützt die Verwendung temporärer Anmeldeinformationen.

Service-verknüpfte Rollen

Mit Diensten verknüpfte Rollen ermöglichen AWS-Services den Zugriff auf Ressourcen in anderen Diensten, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem IAM-Konto aufgelistet und gehören zum Service. Ein -Administrator kann die Berechtigungen für serviceverknüpfte Rollen anzeigen, aber nicht bearbeiten.

Systems Manager unterstützt serviceverknüpfte Rollen. Details zum Erstellen oder Verwalten von serviceverknüpften Systems Manager-Rollen finden Sie unter Verwenden von serviceverknüpften Rollen für Systems Manager.

Servicerollen

Dieses Feature ermöglicht einem Service das Annehmen einer Servicerolle in Ihrem Namen. Diese Rolle gewährt dem Service Zugriff auf Ressourcen in anderen Diensten, um eine Aktion in Ihrem Namen auszuführen. Servicerollen werden in Ihrem IAM-Konto angezeigt und gehören zum Konto. Dies bedeutet, dass ein -Administrator die Berechtigungen für diese Rolle ändern kann. Dies kann jedoch die Funktion des Services beeinträchtigen.

Systems Manager unterstützt Servicerollen.

Auswählen einer IAM-Rolle in Systems Manager

Damit Systems Manager mit Ihren verwalteten Knoten interagieren kann, müssen Sie eine Rolle wählen, die Systems Manager den Zugriff auf Knoten in Ihrem Namen erlaubt. Wenn Sie zuvor eine Servicerolle oder serviceverknüpfte Rolle erstellt haben, stellt Ihnen Systems Manager eine Liste mit Rollen bereit, aus denen Sie wählen können. Es ist wichtig, eine Rolle zu wählen, die den Zugriff auf das Starten und Stoppen von verwalteten Knoten erlaubt.

Um auf EC2 Instanzen zugreifen zu können, müssen Sie Instanzberechtigungen konfigurieren. Informationen finden Sie unter Konfiguration von erforderliche Instance-Berechtigungen für Systems Manager.

Für den Zugriff auf EC2 Nicht-Knoten in einer Hybrid- und Multicloud AWS-Konto benötigen Sie eine IAM-Servicerolle. Weitere Informationen finden Sie unter Erstellen der für Systems Manager in Hybrid- und Multi-Cloud-Umgebungen erforderlichen IAM-Servicerolle.

Ein Automation-Workflow kann im Kontext einer Service-Rolle initiiert werden (oder eine Rolle übernehmen). Auf diese Weise kann der Service Aktionen in Ihrem Namen ausführen. Wenn

Sie keine Übernahmerolle angeben, verwendet Automation den Kontext des Benutzers, der die Ausführung aufgerufen hat. In bestimmten Situationen ist es jedoch erforderlich, dass Sie eine Service-Rolle für Automation angeben. Weitere Informationen finden Sie unter Konfigurieren eines Service-Rollenzugriffs (Rolle übernehmen) für Automatisierungen.

Von AWS Systems Manager verwaltete Richtlinien

AWS adressiert viele gängige Anwendungsfälle durch die Bereitstellung eigenständiger IAM-Richtlinien, die von erstellt und verwaltet AWS werden. Diese von AWS verwalteten Richtlinien erteilen die erforderlichen Berechtigungen für häufige Anwendungsfälle, sodass Sie nicht mühsam ermitteln müssen, welche Berechtigungen erforderlich sind. (Sie können auch Ihre eigenen, benutzerdefinierten IAM-Richtlinien erstellen, um Berechtigungen für Systems Manager-Aktionen und -Ressourcen zu gewähren.)

Weitere Informationen über -verwaltete Richtlinien für Systems Manager finden Sie unter AWS verwaltete Richtlinien für AWS Systems Manager

Allgemeine Informationen über verwaltete Richtlinien finden Sie unter AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS Systems ManagerBeispiele für identitätsbasierte -Richtlinien

Standardmäßig sind AWS Identity and Access Management (IAM-) Entitäten (Benutzer und Rollen) nicht berechtigt, Ressourcen zu erstellen oder zu ändern AWS Systems Manager . Sie können auch keine Aufgaben mit der Systems Manager Manager-Konsole AWS Command Line Interface (AWS CLI) oder der AWS API ausführen. Ein Administrator muss IAM-Richtlinien erstellen, die Benutzern und Rollen die Berechtigung zum Ausführen bestimmter API-Operationen für die angegebenen Ressourcen gewähren, die diese benötigen. Der Administrator muss diese Richtlinien anschließend den -Benutzern oder -Gruppen anfügen, die diese Berechtigungen benötigen.

Im Folgenden finden Sie ein Beispiel für eine Berechtigungsrichtlinie, die es einem Benutzer ermöglicht, Dokumente zu löschen, deren Namen mit **`MyDocument-`** „USA Ost (Ohio)“ (us-east-2) beginnen. AWS-Region

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
```

```
      "Effect" : "Allow",
      "Action" : [
        "ssm:DeleteDocument"
      ],
      "Resource" : [
        "arn:aws:ssm:us-east-1:111122223333:document/MyDocument-*"
      ]
    }
  ]
}
```

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Themen

- Best Practices für Richtlinien
- Beispiel: Erlaubnis zur Nutzung der Systems Manager-Konsole
- Beispiel: Berechtigung dafür, Benutzern die Anzeige der eigenen Berechtigungen zu erlauben
- Beispiel: Berechtigung zum Lesen und Beschreiben von einzelnen Parametern
- Serviceübergreifende Confused-Deputy-Prävention
- Beispiele für vom Kunden verwaltete Richtlinien
- Anzeigen von Systems Manager-Dokumenten basierend auf Tags

Best Practices für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand Systems Manager-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Beachten Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Richtlinien und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen

finden Sie unter Von AWS verwaltete Richtlinien oder Von AWS verwaltete Richtlinien für Auftragsfunktionen im IAM-Benutzerhandbuch.

- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung mit IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Best Practices für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Beispiel: Erlaubnis zur Nutzung der Systems Manager-Konsole

Um auf die Systems Manager-Konsole zuzugreifen, müssen Sie über einen Mindestsatz von Berechtigungen verfügen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details über

die Systems Manager-Ressourcen und andere Ressourcen in Ihrem AWS-Konto aufzulisten und anzuzeigen.

Wenn Sie eine identitätsbasierte Richtlinie erstellen, die restriktiver ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole für IAM-Entitäten (Benutzer oder Rollen) mit dieser Richtlinie nicht wie vorgesehen.

Sie müssen Benutzern, die nur die API oder die API aufrufen, keine Mindestberechtigungen für die AWS CLI Konsole gewähren. AWS Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die Sie ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die Systems Manager Manager-Konsole weiterhin verwenden können, fügen Sie den Entitäten auch die von Amazon SSMFull Access oder von Amazon SSMRead OnlyAccess AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

Beispiel: Berechtigung dafür, Benutzern die Anzeige der eigenen Berechtigungen zu erlauben

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API AWS CLI oder AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
```

```
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Beispiel: Berechtigung zum Lesen und Beschreiben von einzelnen Parametern

Example Lesen und Beschreiben eines Parameters

Sie können Zugriff auf einen Parameter gewähren, indem Sie die folgende Richtlinie an eine Identität anfügen.

JSON

```
{
"Version":"2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ssm:GetParameter",
      "ssm:DescribeParameters"
      ],
    "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/parameter-name"
  }
]
}
```

Serviceübergreifende Confused-Deputy-Prävention

Das Problem des verwirrten Stellvertreters ist ein Sicherheitsproblem, bei dem eine Entität, die keine Berechtigung zur Durchführung einer Aktion hat, eine privilegiertere Entität zur Durchführung der Aktion zwingen kann. In AWS, dienstübergreifender Identitätswechsel kann zum Problem des verwirrten Stellvertreters führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der Anruf-Dienst kann so manipuliert werden, dass er seine Berechtigungen verwendet, um auf die Ressourcen eines anderen Kunden zu reagieren, auf die er sonst nicht zugreifen dürfte. Um dies zu verhindern, AWS bietet Tools, mit denen Sie Ihre Daten für alle Dienste mit Dienstprinzipalen schützen können, denen Zugriff auf Ressourcen in Ihrem Konto gewährt wurde.

Wir empfehlen die Verwendung der globalen Bedingungskontext-Schlüssel `aws:SourceArn` und `aws:SourceAccount` in ressourcenbasierten Richtlinien, um die Berechtigungen, die AWS Systems Manager einem anderen Service erteilt, auf eine bestimmte Ressource einzuschränken. Wenn der `aws:SourceArn`-Wert nicht die Konto-ID enthält, z. B. den Amazon-Ressourcenname (ARN) eines S3-Buckets, müssen Sie beide globale Bedingungskontext-Schlüssel verwenden, um Berechtigungen einzuschränken. Wenn Sie beide globale Bedingungskontextschlüssel verwenden und der `aws:SourceArn`-Wert die Konto-ID enthält, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in der gleichen Richtlinienanweisung verwendet wird. Verwenden Sie `aws:SourceArn`, wenn Sie nur eine Ressource mit dem betriebsübergreifenden Zugriff verknüpfen möchten. Verwenden Sie `aws:SourceAccount`, wenn Sie zulassen möchten, dass Ressourcen in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft werden.

In den folgenden Abschnitten finden Sie Beispielrichtlinien für AWS Systems Manager-Tools.

Beispiel für hybride Aktivierungsrichtlinien

Bei Servicerollen, die bei einer Hybrid-Aktivierung verwendet werden, muss der Wert von `aws:SourceArn` der ARN des AWS-Konto sein. Stellen Sie sicher, dass Sie AWS-Region im ARN angeben, in dem Sie Ihre Hybrid-Aktivierung erstellt haben. Wenn Sie den vollständigen ARN der Ressource nicht kennen oder wenn Sie mehrere Ressourcen angeben, verwenden Sie den globalen Bedingungskontext-Schlüssel `aws:SourceArn` mit Platzhaltern (`*`) für die unbekannten Teile des ARN. Beispiel, `arn:aws:ssm:*:region:123456789012:*`.

Das folgende Beispiel veranschaulicht die Verwendung der globalen Bedingungskontext-Schlüssel `aws:SourceArn` und `aws:SourceAccount` für Automatisierung, um das Confused-Deputy-Problem in der Region USA Ost (Ohio) (us-east-2) zu verhindern.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"",
         "Effect":"Allow",
         "Principal":{
            "Service":"ssm.amazonaws.com"
         },
         "Action":"sts:AssumeRole",
         "Condition":{
            "StringEquals":{
               "aws:SourceAccount":"123456789012"
            },
            "ArnEquals":{
               "aws:SourceArn":"arn:aws:ssm:us-east-1:123456789012:*"
            }
         }
      }
   ]
}
```

Beispiel-Richtlinie für Ressourcen-Datensynchronisierung

Systems Manager Inventory und Compliance ermöglichen es IhnenExplorer, eine Ressourcendatensynchronisierung zu erstellen, um die Speicherung Ihrer Betriebsdaten (OpsData) in einem zentralen Amazon Simple Storage Service-Bucket zu zentralisieren. Wenn Sie eine Ressourcendatensynchronisierung mit AWS Key Management Service (AWS KMS) verschlüsseln möchten, müssen Sie entweder einen neuen Schlüssel erstellen, der die folgende Richtlinie enthält, oder Sie müssen einen vorhandenen Schlüssel aktualisieren und diese Richtlinie hinzufügen. Die `aws:SourceArn` und `aws:SourceAccount`-Bedingungsschlüssel in dieser Richtlinie verhindern das Confused-Deputy-Problem. Hier ist eine Beispielrichtlinie.

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Id": "ssm-access-policy",
    "Statement": [
        {
            "Sid": "ssm-access-policy-statement",
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Resource": "arn:aws:kms:us-east-1:123456789012:key/KMS_key_id",
            "Condition": {
                "StringLike": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:ssm:*:123456789012:role/aws-
service-role/ssm.amazonaws.com/AWSServiceRoleForAmazonSSM"
                }
            }
        }
    ]
}
```

> **Note**
>
> Der ARN im Richtlinienbeispiel ermöglicht es dem System, OpsData aus allen Quellen außer AWS Security Hub CSPM zu verschlüsseln. Wenn Sie Security Hub-CSPM-Daten verschlüsseln müssen, z. B. wenn Sie sie Explorer zum Sammeln von Security Hub-CSPM-Daten verwenden, müssen Sie eine zusätzliche Richtlinie anhängen, die den folgenden ARN festlegt:
> `"aws:SourceArn": "arn:aws:ssm:*:account-id:role/aws-service-role/opsdatasync.ssm.amazonaws.com/AWSServiceRoleForSystemsManagerOpsDataSync"`

Beispiele für vom Kunden verwaltete Richtlinien

Sie können eigenständige Richtlinien erstellen, die Sie in Ihrem eigenen AWS-Konto verwalten. Wir bezeichnen diese als vom Kunden verwaltete Richtlinien. Sie können diese Richtlinien mehreren Hauptentitäten in Ihrem zuordnen. AWS-Konto Wenn Sie eine Richtlinie an eine Auftraggeber-Entität anfügen, gewähren Sie ihr die in der Richtlinie festgelegten Berechtigungen. Weitere Informationen finden Sie unter Beispiele für kundenverwaltete Richtlinien im IAM-Benutzerhandbuch.

Die folgenden Beispiele für Benutzerrichtlinien gewähren Berechtigungen für verschiedene Aktionen von Systems Manager. Verwenden Sie diese, um den Systems Manager-Zugriff für Ihre IAM-Entitäten (Benutzer und Rollen) einzuschränken. Diese Richtlinien funktionieren, wenn Aktionen in der Systems Manager API AWS SDKs, oder der ausgeführt AWS CLI werden. Für Benutzer, die die Konsole verwenden, müssen Sie zusätzliche konsolenspezifische Berechtigungen erteilen. Weitere Informationen finden Sie unter Beispiel: Erlaubnis zur Nutzung der Systems Manager-Konsole.

> **Note**
>
> Alle Beispiele verwenden die Region USA West (Oregon) (us-west-2) und enthalten ein fiktives Konto. IDs Die Konto-ID sollte nicht im Amazon-Ressourcennamen (ARN) für AWS öffentliche Dokumente (Dokumente, die mit beginnen`AWS-*`) angegeben werden.

Beispiele

- Beispiel 1: Zulassen, dass ein Benutzer Systems Manager-Operationen in einer einzelnen Region ausführt
- Beispiel 2: Zulassen, dass ein Benutzer Dokumente für eine einzelne Region auflistet

Beispiel 1: Zulassen, dass ein Benutzer Systems Manager-Operationen in einer einzelnen Region ausführt

Im folgenden Beispiel werden die Berechtigungen zur Ausführung von Systems Manager-Operationen nur in der Region USA Ost (Ohio) (us-east-2) gewährt.

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:*"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:*"
            ]
        }
    ]
}
```

Beispiel 2: Zulassen, dass ein Benutzer Dokumente für eine einzelne Region auflistet

Das folgende Beispiel erteilt Berechtigungen zum Auflisten aller Dokumentennamen, die mit **`Update`** in der Region USA Ost (Ohio) (us-east-2) beginnen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:ListDocuments"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/Update*"
            ]
        }
    ]
}
```

Beispiel 3: Erlauben, dass ein Benutzer ein spezifisches SSM-Dokument zum Ausführen von Befehlen auf bestimmten Knoten verwendet

Die folgende IAM-Beispielrichtlinie ermöglicht es einem Benutzer, in der Region USA Ost (Ohio) (us-east-2) die folgenden Aktionen auszuführen:

- Listen Sie Systems Manager-Dokumente (SSM-Dokumente) und Dokumentversionen auf.
- Zeigen Sie Details zu Dokumenten an.
- Senden Sie einen Befehl mit dem in der Richtlinie angegebenen Dokument. Der Name des Dokuments wird durch den folgenden Eintrag bestimmt.

```
arn:aws:ssm:us-east-2:aws-account-ID:document/Systems-Manager-document-name
```

- Senden Sie einen Befehl an drei Knoten. Die Knoten werden anhand der folgenden Einträge im zweiten `Resource`-Abschnitt bestimmt.

```
"arn:aws:ec2:us-east-2:aws-account-ID:instance/i-02573cafcfEXAMPLE",
"arn:aws:ec2:us-east-2:aws-account-ID:instance/i-0471e04240EXAMPLE",
"arn:aws:ec2:us-east-2:aws-account-ID:instance/i-07782c72faEXAMPLE"
```

- Zeigen Sie Details zu einem Befehl an, nachdem er gesendet wurde.
- Starten und Beenden von Workflows in Automation, einem Tool von AWS Systems Manager.
- Informationen zum Automation-Workflows

Wenn Sie einem Benutzer die Berechtigung gewähren möchten, dieses Dokument zu verwenden, um Befehle an jeden Knoten zu senden, auf den der Benutzer Zugriff hat, können Sie einen Eintrag ähnlich dem folgenden im `Resource`-Abschnitt angeben und die anderen Knoteneinträge entfernen. Im folgenden Beispiel wird die Region USA Ost (Ohio) (us-east-2) verwendet.

```
"arn:aws:ec2:us-east-2:*:instance/*"
```

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "ssm:ListDocuments",
                "ssm:ListDocumentVersions",
                "ssm:DescribeDocument",
                "ssm:GetDocument",
                "ssm:DescribeInstanceInformation",
                "ssm:DescribeDocumentParameters",
```

```
                "ssm:DescribeInstanceProperties"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "ssm:SendCommand",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ec2:us-east-1:111122223333:instance/
i-02573cafcfEXAMPLE",
                "arn:aws:ec2:us-east-1:111122223333:instance/
i-0471e04240EXAMPLE",
                "arn:aws:ec2:us-east-1:111122223333:instance/
i-07782c72faEXAMPLE",

                "arn:aws:ssm:us-east-1:111122223333:document/Systems-Manager-
document-name"
            ]
        },
        {
            "Action": [
                "ssm:CancelCommand",
                "ssm:ListCommands",
                "ssm:ListCommandInvocations"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "ec2:DescribeInstanceStatus",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "ssm:StartAutomationExecution",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:document/*",
                "arn:aws:ssm:us-east-1:111122223333:automation-execution/*"
            ]
        },
        {
            "Action": "ssm:DescribeAutomationExecutions",
```

```
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Action": [
                "ssm:StopAutomationExecution",
                "ssm:GetAutomationExecution"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Anzeigen von Systems Manager-Dokumenten basierend auf Tags

Sie können in Ihrer identitätsbasierten Richtlinie Bedingungen für die Steuerung des Zugriffs auf Systems Manager-Ressourcen auf der Basis von Tags verwenden. Dieses Beispiel zeigt, wie Sie eine Richtlinie erstellen können, die die Anzeige eines SSM-Dokuments ermöglicht. Die Berechtigung wird jedoch nur gewährt, wenn das Dokument-Tag `Owner` den Wert des Benutzernamens dieses Benutzers hat. Diese Richtlinie gewährt auch die Berechtigungen, die für die Ausführung dieser Aktion auf der Konsole erforderlich sind.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ListDocumentsInConsole",
            "Effect": "Allow",
            "Action": "ssm:ListDocuments",
            "Resource": "*"
        },
        {
            "Sid": "ViewDocumentIfOwner",
```

```
            "Effect": "Allow",
            "Action": "ssm:GetDocument",
            "Resource": "arn:aws:ssm:*:*:document/*",
            "Condition": {
                "StringEquals": {"ssm:ResourceTag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

Sie können diese Richtlinie den -Benutzern in Ihrem Konto zuweisen. Wenn ein Benutzer mit dem Namen `richard-roe` versucht, ein Dokument mit dem Namen Systems Manager anzuzeigen, muss das Dokument mit dem Tag `0wner=richard-roe` oder `owner=richard-roe` gekennzeichnet werden. Andernfalls wird diesen der Zugriff verweigert. Der Tag-Schlüssel `0wner` der Bedingung stimmt sowohl mit `0wner` als auch mit `owner` überein, da die Namen von Bedingungsschlüsseln nicht zwischen Groß- und Kleinschreibung unterscheiden. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinien für AWS Systems Manager

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS Verwaltete Richtlinien dienen dazu, Berechtigungen für viele gängige Anwendungsfälle bereitzustellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

Themen

- AWS verwaltete Richtlinie: Amazon SSMService RolePolicy
- AWS verwaltete Richtlinie: Amazon SSMAutomation Role
- AWS verwaltete Richtlinie: Amazon SSMRead OnlyAccess
- AWS verwaltete Richtlinie: AWSSystems ManagerOpsDataSyncServiceRolePolicy
- AWS verwaltete Richtlinie: Amazon SSMManaged EC2 InstanceDefaultPolicy
- AWS verwaltete Richtlinie: SSMQuick SetupRolePolicy
- AWS verwaltete Richtlinie: AWSQuick SetupDeploymentRolePolicy
- AWS verwaltete Richtlinie: AWSQuick SetupPatchPolicyDeploymentRolePolicy
- AWS verwaltete Richtlinie: AWSQuick SetupPatchPolicyBaselineAccess
- AWS verwaltete Richtlinie: AWSSystemsManagerEnableExplorerExecutionPolicy
- AWS verwaltete Richtlinie: AWSSystemsManagerEnableConfigRecordingExecutionPolicy
- AWS verwaltete Richtlinie: AWSQuick SetupDevOpsGuruPermissionsBoundary
- AWS verwaltete Richtlinie: AWSQuick SetupDistributorPermissionsBoundary
- AWS verwaltete Richtlinie: Einrichtung AWSQuick SSMHost MgmtPermissionsBoundary
- AWS verwaltete Richtlinie: AWSQuick SetupPatchPolicyPermissionsBoundary
- AWS verwaltete Richtlinie: AWSQuick SetupSchedulerPermissionsBoundary
- AWS verwaltete Richtlinie: Einrichtung AWSQuick CFGCPacks PermissionsBoundary
- AWS verwaltete Richtlinie: AWSQuick SetupStartStopInstancesExecutionPolicy
- AWS verwaltete Richtlinie: AWSQuick SetupStart SSMAssociations ExecutionPolicy
- AWS verwaltete Richtlinie: AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy
- AWS verwaltete Richtlinie: AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy
- AWS verwaltete Richtlinie: AWS-SSM-RemediationAutomation-AdministrationRolePolicy
- AWS verwaltete Richtlinie: AWS-SSM-RemediationAutomation-ExecutionRolePolicy
- AWS verwaltete Richtlinie: Einrichtung AWSQuick SSMManage ResourcesExecutionPolicy
- AWS verwaltete Richtlinie: AWSQuick Einrichtung SSMLifecycle ManagementExecutionPolicy
- AWS verwaltete Richtlinie: AWSQuick Einrichtung SSMDeployment RolePolicy
- AWS verwaltete Richtlinie: SSMDeployment S3 AWSQuick einrichten BucketRolePolicy
- AWS Von verwaltete Richtlinie: AWSQuickSetupEnableDHMCExecutionPolicy

- AWS Von verwaltete Richtlinie: AWSQuickSetupEnableAREXExecutionPolicy
- AWS verwaltete Richtlinie: AWSQuick SetupManagedInstanceProfileExecutionPolicy
- AWS verwaltete Richtlinie: AWSQuick SetupManage JITNAResources ExecutionPolicy
- AWS verwaltete Richtlinie: Einrichtung AWSQuick JITNADeployment RolePolicy
- AWS verwaltete Richtlinie: AWSSystems ManagerJustInTimeAccessServicePolicy
- AWS verwaltete Richtlinie: AWSSystems ManagerJustInTimeAccessTokenPolicy
- AWS verwaltete Richtlinie: AWSSystems ManagerJustInTimeAccessTokenSessionPolicy
- AWS verwaltete Richtlinie: AWSSystems ManagerJustInTimeNodeAccessRolePropagationPolicy
- AWS verwaltete Richtlinie: AWSSystems ManagerNotificationsServicePolicy
- AWS verwaltete Richtlinie: AWS-SSM-Automation-DiagnosisBucketPolicy
- AWS verwaltete Richtlinie: AWS-SSM-RemediationAutomation-OperationalAccountAdministrationRolePolicy
- AWS verwaltete Richtlinie: AWS-SSM-DiagnosisAutomation-OperationalAccountAdministrationRolePolicy
- Systems ManagerAktualisierungen der verwalteten Richtlinien AWS
- Zusätzliche verwaltete Richtlinien für Systems Manager

AWS verwaltete Richtlinie: Amazon SSMService RolePolicy

Diese Richtlinie ermöglicht den Zugriff auf eine Reihe von AWS Ressourcen, die von Systems Manager Manager-Vorgängen verwaltet AWS Systems Manager oder in diesen verwendet werden.

Sie können keine Verbindungen `AmazonSSMServiceRolePolicy` zu Ihren AWS Identity and Access Management (IAM-) Entitäten herstellen. Diese Richtlinie ist mit einer dienstbezogenen Rolle verknüpft, die es AWS Systems Manager ermöglicht, Aktionen in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter Verwenden von Rollen zum Sammeln und Anzeigen von Inventar OpsData.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm` – Ermöglicht Prinzipalen, Ausführungen sowohl für Run Command als auch für Automation zu starten und auszuführen sowie Informationen über Run Command und Automation-Vorgänge abzurufen, Informationen über Parameter Store-Parameter-Change Calendar-Kalender abzurufen, Informationen zu Systems-Manager-Serviceteinstellungen für OpsCenter-Ressourcen

zu aktualisieren und abzurufen und Informationen über Tags zu lesen, die auf Ressourcen angewendet wurden.

- `cloudformation` – Ermöglicht Prinzipalen das Abrufen von Informationen über Stackset-Operationen und Stackset-Instances sowie das Löschen von Stacksets in der `arn:aws:cloudformation:*:*:stackset/AWS-QuickSetup-SSM*:*`-Ressource. Ermöglicht Prinzipalen das Löschen von Stack-Instances, die den folgenden Ressourcen zugeordnet sind:

```
arn:aws:cloudformation:*:*:stackset/AWS-QuickSetup-SSM*:*
arn:aws:cloudformation:*:*:stackset-target/AWS-QuickSetup-SSM*:*
arn:aws:cloudformation:*:*:type/resource/*
```

- `cloudwatch`— Ermöglicht Prinzipalen das Abrufen von Informationen zu CloudWatch Amazon-Alarmen.
- `compute-optimizer`— Ermöglicht Principals, den Anmeldestatus (Opt-In) eines Kontos für den AWS Compute Optimizer Service abzurufen und Empfehlungen für EC2 Amazon-Instances abzurufen, die bestimmte festgelegte Anforderungen erfüllen.
- `config`— Ermöglicht Prinzipalen das Abrufen von Informationen zur Behebung von Konfigurationen und Konfigurationsrekordern sowie die Feststellung AWS Config, ob die angegebenen AWS Config Regeln und Ressourcen den Anforderungen entsprechen. AWS
- `events`— Ermöglicht Prinzipalen das Abrufen von Informationen über EventBridge Regeln, das Erstellen von EventBridge Regeln und Zielen ausschließlich für den Systems Manager Manager-Dienst (`ssm.amazonaws.com`) und das Löschen von Regeln und Zielen für die Ressource`arn:aws:events:*:*:rule/SSMExplorerManagedRule`.
- `ec2`— Ermöglicht Principals das Abrufen von Informationen über EC2 Amazon-Instances.
- `iam` – Ermöglicht Prinzipalen die Weitergabe von Rollenberechtigungen für den Systems-Manager-Service (`ssm.amazonaws.com`).
- `lambda` – Ermöglicht Prinzipalen das Aufrufen von Lambda-Funktionen, die speziell für die Verwendung durch Systems Manager konfiguriert wurden.
- `resource-explorer-2`— Ermöglicht Prinzipalen das Abrufen von Daten über EC2 Instanzen, um festzustellen, ob jede Instanz derzeit von Systems Manager verwaltet wird oder nicht.

  Die Aktion `resource-explorer-2:CreateManagedView` ist für die `arn:aws:resource-explorer-2:*:*:managed-view/AWSManagedViewForSSM*`-Ressource zulässig.
- `resource-groups`— Ermöglicht Prinzipalen das Abrufen von Listenressourcengruppen und ihren Mitgliedern AWS -Ressourcengruppen aus Ressourcen, die zu einer Ressourcengruppe gehören.

- `securityhub`— Ermöglicht Prinzipalen das Abrufen von Informationen über AWS Security Hub CSPM Hub-Ressourcen im aktuellen Konto.
- `states`— Ermöglicht Prinzipalen das Starten und Abrufen von Informationen AWS Step Functions , die speziell für die Verwendung durch Systems Manager konfiguriert wurden.
- `support` – Ermöglicht Prinzipalen das Abrufen von Informationen über Prüfungen und Fälle in AWS Trusted Advisor.
- `tag` – Ermöglicht Prinzipalen das Abrufen von Informationen über alle markierten oder zuvor markierten Ressourcen, die sich in einer angegebenen AWS-Region für ein Konto befinden.

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter Amazon SSMService RolePolicy im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: Amazon SSMAutomation Role

Sie können die `AmazonSSMAutomationRole`-Richtlinie an Ihre IAM-Identitäten anfügen. Diese Richtlinie gewährt dem AWS Systems Manager Automation-Service Berechtigungen zur Ausführung von Aktivitäten, die in Automation-Runbooks definiert sind.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `lambda` – Ermöglicht Prinzipalen das Aufrufen von Lambda-Funktionen, deren Namen mit „Automation“ beginnen. Das ist erforderlich, damit Automation-Runbooks Lambda-Funktionen als Teil ihres Workflows ausführen können.
- `ec2`— Ermöglicht Principals, verschiedene EC2 Amazon-Operationen durchzuführen, darunter das Erstellen, Kopieren und Abmelden von Images, das Verwalten von Snapshots, das Starten, Ausführen, Stoppen und Beenden von Instances, das Verwalten des Instance-Status sowie das Erstellen, Löschen und Beschreiben von Tags. Diese Berechtigungen ermöglichen es Automation-Runbooks, EC2 Amazon-Ressourcen während der Ausführung zu verwalten.
- `cloudformation`— Ermöglicht Prinzipalen das Erstellen, Beschreiben, Aktualisieren und Löschen CloudFormation von Stacks. Auf diese Weise können Automation-Runbooks die Infrastruktur als Code verwalten. CloudFormation
- `ssm` – Ermöglicht Prinzipalen die Verwendung aller Systems-Manager-Aktionen. Dieser umfassende Zugriff ist erforderlich, damit Automation-Runbooks mit allen Systems-Manager-Funktionen interagieren können.

- `sns` – Ermöglicht Prinzipalen, Nachrichten zu Amazon-SNS-Themen zu veröffentlichen, deren Namen mit „Automation“ beginnen. Dadurch können Automation-Runbooks während der Ausführung Benachrichtigungen senden.
- `ssmmessages` – Ermöglicht Prinzipalen das Öffnen von Datenkanälen für Systems-Manager-Sitzungen. Auf diese Weise können Automation-Runbooks Kommunikationskanäle für sitzungsbasierte Vorgänge einrichten.

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter Amazon SSMAutomation Role im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: Amazon SSMRead OnlyAccess

Sie können die `AmazonSSMReadOnlyAccess`-Richtlinie an Ihre IAM-Identitäten anfügen. Diese Richtlinie gewährt schreibgeschützten Zugriff auf AWS Systems Manager API-Operationen`Describe*`, einschließlich`Get*`, und. `List*`

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter Amazon SSMRead OnlyAccess im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: AWSSystems ManagerOpsDataSyncServiceRolePolicy

Sie können `AWSSystemsManagerOpsDataSyncServiceRolePolicy` nicht an Ihre IAM-Entitäten anhängen. Diese Richtlinie ist an eine servicegebundene Rolle angehängt, die Systems Manager die Durchführung von Aktionen in Ihrem Namen ermöglicht. Weitere Informationen finden Sie unter Verwenden von Rollen zum Erstellen OpsData und OpsItems für Explorer.

`AWSSystemsManagerOpsDataSyncServiceRolePolicy`ermöglicht der `AWSServiceRoleForSystemsManagerOpsDataSync` serviceverknüpften Rolle das Erstellen und Aktualisieren von Ergebnissen OpsItems sowie das Verwenden OpsData von AWS Security Hub CSPM Ergebnissen.

Die Richtlinie erlaubt es Systems Manager, die folgenden Aktionen für alle damit verbundenen Ressourcen (`"Resource": "*"`) auszuführen, außer wenn dies angegeben ist:

- `ssm:GetOpsItem` [1]
- `ssm:UpdateOpsItem` [1]
- `ssm:CreateOpsItem`

- ssm:AddTagsToResource [2]
- ssm:UpdateServiceSetting [3]
- ssm:GetServiceSetting [3]
- securityhub:GetFindings
- securityhub:GetFindings
- securityhub:BatchUpdateFindings [4]

[1] Die Aktionen ssm:GetOpsItem und ssm:UpdateOpsItem sind nur mit der folgenden Bedingung für den Systems Manager-Service zulässig.

```
"Condition": {
    "StringEquals": {
        "aws:ResourceTag/ExplorerSecurityHubOpsItem": "true"
    }
}
```

[2] Die Aktion ssm:AddTagsToResource ist nur für die folgende Ressource zulässig.

```
arn:aws:ssm:*:*:opsitem/*
```

[3] Die Aktionen ssm:UpdateServiceSetting und ssm:GetServiceSetting sind nur für die folgenden Ressourcen zulässig.

```
arn:aws:ssm:*:*:servicesetting/ssm/opsitem/*
arn:aws:ssm:*:*:servicesetting/ssm/opsdata/*
```

[4] Die securityhub:BatchUpdateFindings werden die Berechtigungen durch die folgende Bedingung nur für den Systems Manager-Dienst verweigert.

```
{
   "Effect": "Deny",
   "Action": "securityhub:BatchUpdateFindings",
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "securityhub:ASFFSyntaxPath/Workflow.Status": "SUPPRESSED"
    }
   }
```

```
  },
  {
   "Effect": "Deny",
   "Action": "securityhub:BatchUpdateFindings",
   "Resource": "*",
   "Condition": {
    "Null": {
     "securityhub:ASFFSyntaxPath/Confidence": false
    }
   }
  },
  {
   "Effect": "Deny",
   "Action": "securityhub:BatchUpdateFindings",
   "Resource": "*",
   "Condition": {
    "Null": {
     "securityhub:ASFFSyntaxPath/Criticality": false
    }
   }
  },
  {
   "Effect": "Deny",
   "Action": "securityhub:BatchUpdateFindings",
   "Resource": "*",
   "Condition": {
    "Null": {
     "securityhub:ASFFSyntaxPath/Note.Text": false
    }
   }
  },
  {
   "Effect": "Deny",
   "Action": "securityhub:BatchUpdateFindings",
   "Resource": "*",
   "Condition": {
    "Null": {
     "securityhub:ASFFSyntaxPath/Note.UpdatedBy": false
    }
   }
  },
  {
   "Effect": "Deny",
   "Action": "securityhub:BatchUpdateFindings",
```

```
    "Resource": "*",
    "Condition": {
     "Null": {
      "securityhub:ASFFSyntaxPath/RelatedFindings": false
     }
    }
   },
   {
    "Effect": "Deny",
    "Action": "securityhub:BatchUpdateFindings",
    "Resource": "*",
    "Condition": {
     "Null": {
      "securityhub:ASFFSyntaxPath/Types": false
     }
    }
   },
   {
    "Effect": "Deny",
    "Action": "securityhub:BatchUpdateFindings",
    "Resource": "*",
    "Condition": {
     "Null": {
      "securityhub:ASFFSyntaxPath/UserDefinedFields.key": false
     }
    }
   },
   {
    "Effect": "Deny",
    "Action": "securityhub:BatchUpdateFindings",
    "Resource": "*",
    "Condition": {
     "Null": {
      "securityhub:ASFFSyntaxPath/UserDefinedFields.value": false
     }
    }
   },
   {
    "Effect": "Deny",
    "Action": "securityhub:BatchUpdateFindings",
    "Resource": "*",
    "Condition": {
     "Null": {
      "securityhub:ASFFSyntaxPath/VerificationState": false
```

```
    }
  }
```

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSSystemsManagerOpsDataSyncServiceRolePolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: Amazon SSMManaged EC2 InstanceDefaultPolicy

Sie sollten nur IAM-Rollen für EC2 Amazon-Instances anhängen`AmazonSSMManagedEC2InstanceDefaultPolicy`, für die Sie die Berechtigung zur Nutzung von Systems Manager Funktionen benötigen. Sie sollten diese Rolle nicht anderen IAM-Entitäten wie IAM-Benutzern und IAM-Gruppen oder IAM-Rollen zuordnen, die anderen Zwecken dienen. Weitere Informationen finden Sie unter Automatisches Verwalten von EC2 Instanzen mit der Standard-Host-Management-Konfiguration.

Diese Richtlinie gewährt Ihrer EC2 Amazon-Instance Berechtigungen, mit dem Systems Manager Manager-Service in der Cloud zu kommunizieren, um eine Vielzahl von Aufgaben auszuführen. SSM Agent Sie gewährt auch Berechtigungen für die beiden Services, die Autorisierungstoken bereitstellen, um sicherzustellen, dass Operationen auf der richtigen Instance ausgeführt werden.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm` – Ermöglicht Prinzipalen das Abrufen von Dokumenten, das Ausführen von Befehlen mit Run Command, das Einrichten von Sitzungen mit Session Manager, das Erfassen einer Bestandsaufnahme der Instance und das Scannen nach Patches und Patch-Compliance mit Patch Manager.
- `ssmmessages` – Ermöglicht Prinzipalen, für jede Instance auf ein personalisiertes Autorisierungstoken zuzugreifen, das vom Amazon Message Gateway Service erstellt wurde. Systems Manager validiert das personalisierte Autorisierungs-Token anhand des Amazon-Ressourcennamens (ARN) der Instance, der in des API-Vorgangs angegeben wurde. Dieser Zugriff ist erforderlich, um sicherzustellen, dass SSM Agent die API-Vorgänge auf der richtigen Instance ausführt.
- `ec2messages` – Ermöglicht Prinzipalen, für jede Instance auf ein personalisiertes Autorisierungstoken zuzugreifen, das vom Amazon Message Delivery Service erstellt wurde. Systems Manager validiert das personalisierte Autorisierungs-Token anhand des Amazon-Ressourcennamens (ARN) der Instance, der in des API-Vorgangs angegeben wurde. Dieser Zugriff

ist erforderlich, um sicherzustellen, dass SSM Agent die API-Vorgänge auf der richtigen Instance ausführt.

Weiterführende Informationen zu den `ssmmessages`- und `ec2messages`-Endpunkten einschließlich der Unterschiede zwischen den beiden, finden Sie unter API-Vorgänge (`ssmmessages`- und `ec2messages`-Endpunkte) im Zusammenhang mit Agenten.

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter Amazon SSMManaged EC2 InstanceDefaultPolicy im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: SSMQuick SetupRolePolicy

Sie können keine Verbindungen SSMQuick SetupRolePolicy zu Ihren IAM-Entitäten herstellen. Diese Richtlinie ist an eine servicegebundene Rolle angehängt, die Systems Manager die Durchführung von Aktionen in Ihrem Namen ermöglicht. Weitere Informationen finden Sie unter Verwenden von Rollen zur Erhaltung des Zustands und der Konsistenz der von Quick Setup bereitgestellten Ressourcen.

Diese Richtlinie gewährt schreibgeschützte Berechtigungen, die es Systems Manager ermöglichen, den Zustand der Konfiguration zu überprüfen, die konsistente Verwendung von Parametern und bereitgestellten Ressourcen sicherzustellen und Ressourcen zu korrigieren, wenn Abweichungen festgestellt werden. Sie gewährt auch administrative Berechtigungen zum Erstellen einer serviceverknüpften Rolle.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm` – Ermöglicht Prinzipalen das Lesen von Informationen zu Resource Data Syncs und SSM-Dokumenten im Systems Manager, einschließlich in delegierten Administratorkonten. Dies ist erforderlich, sodass Quick Setup den Status bestimmen kann, in dem sich die konfigurierten Ressourcen befinden sollen.
- `organizations` – Ermöglicht Prinzipalen das Lesen von Informationen über die Mitgliedskonten, die zu einer Organisation gehören, wie sie unter konfiguriert ist. AWS Organizations Dies ist erforderlich, sodass Quick Setup alle Konten in einer Organisation identifizieren kann, in der Ressourcenzustandsprüfungen durchgeführt werden sollen.
- `cloudformation`— Ermöglicht Prinzipalen das Lesen von Informationen von. CloudFormation Dies ist erforderlich, um Daten über die CloudFormation Stacks zu sammeln, die zur Verwaltung

des Ressourcenstatus und der CloudFormation Stackset-Operationen verwendet werden. Quick Setup

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter SSMQuickSetupRolePolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSQuick SetupDeploymentRolePolicy

Die verwaltete Richtlinie `AWSQuickSetupDeploymentRolePolicy` unterstützt mehrere Quick Setup-Konfigurationstypen. Diese Konfigurationstypen erstellen IAM-Rollen und -Automatisierungen, die wiederum häufig verwendete Amazon-Web-Services-Services und -Funktionen mit empfohlenen bewährten Methoden konfigurieren.

Sie können `AWSQuickSetupDeploymentRolePolicy` an Ihre IAM-Entitäten anhängen.

Diese Richtlinie gewährt Administratorberechtigungen, die zum Erstellen von Ressourcen für die folgenden Quick Setup-Konfigurationen erforderlich sind:

- Amazon-EC2-Host-Verwaltung mit Quick Setup einrichten
- Erstellen Sie einen AWS Config-Konfigurationsrekorder mit Quick Setup
- Stellen Sie das AWS Config Conformance Pack bereit mit Quick Setup
- DevOps-Guru einrichten mit Quick Setup
- Bereitstellen von Distributor-Paketen mit Quick Setup
- Automatisches Stoppen und Starten von EC2-Instances nach einem Zeitplan mit Quick Setup

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm`— Ermöglicht es Prinzipalen, SSM-Dokumente zu lesen, zu erstellen, zu aktualisieren und zu löschen, deren Namen mit „AWSQuickSetup-“ oder „AWSOperationsPack-“ beginnen, wenn sie über aufgerufen werden CloudFormation; bestimmte AWS eigene Dokumente wie "AWSQuickSetupType-ManageInstanceProfile„zu lesen; Verknüpfungen für Dokumente und AWS eigene Quick Setup Dokumente zu erstellen, zu aktualisieren und zu löschen, wenn sie über aufgerufen werden CloudFormation; und ältere Ressourcen zu bereinigen, die mit gekennzeichnet sind. `QuickSetupID` Dies ermöglicht Quick Setup die Bereitstellung und Verwaltung von Automatisierungsworkflows und -zuordnungen.

- `cloudformation`— Ermöglicht Prinzipalen, Informationen über CloudFormation Stacks und Stack-Sets zu lesen und CloudFormation Stacks zu erstellen, zu aktualisieren und zu löschen und Sets für Ressourcen zu ändern, deren Namen mit "StackSet-AWS- -“ beginnen. QuickSetup Dies ermöglicht Quick Setup die Verwaltung von Infrastrukturbereitstellungen über Konten und Regionen hinweg.
- `config`— Ermöglicht Principals, Informationen über AWS Config Conformance Packs und deren Status zu lesen und Conformance Packs zu erstellen und zu löschen, deren Namen mit „AWS-QuickSetup -“ beginnen, wenn sie über aufgerufen werden. CloudFormation Dies ermöglicht Quick Setup die Implementierung von Konfigurationen zur Compliance-Überwachung.
- `events`— Ermöglicht Prinzipalen die Verwaltung von EventBridge Regeln und Zielen für Ressourcen, deren Namen "-“ enthalten. QuickSetup Dies ermöglicht Quick Setup die Erstellung von geplanten Automatisierungsworkflows.
- `iam`— Ermöglicht es Prinzipalen, serviceverknüpfte Rollen für AWS Config und Systems Manager zu erstellen, zu verwalten und zu löschen, deren Namen mit „AWS-“ oder "AWSOperationsPack QuickSetup -“ beginnen, wenn sie über aufgerufen werden CloudFormation; diese Rollen an Systems Manager und EventBridge Services weiterzugeben; diesen Rollen spezifische AWS verwaltete Richtlinien zuzuordnen; und Berechtigungsgrenzen mithilfe bestimmter Quick Setup verwalteter Richtlinien festzulegen. Auf diese Weise kann Quick Setup die für den Betrieb erforderlichen Servicerollen erstellen.
- `resource-groups` – Ermöglicht Prinzipalen das Abrufen von Ressourcengruppenabfragen. Dies ermöglicht Quick Setup die gezielte Nutzung bestimmter Ressourcensätze für das Konfigurationsmanagement.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetupDeploymentRolePolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSQuick SetupPatchPolicyDeploymentRolePolicy

Die verwaltete Richtlinie `AWSQuickSetupPatchPolicyDeploymentRolePolicy` unterstützt den Typ Das Patchen für Instances in einer Organisation mithilfe einer Patch-Richtlinie für Quick Setup konfigurieren Quick Setup. Dieser Konfigurationstyp hilft dabei, das Patchen von Anwendungen und Knoten in einem einzigen Konto oder in Ihrem gesamten Organisation zu automatisieren.

Sie können `AWSQuickSetupPatchPolicyDeploymentRolePolicy` zu Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie auch einer Servicerolle zu, mit der Sie Systems Manager in Ihrem Namen ausführen können.

Diese Richtlinie erteilt Administratorberechtigungen, die es Quick Setup ermöglichen, Ressourcen zu erstellen, die mit einer Patch-Richtlinienkonfiguration verknüpft sind.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `iam` – Ermöglicht Prinzipalen das Verwalten und Löschen von IAM-Rollen, die für Automation-Konfigurationsaufgaben erforderlich sind, sowie das Verwalten von Automation-Rollenrichtlinien.
- `cloudformation`— Ermöglicht Prinzipalen das Lesen von CloudFormation Stack-Informationen und das Steuern von CloudFormation Stacks, die Quick Setup mithilfe von CloudFormation StackSets erstellt wurden.
- `ssm` – Ermöglicht Prinzipalen das Erstellen, Aktualisieren, Lesen und Löschen von Automation-Runbooks, die für Konfigurationsaufgaben erforderlich sind, sowie das Erstellen, Aktualisieren und Löschen von State Manager-Verknüpfungen.

- `resource-groups` – Ermöglicht Prinzipalen das Abrufen von Ressourcenabfragen, die mit Ressourcengruppen verknüpft sind, auf die Quick Setup-Konfigurationen abzielen.

- `s3` – Ermöglicht Prinzipalen, Amazon-S3-Buckets aufzulisten und die Buckets zum Speichern von Zugriffsprotokollen für Patch-Richtlinien zu verwalten.
- `lambda`— Ermöglicht es den Prinzipalen, AWS Lambda Korrekturfunktionen zu verwalten, die dafür sorgen, dass die Konfigurationen im richtigen Zustand bleiben.
- `logs` – Ermöglicht Prinzipalen, Protokollgruppen für Lambda-Konfigurationsressourcen zu beschreiben und zu verwalten.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetupPatchPolicyDeploymentRolePolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSQuick SetupPatchPolicyBaselineAccess

Die verwaltete Richtlinie `AWSQuickSetupPatchPolicyBaselineAccess` unterstützt den Typ Das Patchen für Instances in einer Organisation mithilfe einer Patch-Richtlinie für Quick Setup konfigurieren Quick Setup. Dieser Konfigurationstyp hilft dabei, das Patchen von Anwendungen und Knoten in einem einzigen Konto oder in Ihrem gesamten Organisation zu automatisieren.

Sie können `AWSQuickSetupPatchPolicyBaselineAccess` zu Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie auch einer Servicerolle zu, mit der Sie Systems Manager in Ihrem Namen ausführen können.

Diese Richtlinie gewährt nur Leseberechtigungen für den Zugriff auf Patch-Baselines, die von einem Administrator in der aktuellen AWS-Konto Version oder in der Organisation konfiguriert wurden, die sie verwendet. Quick Setup Die Patch-Baselines werden in einem Amazon-S3-Bucket gespeichert und können für das Patchen von Instances in einem einzelnen Konto oder in einer gesamten Organisation verwendet werden.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgende Berechtigung.

- `s3` – Ermöglicht Prinzipalen das Lesen von Patch-Baseline Overrides, die in Amazon-S3-Buckets gespeichert sind.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetupPatchPolicyBaselineAccess im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: **`AWSSystemsManagerEnableExplorerExecutionPolicy`**

Die verwaltete Richtlinie `AWSSystemsManagerEnableExplorerExecutionPolicy` unterstützt die AktivierungExplorer, ein Tool in AWS Systems Manager.

Sie können `AWSSystemsManagerEnableExplorerExecutionPolicy` zu Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie auch einer Servicerolle zu, mit der Sie Systems Manager in Ihrem Namen ausführen können.

Diese Richtlinie gewährt administrative Berechtigungen für die Aktivierung von Explorer. Dazu gehören Berechtigungen zum Aktualisieren der zugehörigen Systems Manager-Serviceeinstellungen und zum Erstellen einer servicebezogenen Rolle für Systems Manager.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `config` – Ermöglicht Prinzipalen, bei der Aktivierung von Explorer zu helfen, indem sie schreibgeschützten Zugriff auf die Details des Konfigurationsrekorders gewähren.

- `iam` – Ermöglicht Prinzipalen, bei der Aktivierung von Explorer zu helfen.
- `ssm` – Ermöglicht Prinzipalen, einen Automation-Workflow zu starten, der Explorer aktiviert.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSSystemsManagerEnableExplorerExecutionPolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: **`AWSSystemsManagerEnableConfigRecordingExecutionPolicy`**

Die verwaltete Richtlinie `AWSSystemsManagerEnableConfigRecordingExecutionPolicy` unterstützt den Konfigurationstypen Erstellen Sie einen AWS Config-Konfigurationsrekorder mit Quick Setup Quick Setup. Dieser Konfigurationstyp ermöglicht Quick Setup das Nachverfolgen und Aufzeichnen von Änderungen an den AWS Ressourcentypen, für die Sie sich entscheiden AWS Config. Darüber hinaus kann Quick Setup die Übermittlungs- und Benachrichtigungsoptionen für die aufgezeichneten Daten konfigurieren.

Sie können `AWSSystemsManagerEnableConfigRecordingExecutionPolicy` zu Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie auch einer Servicerolle zu, mit der Sie Systems Manager in Ihrem Namen ausführen können.

Diese Richtlinie gewährt Administratorberechtigungen, mit denen Sie Quick Setup die AWS Config Konfigurationsaufzeichnung aktivieren und konfigurieren können.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `s3` – Ermöglicht Prinzipalen das Erstellen und Konfigurieren von Amazon-S3-Buckets für die Bereitstellung von Konfigurationsaufzeichnungen.
- `sns` – Ermöglicht Prinzipalen das Auflisten und Erstellen von Amazon-SNS-Themen.
- `config` – Ermöglicht Prinzipalen, den Konfigurationsrekorder zu konfigurieren und zu starten und bei der Aktivierung von Explorer zu helfen.
- `iam`— Ermöglicht es Prinzipalen, eine dienstbezogene Rolle für Systems Manager zu erstellen, abzurufen und zu übergeben AWS Config, eine dienstbezogene Rolle für Systems Manager zu erstellen und bei der Aktivierung zu helfen. Explorer
- `ssm` – Ermöglicht Prinzipalen, einen Automation-Workflow zu starten, der Explorer aktiviert.

- `compute-optimizer`— Ermöglicht es den Prinzipalen, bei der Aktivierung zu helfen, Explorer indem sie nur Lesezugriff gewähren, um festzustellen, ob eine Ressource registriert ist. AWS Compute Optimizer
- `support` – Ermöglicht Prinzipalen, bei der Aktivierung von Explorer zu helfen, indem sie schreibgeschützten Zugriff gewähren, um festzustellen, ob eine Ressource mit AWS Compute Optimizer registriert ist.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSSystemsManagerEnableConfigRecordingExecutionPolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSQuick SetupDevOpsGuruPermissionsBoundary

> **Note**
>
> Bei dieser Richtlinie handelt es sich um eine Berechtigungsgrenze. Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Sie sollten Richtlinien für Quick Setup-Berechtigungsgrenzen nicht eigenständig verwenden und anhängen. Quick Setup Richtlinien für Berechtigungsgrenzen sollten nur von Quick Setup verwalteten Rollen zugewiesen werden. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.

Die verwaltete Richtlinie `AWSQuickSetupDevOpsGuruPermissionsBoundary` unterstützt den Typ DevOps-Guru einrichten mit Quick Setup. Der Konfigurationstyp aktiviert den auf maschinellem Lernen basierenden Amazon Guru. DevOps Der DevOps Guru-Service kann dazu beitragen, die Betriebsleistung und Verfügbarkeit einer Anwendung zu verbessern.

Wenn Sie eine `AWSQuickSetupDevOpsGuruPermissionsBoundary`-Konfiguration mit Quick Setup erstellen, wendet das System diese Berechtigungsgrenze auf die IAM-Rollen an, die bei der Bereitstellung der Konfiguration erstellt werden. Die Berechtigungsgrenze beschränkt den Geltungsbereich der Rollen, die Quick Setup erstellt werden.

Diese Richtlinie gewährt Administratorberechtigungen, die es ermöglichen, Amazon DevOps Guru Quick Setup zu aktivieren und zu konfigurieren.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `iam`— Ermöglicht Principals, dienstbezogene Rollen für DevOps Guru und Systems Manager zu erstellen und Rollen aufzulisten, die bei der Aktivierung helfen. Explorer
- `cloudformation` – Ermöglicht Prinzipalen das Auflisten und Beschreiben von CloudFormation -Stacks.
- `sns` – Ermöglicht Prinzipalen das Auflisten und Erstellen von Amazon-SNS-Themen.
- `devops-guru`— Ermöglicht Prinzipalen, DevOps Guru zu konfigurieren und einen Benachrichtigungskanal hinzuzufügen.
- `config` – – Ermöglicht Prinzipalen, bei der Aktivierung von Explorer zu helfen, indem sie schreibgeschützten Zugriff auf die Details des Konfigurationsrekorders gewähren.
- `ssm` – Ermöglicht Prinzipalen, einen Automation-Workflow zu starten, der Explorer, aktiviert, sowie Explorer-Serviceeinstellungen zu lesen und zu aktualisieren.
- `compute-optimizer`— Ermöglicht es den Prinzipalen, bei der Aktivierung zu helfen, Explorer indem sie nur Lesezugriff gewähren, um festzustellen, ob eine Ressource registriert ist. AWS Compute Optimizer
- `support` – Ermöglicht Prinzipalen, bei der Aktivierung von Explorer zu helfen, indem sie schreibgeschützten Zugriff gewähren, um festzustellen, ob eine Ressource mit AWS Compute Optimizer registriert ist.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetupDevOpsGuruPermissionsBoundary im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSQuick SetupDistributorPermissionsBoundary

> **Note**
>
> Bei dieser Richtlinie handelt es sich um eine Berechtigungsgrenze. Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Sie sollten Richtlinien für Quick Setup-Berechtigungsgrenzen nicht eigenständig verwenden und anhängen. Quick Setup Richtlinien für Berechtigungsgrenzen sollten nur von Quick Setup verwalteten Rollen zugewiesen werden. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.

Die verwaltete Richtlinie `AWSQuickSetupDistributorPermissionsBoundary` unterstützt den Konfigurationstypen Bereitstellen von Distributor-Paketen mit Quick Setup Quick Setup. Der Konfigurationstyp ermöglicht die Verteilung von Softwarepaketen, z. B. Agenten, an Ihre Amazon Elastic Compute Cloud (Amazon EC2) -Instances mithilfe von Distributor, einem Tool in AWS Systems Manager.

Wenn Sie eine `AWSQuickSetupDistributorPermissionsBoundary`-Konfiguration mit Quick Setup erstellen, wendet das System diese Berechtigungsgrenze auf die IAM-Rollen an, die bei der Bereitstellung der Konfiguration erstellt werden. Die Berechtigungsgrenze beschränkt den Geltungsbereich der Rollen, die Quick Setup erstellt werden.

Diese Richtlinie gewährt Administratorberechtigungen, die die Verteilung von Softwarepaketen, z. B. Agenten, an Ihre EC2 Amazon-Instances mithilfe von Distributor ermöglichenQuick Setup.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `iam`— Ermöglicht Principals das Abrufen und Weitergeben der Vertriebsautomatisierungsrolle; das Erstellen, Lesen, Aktualisieren und Löschen der Standard-Instance-Rolle; das Übergeben der Standard-Instance-Rolle an Amazon EC2 und Systems Manager; das Anhängen von Instance-Management-Richtlinien an Instance-Rollen; das Erstellen einer serviceverknüpften Rolle für Systems Manager; das Hinzufügen der Standard-Instance-Rolle zu Instance-Profilen; das Lesen von Informationen über IAM-Rollen und Instance-Profile; und das Erstellen des Standard-Instance-Profils.
- `ec2`— Ermöglicht Principals, das Standard-Instanzprofil EC2 Instanzen zuzuordnen und bei der Aktivierung zu helfenExplorer.
- `ssm` – Ermöglicht Prinzipalen, Automatisierungs-Workflows zu starten, die Instances konfigurieren und Pakete installieren, sowie den Automatisierungs-Workflow zu starten, der Explorer aktiviert, und Explorer-Serviceeinstellungen zu lesen und zu aktualisieren.
- `config` – Ermöglicht Prinzipalen, bei der Aktivierung von Explorer zu helfen, indem sie schreibgeschützten Zugriff auf die Details des Konfigurationsrekorders gewähren.
- `compute-optimizer`— Ermöglicht es den Prinzipalen, bei der Aktivierung zu helfen, Explorer indem sie nur Lesezugriff gewähren, um festzustellen, ob eine Ressource registriert ist. AWS Compute Optimizer
- `support` – Ermöglicht Prinzipalen, bei der Aktivierung von Explorer zu helfen, indem sie schreibgeschützten Zugriff gewähren, um festzustellen, ob eine Ressource mit AWS Compute Optimizer registriert ist.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetupDistributorPermissionsBoundary im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: Einrichtung AWSQuick SSMHost MgmtPermissionsBoundary

> **Note**
>
> Bei dieser Richtlinie handelt es sich um eine Berechtigungsgrenze. Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Sie sollten Richtlinien für Quick Setup-Berechtigungsgrenzen nicht eigenständig verwenden und anhängen. Quick Setup Richtlinien für Berechtigungsgrenzen sollten nur von Quick Setup verwalteten Rollen zugewiesen werden. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.

Die verwaltete Richtlinie `AWSQuickSetupSSMHostMgmtPermissionsBoundary` unterstützt den Konfigurationstypen Amazon-EC2-Host-Verwaltung mit Quick Setup einrichten Quick Setup. Dieser Konfigurationstyp konfiguriert IAM-Rollen und ermöglicht häufig verwendete Systems Manager Manager-Tools zur sicheren Verwaltung Ihrer EC2 Amazon-Instances.

Wenn Sie eine `AWSQuickSetupSSMHostMgmtPermissionsBoundary`-Konfiguration mit Quick Setup erstellen, wendet das System diese Berechtigungsgrenze auf die IAM-Rollen an, die bei der Bereitstellung der Konfiguration erstellt werden. Die Berechtigungsgrenze beschränkt den Geltungsbereich der Rollen, die Quick Setup erstellt werden.

Diese Richtlinie gewährt Administratorberechtigungen, mit denen Quick Setup die Systems Manager Manager-Tools aktiviert und konfiguriert werden können, die für die sichere Verwaltung von EC2 Instanzen erforderlich sind.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `iam` – Ermöglicht Prinzipalen das Abrufen und Übergeben der Servicerolle an Automation. Ermöglicht Principals, die Standard-Instance-Rolle zu erstellen, zu lesen, zu aktualisieren und zu löschen, die Standard-Instance-Rolle an Amazon EC2 und Systems Manager zu übergeben,

Instance-Management-Richtlinien an Instance-Rollen anzuhängen, eine serviceverknüpfte Rolle für Systems Manager zu erstellen, die Standard-Instance-Rolle zu Instance-Profilen hinzuzufügen, Informationen über IAM-Rollen und Instance-Profile zu lesen und das Standard-Instance-Profil zu erstellen.

- `ec2`— Ermöglicht Principals, das Standard-Instance-Profil Instances zuzuordnen und zu trennen. EC2
- `ssm` – Ermöglicht Prinzipalen, Automation-Workflows zu starten, die Explorer aktivieren, Explorer-Serviceeinstellungen zu lesen und zu aktualisieren, Instances zu konfigurieren und Systems-Manager-Tools in Instances zu aktivieren.
- `compute-optimizer`— Ermöglicht es den Prinzipalen, bei der Aktivierung zu helfen, Explorer indem sie nur Lesezugriff gewähren, um festzustellen, ob eine Ressource registriert ist. AWS Compute Optimizer
- `support` – Ermöglicht Prinzipalen, bei der Aktivierung von Explorer zu helfen, indem sie schreibgeschützten Zugriff gewähren, um festzustellen, ob eine Ressource mit AWS Compute Optimizer registriert ist.

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetup SSMHost MgmtPermissionsBoundary im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSQuick SetupPatchPolicyPermissionsBoundary

> **Note**
>
> Bei dieser Richtlinie handelt es sich um eine Berechtigungsgrenze. Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Sie sollten Richtlinien für Quick Setup-Berechtigungsgrenzen nicht eigenständig verwenden und anhängen. Quick Setup Richtlinien für Berechtigungsgrenzen sollten nur von Quick Setup verwalteten Rollen zugewiesen werden. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.

Die verwaltete Richtlinie `AWSQuickSetupPatchPolicyPermissionsBoundary` unterstützt den Typ Das Patchen für Instances in einer Organisation mithilfe einer Patch-Richtlinie für Quick Setup

konfigurieren Quick Setup. Dieser Konfigurationstyp hilft dabei, das Patchen von Anwendungen und Knoten in einem einzigen Konto oder in Ihrem gesamten Organisation zu automatisieren.

Wenn Sie eine `AWSQuickSetupPatchPolicyPermissionsBoundary`-Konfiguration mit Quick Setup erstellen, wendet das System diese Berechtigungsgrenze auf die IAM-Rollen an, die bei der Bereitstellung der Konfiguration erstellt werden. Die Berechtigungsgrenze beschränkt den Geltungsbereich der Rollen, die Quick Setup erstellt werden.

Diese Richtlinie erteilt Administratorberechtigungen, die es Quick Setup ermöglichen, Patch-Richtlinien in Patch Manager zu aktivieren und zu konfigurieren, einem Tool in AWS Systems Manager.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `iam`— Ermöglicht Principals, die Patch Manager Automatisierungsrolle zu erhalten, Automatisierungsrollen an Patch Manager Patching-Operationen weiterzugeben, die Standard-Instance-Rolle zu erstellen`AmazonSSMRoleForInstancesQuickSetup`, die Standard-Instance-Rolle an Amazon EC2 und Systems Manager zu übergeben, ausgewählte AWS verwaltete Richtlinien an die Instance-Rolle anzuhängen, eine serviceverknüpfte Rolle für Systems Manager zu erstellen, die Standard-Instance-Rolle zu Instance-Profilen hinzuzufügen, Informationen über Instance-Profile und Rollen zu lesen, ein Standard-Instance-Profil zu erstellen und Rollen zu taggen, die Berechtigungen haben für liest Patch-Baseline-Overrides.
- `ssm` – Ermöglicht Prinzipalen, die Instance-Rolle zu aktualisieren (diese wird von Systems Manager verwaltet), Zuordnungen zu verwalten, die durch die in Quick Setup erstellten Patch Manager-Patch-Richtlinien erstellt wurden, Instances zu markieren, auf die eine Patch-Richtlinienkonfiguration abzielt, Informationen über Instances und den Patching-Status zu lesen, Automation-Workflows zu starten, die Instance-Patching konfigurieren, aktivieren und korrigieren, Automatisierungsworkflows zu starten, die Explorer aktivieren, bei der Aktivierung von Explorer zu helfen sowie Explorer-Serviceeinstellungen zu lesen und zu aktualisieren.
- `ec2`— Ermöglicht es Prinzipalen, das Standard-Instanzprofil Instanzen zuzuordnen und zu trennen, EC2 Instances zu taggen, auf die eine Patch-Policy-Konfiguration abzielt, Instances zu taggen, auf die eine Patch-Policy-Konfiguration abzielt, und bei der Aktivierung zu helfen. Explorer
- `s3` – Ermöglicht Prinzipalen die Erstellung und Konfiguration von S3-Buckets zum Speichern von Patch-Baseline-Overrides.

- `lambda`— Ermöglicht Prinzipalen das Aufrufen von AWS Lambda Funktionen zur Konfiguration von Patches und das Ausführen von Bereinigungsvorgängen, nachdem eine Quick Setup Patch-Policy-Konfiguration gelöscht wurde.
- `logs`— Ermöglicht Prinzipalen, die Protokollierung für Funktionen zu konfigurieren. Patch Manager Quick Setup AWS Lambda
- `config` – Ermöglicht Prinzipalen, bei der Aktivierung von Explorer zu helfen, indem sie schreibgeschützten Zugriff auf die Details des Konfigurationsrekorders gewähren.
- `compute-optimizer`— Ermöglicht es den Prinzipalen, bei der Aktivierung zu helfen, Explorer indem sie nur Lesezugriff gewähren, um festzustellen, ob eine Ressource registriert ist. AWS Compute Optimizer
- `support` – Ermöglicht Prinzipalen, bei der Aktivierung von Explorer zu helfen, indem sie schreibgeschützten Zugriff gewähren, um festzustellen, ob eine Ressource mit AWS Compute Optimizer registriert ist.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetupPatchPolicyPermissionsBoundary im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSQuick SetupSchedulerPermissionsBoundary

> **Note**
>
> Bei dieser Richtlinie handelt es sich um eine Berechtigungsgrenze. Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Sie sollten Richtlinien für Quick Setup-Berechtigungsgrenzen nicht eigenständig verwenden und anhängen. Quick Setup Richtlinien für Berechtigungsgrenzen sollten nur von Quick Setup verwalteten Rollen zugewiesen werden. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.

Die verwaltete Richtlinie `AWSQuickSetupSchedulerPermissionsBoundary` unterstützt den Konfigurationstypen Automatisches Stoppen und Starten von EC2-Instances nach einem Zeitplan mit Quick Setup Quick Setup. Mit diesem Konfigurationstyp können Sie Ihre EC2 Instances und andere Ressourcen zu den von Ihnen angegebenen Zeiten beenden und starten.

Wenn Sie eine `AWSQuickSetupSchedulerPermissionsBoundary`-Konfiguration mit Quick Setup erstellen, wendet das System diese Berechtigungsgrenze auf die IAM-Rollen an, die bei der Bereitstellung der Konfiguration erstellt werden. Die Berechtigungsgrenze beschränkt den Geltungsbereich der Rollen, die Quick Setup erstellt werden.

Diese Richtlinie gewährt Administratorberechtigungen, die es Quick Setup ermöglichen, geplante Operationen auf EC2 Instanzen und anderen Ressourcen zu aktivieren und zu konfigurieren.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `iam`— Ermöglicht Principals das Abrufen und Übergeben von Rollen für Automatisierungsaktionen der Instanzverwaltung, das Verwalten, Weitergeben und Anhängen von Standard-Instanzrollen für das EC2 Instanzmanagement, das Erstellen von Standard-Instanzprofilen, das Hinzufügen von Standard-Instanzrollen zu Instanzprofilen, das Erstellen einer serviceverknüpften Rolle fürSystems Manager, das Lesen von Informationen über IAM-Rollen und Instanzprofile, das Zuordnen eines Standard-Instanzprofils zu EC2 Instances und das Starten von Automatisierungs-Workflows zur Konfiguration von Instanzen und zum Aktivieren von Systems Manager Tools für diese.
- `ssm` – Ermöglicht Prinzipalen, Automation-Workflows zu starten, die Explorer aktivieren, sowie Explorer-Serviceeinstellungen zu lesen und zu aktualisieren.
- ec2 – Ermöglicht Prinzipalen, gezielte Instances zu lokalisieren und sie nach einem Zeitplan zu starten und zu beenden.
- `config` – Ermöglicht Prinzipalen, bei der Aktivierung von Explorer zu helfen, indem sie schreibgeschützten Zugriff auf die Details des Konfigurationsrekorders gewähren.
- `compute-optimizer`— Ermöglicht es den Prinzipalen, bei der Aktivierung zu helfen, Explorer indem sie nur Lesezugriff gewähren, um festzustellen, ob eine Ressource registriert ist AWS Compute Optimizer.
- `support`— Ermöglicht es den Prinzipalen, bei der Aktivierung zu helfen, Explorer indem sie nur Lesezugriff auf Schecks für ein Konto gewähren. AWS Trusted Advisor

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetupSchedulerPermissionsBoundary im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: Einrichtung AWSQuick CFGCPacks PermissionsBoundary

> **Note**
>
> Bei dieser Richtlinie handelt es sich um eine Berechtigungsgrenze. Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Sie sollten Richtlinien für Quick Setup-Berechtigungsgrenzen nicht eigenständig verwenden und anhängen. Quick Setup Richtlinien für Berechtigungsgrenzen sollten nur von Quick Setup verwalteten Rollen zugewiesen werden. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.

Die verwaltete Richtlinie `AWSQuickSetupCFGCPacksPermissionsBoundary` unterstützt den Konfigurationstypen Stellen Sie das AWS Config Conformance Pack bereit mit Quick Setup Quick Setup. Dieser Konfigurationstyp stellt AWS Config Conformance Packs bereit. Conformance Packs sind Sammlungen von AWS Config Regeln und Behebungsmaßnahmen, die als eine Einheit bereitgestellt werden können.

Wenn Sie eine `AWSQuickSetupCFGCPacksPermissionsBoundary`-Konfiguration mit Quick Setup erstellen, wendet das System diese Berechtigungsgrenze auf die IAM-Rollen an, die bei der Bereitstellung der Konfiguration erstellt werden. Die Berechtigungsgrenze beschränkt den Geltungsbereich der Rollen, die Quick Setup erstellt werden.

Diese Richtlinie erteilt Administratorberechtigungen, die es Quick Setup ermöglichen, Konformitätspakete AWS Config bereitzustellen.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `iam`— Ermöglicht Prinzipalen das Erstellen, Abrufen und Weitergeben einer dienstbezogenen Rolle für. AWS Config
- `sns` – Ermöglicht Prinzipalen das Auflisten von Plattformanwendungen in Amazon SNS.
- `config`— Ermöglicht Principals die Bereitstellung von AWS Config Conformance Packs, das Abrufen des Status von Conformance Packs und das Abrufen von Informationen zu Konfigurationsrekordern.

- `ssm` – Ermöglicht Prinzipalen das Abrufen von Informationen über SSM-Dokumente und Automation-Workflows, das Abrufen von Informationen über Ressourcen-Tags und das Abrufen von Informationen über Serviceeinstellungen und deren Aktualisierung.
- `compute-optimizer` – Ermöglicht Prinzipalen das Abrufen des Opt-in-Status eines Accounts.
- `support` – Ermöglicht Prinzipalen das Abrufen von Informationen über AWS Trusted Advisor - Prüfungen.

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetup CFGCPacks PermissionsBoundary im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSQuick SetupStartStopInstancesExecutionPolicy

Sie können `AWSQuickSetupStartStopInstancesExecutionPolicy` an Ihre IAM-Entitäten anhängen. Diese Richtlinie bietet Berechtigungen für Quick Setup die Verwaltung des Starts und Stoppens von EC2 Amazon-Instances mithilfe der Systems Manager Manager-Automatisierung.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ec2`— Ermöglicht Principals, EC2 Amazon-Instances, ihren Status, ihre Regionen und Tags zu beschreiben. Ermöglicht auch das Starten und Stoppen bestimmter EC2 Amazon-Instances.
- `ssm`— Ermöglicht es Prinzipalen, den Kalenderstatus aus Quick Setup Änderungskalendern abzurufen, Verknüpfungen zu starten und Automatisierungsdokumente für die Instanzplanung auszuführen.
- `iam`— Ermöglicht Principals, Quick Setup IAM-Rollen zur automatisierten Ausführung an Systems Manager zu übergeben, mit Bedingungen, die den Service auf ssm.amazonaws.com und bestimmte Ressourcen beschränken. ARNs

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetupStartStopInstancesExecutionPolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSQuick SetupStart SSMAssociations ExecutionPolicy

Diese Richtlinie gewährt Quick Setup Berechtigungen, mit denen das `AWSQuickSetupType-Scheduler-ChangeCalendarState`-Automation-Runbook ausgeführt werden kann. Dieses

Runbook wird zur Verwaltung von Änderungskalenderstatus für geplante Operationen in Quick Setup Konfigurationen verwendet.

Sie können `AWSQuickSetupStartSSMAssociationsExecutionPolicy` zu Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie auch einer Servicerolle zu, mit der Sie Systems Manager in Ihrem Namen ausführen können.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm` – Ermöglicht Prinzipalen, Automatisierungsausführungen speziell für das `AWSQuickSetupType-Scheduler-ChangeCalendarState`-Dokument zu starten. Dies ist erforderlichQuick Setup, um den Status des Änderungskalenders für geplante Operationen zu verwalten.
- `iam`— Ermöglicht Prinzipalen, Rollen, deren Namen mit „AWS- QuickSetup -“ beginnen, an den Systems Manager Manager-Service zu übergeben. Diese Berechtigung ist auf die Verwendung mit bestimmten SSM-Dokumenten beschränkt, die sich auf die Verwaltung von Änderungskalendern beziehen. Dies ist erforderlich für Quick Setup, um die entsprechende Ausführungsrolle an den Automatisierungsprozess zu übergeben.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetupStartSSMAssociationsExecutionPolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy

Die Richtlinie `AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy` bietet Berechtigungen für die Diagnose von Problemen mit Knoten, die mit Systems-Manager-Services interagieren, indem Automation-Workflows in Konten und Regionen gestartet werden, in denen Knoten verwaltet werden.

Sie können `AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy` zu Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie auch einer Servicerolle zu, mit der Systems Manager Diagnosen in Ihrem Namen ausführen kann.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm` – Ermöglicht Prinzipalen das Ausführen von Automation-Runbooks, die Knotenprobleme diagnostizieren, auf den Ausführungsstatus eines Workflows zugreifen und Details zur Automation-Ausführung abrufen. Die Richtlinie gewährt Berechtigungen zur Beschreibung von Automatisierungsausführungen, zur Beschreibung von Ausführungen von Automatisierungsschritten, zum Abrufen von Details zur Automatisierungsausführung und zum Starten von Automatisierungsausführungen für Dokumente im Zusammenhang mit der Diagnose.
- `kms` – Ermöglicht Prinzipalen die Verwendung von kundenspezifischen AWS Key Management Service -Schlüsseln für die Entschlüsselung und Datenschlüsselgenerierung beim Zugriff auf verschlüsselte Objekte in Amazon-S3-Buckets, die für Diagnosevorgänge verwendet werden. Diese Berechtigungen sind auf Schlüssel beschränkt, die mit `SystemsManagerManaged` gekennzeichnet sind und mit dem Amazon-S3-Service verwendet werden, mit spezifischen Anforderungen an den Verschlüsselungskontext.
- `sts` – Ermöglicht Prinzipalen, Rollen zur Ausführung von Diagnosen zu übernehmen, um Automation-Runbooks im selben Konto auszuführen. Diese Berechtigung ist auf Rollen mit dem `AWS-SSM-DiagnosisExecutionRole`-Benennungsmuster beschränkt und beinhaltet eine Bedingung, die sicherstellt, dass das Ressourcenkonto mit dem Prinzipalkonto übereinstimmt.
- `iam` – Ermöglicht Prinzipalen, die Verwaltungsrolle der Diagnose an Systems Manager zur Ausführung von Automation-Runbooks zu übergeben. Diese Berechtigung ist auf Rollen mit dem `AWS-SSM-DiagnosisAdminRole`-Benennungsmuster beschränkt und kann nur an den Systems-Manager-Service weitergegeben werden.
- `s3` – Ermöglicht Prinzipalen, auf Objekte in Amazon-S3-Buckets, die für Diagnosevorgänge verwendet werden, zuzugreifen, diese zu lesen, zu schreiben und zu löschen. Diese Berechtigungen sind auf Buckets mit dem `do-not-delete-ssm-diagnosis--` Benennungsmuster beschränkt und beinhalten Bedingungen, um sicherzustellen, dass Vorgänge innerhalb desselben Kontos ausgeführt werden.

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWS-SSM- DiagnosisAutomation — AdministrationRolePolicy im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy

Die verwaltete Richtlinie `AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy` gewährt Administratorberechtigungen für die Ausführung von Automations-Runbooks in einer Zielregion AWS-Konto und Region, um Probleme mit verwalteten Knoten zu diagnostizieren, die mit Systems-Manager-Services interagieren.

Sie können `AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy` zu Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie auch einer Servicerolle zu, mit der Sie Systems Manager in Ihrem Namen ausführen können.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ec2`— Ermöglicht es Principals, Amazon EC2 - und Amazon VPC-Ressourcen und deren Konfigurationen zu beschreiben, um Probleme mit Systems Manager Diensten zu diagnostizieren. Dazu gehören Berechtigungen zur Beschreibung VPCs von VPC-Attributen, VPC-Endpunkten, Subnetzen, Sicherheitsgruppen, Instanzen und Internet-Gateways.
- `ssm` – Ermöglicht Prinzipalen, diagnosespezifische Automation-Runbooks auszuführen und auf den Status und die Ausführungsmetadaten des Automation-Workflows zuzugreifen. Dazu gehören Berechtigungen zur Beschreibung von Automatisierungsschrittausführungen, zur Beschreibung von Instanzinformationen, zur Beschreibung von Automatisierungsausführungen, zum Abrufen von Details zur Automatisierungsausführung und zum Starten von Automatisierungsausführungen für bestimmte nicht verwaltete Diagnosedokumente. AWS EC2
- `kms` – Ermöglicht Prinzipalen die Verwendung von kundenspezifischen AWS Key Management Service -Schlüsseln für die Entschlüsselung und Datenschlüsselgenerierung beim Zugriff auf verschlüsselte Objekte in Amazon-S3-Buckets, die für Diagnosevorgänge verwendet werden. Diese Berechtigungen sind auf Schlüssel beschränkt, die mit `SystemsManagerManaged` gekennzeichnet sind und mit dem Amazon-S3-Service verwendet werden, mit spezifischen Anforderungen an den Verschlüsselungskontext für Diagnosebuckets.
- `iam` – Ermöglicht Prinzipalen, die Ausführungsrolle für die Diagnose an Systems Manager zur Ausführung von Automation-Dokumenten zu übergeben. Diese Berechtigung ist auf Rollen mit dem `AWS-SSM-DiagnosisExecutionRole`-Benennungsmuster beschränkt und kann nur an den Systems-Manager-Service weitergegeben werden.

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWS-SSM- DiagnosisAutomation — ExecutionRolePolicy im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWS-SSM-RemediationAutomation-AdministrationRolePolicy

Die Richtlinie `AWS-SSM-RemediationAutomation-AdministrationRolePolicy` bietet Berechtigungen zur Behebung von Problemen mit Systems-Manager-Services durch Ausführung von Aktivitäten, die in Automatisierungsdokumenten definiert sind und hauptsächlich für die Ausführung der Automatisierungsdokumente verwendet werden. Diese Richtlinie ermöglicht das Starten von Automatisierungsworkflows in Konten und Regionen, in denen Knoten verwaltet werden, um Konnektivitäts- und Konfigurationsprobleme zu beheben.

Sie können `AWS-SSM-RemediationAutomation-AdministrationRolePolicy` zu Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie auch einer Servicerolle zu, mit der Systems Manager Abhilfemaßnahmen in Ihrem Namen ausführen kann.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm` – Ermöglicht Prinzipalen das Ausführen von Automation-Runbooks, die Knotenprobleme beheben, auf den Ausführungsstatus eines Workflows zugreifen und Details zur Automation-Ausführung abrufen. Die Richtlinie gewährt Berechtigungen zur Beschreibung von Automatisierungsausführungen, zur Beschreibung von Ausführungen von Automatisierungsschritten, zum Abrufen von Details zur Automatisierungsausführung und zum Starten von Automatisierungsausführungen für Dokumente im Zusammenhang mit der Behebung.
- `kms` – Ermöglicht Prinzipalen die Verwendung von kundenspezifischen AWS Key Management Service -Schlüsseln für die Entschlüsselung und Datenschlüsselgenerierung beim Zugriff auf verschlüsselte Objekte in Amazon-S3-Buckets, die für Behebungsvorgänge verwendet werden. Diese Berechtigungen sind auf Schlüssel beschränkt, die mit `SystemsManagerManaged` gekennzeichnet sind und mit dem Amazon-S3-Service verwendet werden, mit spezifischen Anforderungen an den Verschlüsselungskontext.
- `sts` – Ermöglicht Prinzipalen, Rollen zur Ausführung von Behebungen zu übernehmen, um Automation-Runbooks im selben Konto auszuführen. Diese Berechtigung ist auf Rollen mit dem `AWS-SSM-RemediationExecutionRole`-Benennungsmuster beschränkt und beinhaltet eine Bedingung, die sicherstellt, dass das Ressourcenkonto mit dem Prinzipalkonto übereinstimmt.
- `iam` – Ermöglicht Prinzipalen, die Verwaltungsrolle der Behebung an Systems Manager zur Ausführung von Automation-Runbooks zu übergeben. Diese Berechtigung ist auf Rollen mit dem `AWS-SSM-RemediationAdminRole`-Benennungsmuster beschränkt und kann nur an den Systems-Manager-Service weitergegeben werden.

- `s3` – Ermöglicht Prinzipalen, auf Objekte in Amazon-S3-Buckets, die für Behebungsvorgänge verwendet werden, zuzugreifen, diese zu lesen, zu schreiben und zu löschen. Diese Berechtigungen sind auf Buckets mit dem `do-not-delete-ssm-diagnosis--` Benennungsmuster beschränkt und beinhalten Bedingungen, um sicherzustellen, dass Vorgänge innerhalb desselben Kontos ausgeführt werden.

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWS-SSM- RemediationAutomation — AdministrationRolePolicy im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWS-SSM-RemediationAutomation-ExecutionRolePolicy

Die verwaltete Richtlinie `AWS-SSM-RemediationAutomation-ExecutionRolePolicy` bietet die Berechtigungen zum Ausführen von Automation-Runbooks in einem bestimmten Zielkonto und einer bestimmten Region, um Netzwerk- und Konnektivitätsprobleme mit verwalteten Knoten zu beheben, die mit Systems-Manager-Services interagieren. Diese Richtlinie ermöglicht Abhilfemaßnahmen, die in Automation-Dokumenten definiert sind und hauptsächlich für die Ausführung der Automation-Dokumente zur Behebung von Konnektivitäts- und Konfigurationsproblemen verwendet werden.

Sie können die -Richtlinie auch Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie einer Servicerolle hinzu, mit der Systems Manager in Ihrem Namen Abhilfemaßnahmen ausführen kann.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm` – Ermöglicht Prinzipalen das Abrufen von Informationen über Automation-Ausführungen und deren Schrittausführungen sowie das Starten bestimmter Runbooks für die Automatisierung von Problembehebungen, einschließlich `AWS-OrchestrateUnmanagedEC2Actions-` und `AWS-RemediateSSMAgent`-Dokumenten. Die Richtlinie gewährt Berechtigungen zur Beschreibung von Automatisierungsausführungen, zur Beschreibung von Ausführungen von Automatisierungsschritten, zum Abrufen von Details zur Automatisierungsausführung und zum Starten von Automatisierungsausführungen für Dokumente im Zusammenhang mit der Behebung.
- `ec2` – Ermöglicht Prinzipalen, Amazon-VPC-Netzwerkressourcen zu beschreiben und zu modifizieren, um Verbindungsprobleme zu beheben. Dies umfasst:
  - Beschreiben von Amazon-VPC-Attributen, Subnetzen, Amazon-VPC-Endpunkten und Sicherheitsgruppen.

  - Erstellen von Amazon-VPC-Endpunkten für Systems-Manager-Services (`ssm`, `ssmmessages` und `ec2messages`) mit den erforderlichen Tags.
  - Ändern der Amazon-VPC-Attribute, um DNS-Unterstützung und Hostnamen zu aktivieren.
  - Erstellen und Verwalten von Sicherheitsgruppen mit spezifischen Tags für den Zugriff auf Amazon-VPC-Endgeräte.
  - Autorisierung und Widerruf von Sicherheitsgruppenregeln für den HTTPS-Zugriff mit entsprechenden Tags.
  - Erstellen von Tags auf Amazon-VPC-Endpunkten, Sicherheitsgruppen und Sicherheitsgruppenregeln während der Ressourcenerstellung.
- `kms` – Ermöglicht Prinzipalen die Verwendung von kundenspezifischen AWS Key Management Service -Schlüsseln für die Entschlüsselung und Datenschlüsselgenerierung beim Zugriff auf verschlüsselte Objekte in Amazon-S3-Buckets, die für Behebungsvorgänge verwendet werden. Diese Berechtigungen sind auf Schlüssel beschränkt, die mit `SystemsManagerManaged` gekennzeichnet sind und mit dem Amazon-S3-Service verwendet werden, mit spezifischen Anforderungen an den Verschlüsselungskontext.
- `iam` – Ermöglicht Prinzipalen, die Ausführungsrolle für die Problembehebung an Systems Manager zu übergeben, um Automation-Runbooks auszuführen. Diese Berechtigung ist auf Rollen mit dem `AWS-SSM-RemediationExecutionRole`-Benennungsmuster beschränkt und kann nur an den Systems-Manager-Service weitergegeben werden.

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWS-SSM- RemediationAutomation — ExecutionRolePolicy im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: Einrichtung AWSQuick SSMManage ResourcesExecutionPolicy

Diese Richtlinie gewährt Quick Setup Berechtigungen, mit denen das `AWSQuickSetupType-SSM-SetupResources`-Automation-Runbook ausgeführt werden kann. Dieses Runbook erstellt IAM-Rollen für Quick Setup-Zuordnungen, die wiederum durch eine `AWSQuickSetupType-SSM`-Bereitstellung erstellt werden. Es gewährt auch Berechtigungen zum Bereinigen eines zugehörigen Amazon-S3-Buckets während eines Quick Setup-Löschvorgangs.

Sie können die Richtlinie zu Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie auch einer Servicerolle zu, mit der Sie Systems Manager in Ihrem Namen ausführen können.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `iam` – Ermöglicht Prinzipalen das Auflisten und Verwalten von IAM-Rollen zur Verwendung mit Quick Setup-Systems-Manager-Explorer-Vorgängen sowie das Anzeigen, Anhängen und Trennen von IAM-Richtlinien für die Verwendung mit Quick Setup und Systems Manager Explorer. Diese Berechtigungen sind erforderlich, damit Quick Setup Rollen erstellen kann, die für einige seiner Konfigurationsvorgänge erforderlich sind.
- `s3` – Ermöglicht Prinzipalen das Abrufen von Informationen über Objekte in Amazon-S3-Buckets und das Löschen von Objekten aus Amazon-S3-Buckets im Prinzipalkonto, die speziell für Quick Setup-Konfigurationsvorgänge verwendet werden. Dies ist erforderlich, damit S3-Objekte, die nach der Konfiguration nicht mehr benötigt werden, entfernt werden können.

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetup SSMManage ResourcesExecutionPolicy im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSQuick Einrichtung SSMLifecycle ManagementExecutionPolicy

Die `AWSQuickSetupSSMLifecycleManagementExecutionPolicy` Richtlinie gewährt Administratorberechtigungen, die es Quick Setup ermöglichen, eine CloudFormation benutzerdefinierte Ressource bei Lebenszyklusereignissen während der Quick Setup Bereitstellung in auszuführenSystems Manager.

Sie können diese Richtlinie zu Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie auch einer Servicerolle zu, mit der Systems Manager Korrekturmaßnahmen in Ihrem Namen ausführen kann.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm` – Ermöglicht Prinzipalen, Informationen über Automatisierungsausführungen abzurufen und Automatisierungsausführungen für die Einrichtung bestimmter Quick Setup-Operationen zu starten.
- `iam` – Ermöglicht Prinzipalen, Rollen von IAM für die Einrichtung bestimmter Quick Setup-Ressourcen zu übergeben.

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetup SSMLifecycle ManagementExecutionPolicy im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSQuick Einrichtung SSMDeployment RolePolicy

Die verwaltete Richtlinie `AWSQuickSetupSSMDeploymentRolePolicy` gewährt Administratorberechtigungen, mit denen Quick Setup-Ressourcen erstellt werden können, die während des Systems-Manager-Onboarding-Prozesses verwendet werden.

Obwohl Sie diese Richtlinie manuell an Ihre IAM-Entitäten anhängen können, wird dies nicht empfohlen. Quick Setup erstellt Entitäten, die diese Richtlinie an eine Servicerolle anhängen, die es Systems Manager ermöglicht, Aktionen in Ihrem Namen auszuführen.

Diese Richtlinie hat nichts mit der `SSMQuickSetupRolePolicy`-Richtlinie zu tun, die verwendet wird, um Berechtigungen für die servicebezogene Rolle `AWSServiceRoleForSSMQuickSetup` bereitzustellen.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm`— Ermöglicht es Prinzipalen, Zuordnungen für bestimmte Ressourcen zu verwalten, die mithilfe von AWS CloudFormation Vorlagen und einem bestimmten Satz von SSM-Dokumenten erstellt wurden, Rollen und Rollenrichtlinien zu verwalten, die zur Diagnose und Behebung verwalteter Knoten mithilfe von CloudFormation Vorlagen verwendet werden, und Richtlinien für Lebenszyklusereignisse anzuhängen und zu löschen Quick Setup
- `iam` – Ermöglicht Prinzipalen die Kennzeichnung von Tags und Weitergabe von Rollenberechtigungen für den Systems-Manager-Service und den Lambda-Service sowie die Weitergabe von Rollenberechtigungen für Diagnosevorgänge.
- `lambda`— Ermöglicht es Prinzipalen, Funktionen für den Quick Setup Lebenszyklus im Hauptkonto mithilfe von Vorlagen zu kennzeichnen und zu verwalten. CloudFormation
- `cloudformation`— Ermöglicht es Prinzipalen, Informationen von zu lesen. CloudFormation Dies ist erforderlich, um Daten über die CloudFormation Stacks zu sammeln, die zur Verwaltung des Ressourcenstatus und der CloudFormation Stackset-Operationen verwendet werden. Quick Setup

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetup SSMDeployment RolePolicy im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: SSMDeployment S3 AWSQuick einrichten BucketRolePolicy

Die `AWSQuickSetupSSMDeploymentS3BucketRolePolicy`-Richtlinie gewährt Berechtigungen zum Auflisten aller S3-Buckets in einem Konto sowie zum Verwalten und Abrufen von Informationen zu bestimmten Buckets im Hauptkonto, die über CloudFormation -Vorlagen verwaltet werden.

Sie können `AWSQuickSetupSSMDeploymentS3BucketRolePolicy` zu Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie auch einer Servicerolle zu, mit der Sie Systems Manager in Ihrem Namen ausführen können.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `s3`— Ermöglicht Principals, alle S3-Buckets in einem Konto aufzulisten und Informationen zu bestimmten Buckets im Hauptkonto zu verwalten und abzurufen, die über Vorlagen verwaltet werden. CloudFormation

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetup SSMDeployment S3 BucketRolePolicy im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS Von verwaltete Richtlinie: AWSQuickSetupEnableDHMCExecutionPolicy

Diese Richtlinie gewährt Administratorberechtigungen, die es Prinzipalen ermöglichen, das `AWSQuickSetupType-EnableDHMC`-Automation-Runbook auszuführen, wodurch die Standard-Hostverwaltungskonfiguration aktiviert wird. Die Einstellung Standard-Host-Management-Konfiguration ermöglicht es Systems Manager, EC2 Amazon-Instances automatisch als verwaltete Instances zu verwalten. Eine verwaltete Instanz ist eine EC2 Instanz, die für die Verwendung mit Systems Manager konfiguriert ist. Diese Richtlinie gewährt auch Berechtigungen zum Erstellen von IAM-Rollen, die in den Systems-Manager-Serviceeinstellungen als Standardrollen für SSM Agent angegeben sind.

Sie können `AWSQuickSetupEnableDHMCExecutionPolicy` zu Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie auch einer Servicerolle zu, mit der Sie Systems Manager in Ihrem Namen ausführen können.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm` – Ermöglicht Prinzipalen das Aktualisieren und Abrufen von Informationen zu den Systems-Manager-Serviceeinstellungen.
- `iam` – Ermöglicht Prinzipalen das Erstellen und Abrufen von Informationen zu IAM-Rollen für Quick Setup-Operationen.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetupEnableDHMCExecutionPolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS Von verwaltete Richtlinie: AWSQuickSetupEnableAREXExecutionPolicy

Diese Richtlinie gewährt Administratorberechtigungen, die es Systems Manager ermöglichen, das `AWSQuickSetupType-EnableAREX`-Automation-Runbook auszuführen, das die Verwendung von AWS Resource Explorer mit Systems Manager ermöglicht. Resource Explorer ermöglicht das Anzeigen von Ressourcen in Ihrem Konto mit einer Suchfunktion, die einer Internet-Suchmaschine ähnelt. Die Richtlinie gewährt auch Berechtigungen für die Verwaltung von Resource Explorer-Indizes und -Ansichten.

Sie können `AWSQuickSetupEnableAREXExecutionPolicy` zu Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie auch einer Servicerolle zu, mit der Sie Systems Manager in Ihrem Namen ausführen können.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `iam`— Ermöglicht Prinzipalen das Erstellen einer dienstbezogenen Rolle im AWS Identity and Access Management (IAM-) Dienst.
- `resource-explorer-2` – Ermöglicht Prinzipalen das Abrufen von Informationen über Resource-Explorer-Ansichten und -Indizes, das Erstellen von Resource-Explorer-Ansichten und -Indizes sowie das Ändern des Indextyps für Indizes, die in Quick Setup angezeigt werden.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetupEnableAREXExecutionPolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSQuick SetupManagedInstanceProfileExecutionPolicy

Diese Richtlinie gewährt Administratorberechtigungen, die es Systems Manager ermöglichen, ein Standard-IAM-Instance-Profil für das Quick Setup Tool zu erstellen und es an EC2 Amazon-Instances anzuhängen, denen noch kein Instance-Profil angehängt ist. Die Richtlinie gewährt auch Systems Manager die Möglichkeit, Berechtigungen an bestehende Instance-Profile anzuhängen. Auf diese Weise wird sichergestellt, dass die für Systems Manager die Kommunikation mit SSM Agent EC2 Instances erforderlichen Berechtigungen vorhanden sind.

Sie können `AWSQuickSetupManagedInstanceProfileExecutionPolicy` zu Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie auch einer Servicerolle zu, mit der Sie Systems Manager in Ihrem Namen ausführen können.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm` – Ermöglicht Prinzipalen das Starten von Automatisierungsworkflows, die mit Quick Setup-Prozessen verknüpft sind.
- `ec2`— Ermöglicht Principals, IAM-Instanzprofile an EC2 Instanzen anzuhängen, die von verwaltet werden. Quick Setup
- `iam` – Ermöglicht Prinzipalen das Erstellen, Aktualisieren und Abrufen von Informationen über Rollen aus IAM, die in Quick Setup-Prozessen verwendet werden; das Erstellen von IAM-Instance-Profilen; das Anhängen der `AmazonSSMManagedInstanceCore`-verwalteten Richtlinie an IAM-Instance-Profile.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetupManagedInstanceProfileExecutionPolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSQuick SetupManage JITNAResources ExecutionPolicy

Die verwaltete Richtlinie `AWSQuickSetupManageJITNAResourcesExecutionPolicy` ermöglicht esQuick Setup, einem Tool in Systems Manager, den just-in-time Knotenzugriff einzurichten.

Sie können `AWSQuickSetupManageJITNAResourcesExecutionPolicy` zu Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie auch einer Servicerolle zu, mit der Sie Systems Manager in Ihrem Namen ausführen können.

Diese Richtlinie gewährt Administratorberechtigungen, mit denen Ressourcen Systems Manager für den just-in-time Knotenzugriff erstellt werden können.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm`— Ermöglicht Prinzipalen, die Diensteinstellung abzurufen und zu aktualisieren, die den Identitätsanbieter für den just-in-time Knotenzugriff angibt.
- `iam`— Ermöglicht es Prinzipalen, Rollen zu erstellen, zu kennzeichnen und abzurufen, Rollenrichtlinien für verwaltete Richtlinien für den just-in-time Knotenzugriff anzuhängen und dienstbezogene Rollen für den just-in-time Knotenzugriff und Benachrichtigungen zu erstellen.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetupManageJITNAResourcesExecutionPolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: Einrichtung AWSQuick JITNADeployment RolePolicy

Die verwaltete Richtlinie `AWSQuickSetupJITNADeploymentRolePolicy` ermöglicht Quick Setup die Bereitstellung des Konfigurationstyps, der für die Einrichtung des just-in-time Knotenzugriffs erforderlich ist.

Sie können `AWSQuickSetupJITNADeploymentRolePolicy` zu Ihren IAM-Entitäten anfügen. Systems Manager fügt diese Richtlinie auch einer Servicerolle zu, mit der Sie Systems Manager in Ihrem Namen ausführen können.

Diese Richtlinie gewährt Administratorberechtigungen, mit denen Ressourcen Systems Manager für den just-in-time Knotenzugriff erstellt werden können.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `cloudformation`— Ermöglicht Prinzipalen das Erstellen, Aktualisieren, Löschen und Lesen von CloudFormation Stacks.

- `ssm`— Ermöglicht Prinzipalen das Erstellen, Löschen, Aktualisieren und Lesen von State Manager Verknüpfungen, die von aufgerufen werden. CloudFormation
- `iam`— Ermöglicht Prinzipalen das Erstellen, Löschen, Lesen und Markieren von IAM-Rollen, die von aufgerufen werden. CloudFormation

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSQuickSetup JITNADeployment RolePolicy im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSSystems ManagerJustInTimeAccessServicePolicy

Die verwaltete Richtlinie `AWSSystemsManagerJustInTimeAccessServicePolicy` ermöglicht den Zugriff auf AWS Ressourcen, die vom AWS Systems Manager just-in-time Access-Framework verwaltet oder genutzt werden. Dieses Richtlinienupdate fügt Tagging-Berechtigungen für die automatische Ausführung hinzu, sodass Kunden die Bedienerberechtigungen auf bestimmte Tags einschränken können.

Sie können `AWSSystemsManagerJustInTimeAccessServicePolicy` nicht an Ihre IAM-Entitäten anhängen. Diese Richtlinie ist an eine servicegebundene Rolle angehängt, die Systems Manager die Durchführung von Aktionen in Ihrem Namen ermöglicht. Weitere Informationen finden Sie unter Verwenden von Rollen zum Aktivieren des just-in-time Knotenzugriffs.

Diese Richtlinie gewährt Administratorberechtigungen, die den Zugriff auf Ressourcen ermöglichen, die mit dem just-in-time Knotenzugriff verknüpft sind.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm` – Ermöglicht Prinzipalen das Erstellen und VerwaltenOpsItems, Hinzufügen von Tags zu OpsItems und das Automatisieren von Ausführungen, das Abrufen und Aktualisieren von OpsItems, Abrufen und Beschreiben von Dokumenten, Beschreiben von OpsItems und Sitzungen, das Auflisten von Dokumenten und Tags für verwaltete Instances.
- `ssm-guiconnect` – Ermöglicht Prinzipalen das Auflisten von Verbindungen.
- `identitystore`— Ermöglicht es Prinzipalen, Benutzer und Gruppen abzurufen IDs, Benutzer zu beschreiben und Gruppenmitgliedschaften aufzulisten.
- `sso-directory` – Ermöglicht Prinzipalen, Benutzer zu beschreiben und festzustellen, ob ein Benutzer ein Mitglied einer Gruppe ist.

- `sso` – Ermöglicht Prinzipalen, registrierte Regionen zu beschreiben und Instances und Verzeichniszuordnungen aufzulisten.
- `cloudwatch` – Ermöglicht Prinzipalen, Metrikdaten für den `AWS/SSM/JustInTimeAccess`-Namespace einzugeben.
- `ec2` – Ermöglicht Prinzipalen die Beschreibung von Tags.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSSystemsManagerJustInTimeAccessServicePolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSSystems ManagerJustInTimeAccessTokenPolicy

Die verwaltete Richtlinie `AWSSystemsManagerJustInTimeAccessTokenPolicy` gewährt Benutzern die Erlaubnis, sichere Verbindungen zu EC2 Amazon-Instances und verwalteten Instances über Session Manager RDP-Verbindungen von Systems Manager GUI Connect als Teil von Workflows für den just-in-time Knotenzugriff herzustellen.

Sie können `AWSSystemsManagerJustInTimeAccessTokenPolicy` an Ihre IAM-Entitäten anhängen.

Diese Richtlinie gewährt Mitwirkenden Berechtigungen, die es Benutzern ermöglichen, sichere Sitzungen zu starten und zu verwalten, RDP-Verbindungen herzustellen und die für den Knotenzugriff erforderlichen kryptografischen Operationen durchzuführen. just-in-time

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm`— Ermöglicht Principals, Session Manager Sitzungen auf EC2 Amazon-Instances und verwalteten Instances mithilfe des SSM-Dokuments zu starten. SessionManagerRunShell Ermöglicht auch das Beenden und Wiederaufnehmen von Sitzungen, das Abrufen von Befehlsaufrufdetails und das Senden von Befehlen an Instanzen für die SSO-Benutzereinrichtung, wenn sie über Systems Manager GUI Connect aufgerufen werden. Ermöglicht außerdem das Starten von Portweiterleitungssitzungen für RDP-Verbindungen, wenn sie über Systems Manager GUI Connect aufgerufen werden.
- `ssmmessages`— Ermöglicht Prinzipalen das Öffnen von Datenkanälen für eine sichere Kommunikation während Session Manager Sitzungen.

- `ssm-guiconnect`— Ermöglicht Prinzipalen, Systems Manager GUI Connect RDP-Verbindungen zu Instanzen zu starten, Details zu diesen abzurufen und abzubrechen.
- `kms`— Ermöglicht Prinzipalen das Generieren von Datenschlüsseln für die Session Manager Verschlüsselung und das Erstellen von Zuweisungen für RDP-Verbindungen. Diese Berechtigungen sind auf AWS KMS Schlüssel beschränkt, die mit gekennzeichnet sind. `SystemsManagerJustInTimeNodeAccessManaged=true` Die Erstellung von Zuschüssen ist außerdem darauf beschränkt, dass sie nur über den Systems Manager GUI Connect-Dienst verwendet werden kann.
- `sso`— Ermöglicht Prinzipalen, Verzeichniszuordnungen aufzulisten, wenn sie über Systems Manager GUI Connect aufgerufen werden. Dies ist für die RDP-SSO-Benutzereinrichtung erforderlich.
- `identitystore`— Ermöglicht Prinzipalen, Benutzer im Identitätsspeicher zu beschreiben, wenn sie über Systems Manager GUI Connect aufgerufen werden. Dies ist für die Einrichtung von RDP-SSO-Benutzern erforderlich.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSSystemsManagerJustInTimeAccessTokenPolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSSystems ManagerJustInTimeAccessTokenSessionPolicy

Die verwaltete Richtlinie `AWSSystemsManagerJustInTimeAccessTokenSessionPolicy` ermöglicht die Anwendung von Berechtigungen Systems Manager mit eingeschränktem Geltungsbereich auf ein just-in-time Knotenzugriffstoken.

Sie können `AWSSystemsManagerJustInTimeAccessTokenSessionPolicy` an Ihre IAM-Entitäten anhängen.

Diese Richtlinie gewährt Administratorberechtigungen, die es ermöglichen, Berechtigungen für just-in-time Knotenzugriffstoken nach unten Systems Manager zu beschränken.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm` – Ermöglicht es den Prinzipalen, Session Manager-Sitzungen mit dem `SSM-SessionManagerRunShell`-Dokument zu starten. Auch beim ersten Aufruf über `ssm-`

guiconnect können Sitzungen mithilfe des AWS-StartPortForwardingSession-Dokuments gestartet, Befehlsaufrufe aufgelistet und Befehle mithilfe des AWSSSO-CreateSSOUser-Dokuments gesendet werden.

- ssm-guiconnect – Ermöglicht Prinzipalen das Abbrechen, Abrufen und Starten von Verbindungen zu allen Ressourcen.
- kms— Ermöglicht es Prinzipalen, Berechtigungen zu erstellen und Datenschlüssel für Schlüssel zu generieren, die mit markiert sindSystemsManagerJustInTimeNodeAccessManaged, wenn sie ssm-guiconnect über einen AWS Dienst aufgerufen werden.
- sso – Ermöglicht Prinzipalen, Verzeichniszuordnungen aufzulisten, wenn sie über ssm-guiconnect aufgerufen werden.
- identitystore – Ermöglicht es den Prinzipalen, einen Benutzer zu beschreiben, wenn er über ssm-guiconnect aufgerufen wird.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSSystemsManagerJustInTimeAccessTokenSessionPolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSSystems ManagerJustInTimeNodeAccessRolePropagationPolicy

Die verwaltete Richtlinie AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy ermöglicht Systems Manager die gemeinsame Nutzung von Zugriffsverweigerungsrichtlinien aus dem delegierten Administratorkonto für Mitgliedskonten und die Replikation der Richtlinien in mehreren AWS-Regionen.

Sie können AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy an Ihre IAM-Entitäten anhängen.

Diese Richtlinie bietet die Administratorberechtigungen, die für das Teilen und die Erstellung von Zugriffsverweigerungsrichtlinien durch Systems Manager erforderlich sind. Dadurch wird sichergestellt, dass Richtlinien zur Zugriffsverweigerung auf alle Konten in einer AWS Organizations Organisation und in Regionen angewendet werden, die für den just-in-time Knotenzugriff konfiguriert sind.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ssm` – Ermöglicht Prinzipalen die Verwaltung von SSM-Dokumenten und Ressourcenrichtlinien.
- `ssm-quicksetup` – Ermöglicht Prinzipalen das Lesen von Quick Setup-Konfigurationsmanagern.
- `organizations` – Ermöglicht Prinzipalen, Details über eine AWS Organizations -Organisation und delegierte Administratoren aufzulisten.
- `ram` – Ermöglicht Prinzipalen das Erstellen, Markieren und Beschreiben von Ressourcenfreigaben.
- `iam` – Ermöglicht Prinzipalen, eine Servicerolle zu beschreiben.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWSSystems ManagerNotificationsServicePolicy

Die verwaltete Richtlinie `AWSSystemsManagerNotificationsServicePolicy` ermöglicht Systems Manager das Senden von E-Mail-Benachrichtigungen für just-in-time Knotenzugriffsanfragen an Genehmiger von Zugriffsanfragen.

Sie können `AWSSystemsManagerJustInTimeAccessServicePolicy` nicht an Ihre IAM-Entitäten anhängen. Diese Richtlinie ist an eine servicegebundene Rolle angehängt, die Systems Manager die Durchführung von Aktionen in Ihrem Namen ermöglicht. Weitere Informationen finden Sie unter Verwenden von Rollen zum Senden von Benachrichtigungen über just-in-time Knotenzugriffsanfragen.

Diese Richtlinie gewährt Administratorberechtigungen, die es ermöglichenSystems Manager, E-Mail-Benachrichtigungen für just-in-time Knotenzugriffsanfragen an Genehmiger von Zugriffsanfragen zu senden.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `identitystore` – Ermöglicht Prinzipalen, Benutzer und Gruppenmitgliedschaften aufzulisten und zu beschreiben.
- `sso` – Ermöglicht Prinzipalen, Instances und Verzeichnisse aufzulisten und registrierte Regionen zu beschreiben.
- `sso-directory` – Ermöglicht Prinzipalen, Benutzer zu beschreiben und Mitglieder einer Gruppe aufzulisten.

- `iam` – Ermöglicht Prinzipalen das Abrufen von Informationen über Rollen.

Weitere Einzelheiten zu dieser Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWSSystemsManagerNotificationsServicePolicy im AWS Referenzhandbuch für verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWS-SSM-Automation-DiagnosisBucketPolicy

Die verwaltete Richtlinie `AWS-SSM-Automation-DiagnosisBucketPolicy` bietet Berechtigungen für die Diagnose von Problemen mit Knoten, die mit AWS Systems Manager Diensten interagieren, indem sie den Zugriff auf S3-Buckets ermöglicht, die zur Diagnose und Behebung von Problemen verwendet werden.

Sie können die `AWS-SSM-Automation-DiagnosisBucketPolicy`-Richtlinie an Ihre IAM-Identitäten anfügen. Systems Manager fügt diese Richtlinie auch einer IAM-Rolle zu, mit der Systems Manager in Ihrem Namen Diagnoseaktionen ausführen kann.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `s3` – Ermöglicht Prinzipalen, auf Objekte zuzugreifen und diese in einen Amazon-S3-Bucket zu schreiben.

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWS-SSM-Automation- DiagnosisBucketPolicy im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWS-SSM-RemediationAutomation-OperationalAccountAdministrationRolePolicy

Die verwaltete Richtlinie `AWS-SSM-RemediationAutomation-OperationalAccountAdministrationRolePolicy` stellt Berechtigungen für ein Betriebskonto bereit, um Probleme mit Knoten zu diagnostizieren, indem sie organisationsspezifische Berechtigungen bereitstellt.

Sie können die `AWS-SSM-RemediationAutomation-OperationalAccountAdministrationRolePolicy` an Ihre IAM-Identitäten anfügen. Systems

Manager fügt diese Richtlinie auch einer IAM-Rolle zu, mit der Systems Manager in Ihrem Namen Diagnoseaktionen ausführen kann.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `organizations` – Ermöglicht Prinzipalen, ein Stammverzeichnis der Organisation aufzulisten und über Mitgliedskonten die Zielkonten zu bestimmen.
- `sts` – Ermöglicht Prinzipalen, Rollen zur Ausführung von Problembehebungen zu übernehmen, um SSM-Automation-Dokumente konto- und regionsübergreifend innerhalb derselben Organisation auszuführen.

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWS-SSM- RemediationAutomation — OperationalAccountAdministrationRolePolicy im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: AWS-SSM-DiagnosisAutomation-OperationalAccountAdministrationRolePolicy

Die verwaltete Richtlinie `AWS-SSM-DiagnosisAutomation-OperationalAccountAdministrationRolePolicy` stellt Berechtigungen für ein Betriebskonto bereit, um Probleme mit Knoten zu diagnostizieren, indem sie organisationsspezifische Berechtigungen bereitstellt.

Sie können die `AWS-SSM-DiagnosisAutomation-OperationalAccountAdministrationRolePolicy`-Richtlinie an Ihre IAM-Identitäten anfügen. Systems Manager fügt diese Richtlinie auch einer IAM-Rolle zu, mit der Systems Manager in Ihrem Namen Diagnoseaktionen ausführen kann.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `organizations` – Ermöglicht Prinzipalen, ein Stammverzeichnis der Organisation aufzulisten und über Mitgliedskonten die Zielkonten zu bestimmen.
- `sts` – Ermöglicht Prinzipalen, Rollen bei der Ausführung von Diagnosen zu übernehmen, um SSM-Automation-Dokumente konto- und regionsübergreifend innerhalb derselben Organisation auszuführen.

Weitere Informationen zur Richtlinie, einschließlich der neuesten Version des JSON-Richtliniendokuments, finden Sie unter AWS-SSM- DiagnosisAutomation — OperationalAccountAdministrationRolePolicy im Referenzhandbuch für AWS verwaltete Richtlinien.

Systems ManagerAktualisierungen der verwalteten Richtlinien AWS

In der folgenden Tabelle finden Sie Details zu Aktualisierungen AWS verwalteter Richtlinien Systems Manager seit Beginn der Erfassung dieser Änderungen durch diesen Dienst am 12. März 2021. Informationen zu anderen verwalteten Richtlinien für den Systems-Manager-Service finden Sie unter Zusätzliche verwaltete Richtlinien für Systems Manager weiter unten in diesem Thema. Um automatische Warnungen über Änderungen an dieser Seite erhalten, abonnieren Sie den RSS-Feed auf der Systems Manager Dokumentverlauf-Seite.

Änderungen	Beschreibung	Datum
AWSSystemsManagerJustInTimeAccessTokenPolicy – Aktualisierung auf eine bestehende Richtlinie	Systems Managerhat die verwaltete Richtlinie aktualisiert`AWSSystemsManagerJustInTimeAccessTokenPolicy` . Die Anweisung (`SID`) `TerminateAndResumeSession` wurde in umbenannt `TerminateAndResumeSessionAndOpenDataChannel` und umfasst nun die `ssmmessages:OpenDataChannel` Aktion, die Sitzungsverwaltung und Datenkanalberechtigungen in einer einzigen Anweisung kombiniert.	25. September 2025
Aktualisierte verwaltete Richtlinien:	Systems Managerdrei verwaltete Richtlinien	12. September 2025

Änderungen	Beschreibung	Datum
• AWSQuickEinrichtung SSMLifecycle Managemen tExecutionPolicy • AWSQuickSetupPatch PolicyPermissionsBoundary • AWSQuickSetupManag edInstanceProfileE xecutionPolicy	wurden aktualisiert, um Unterstützung für das Starten von Automatisierungsau sführungen auf zusätzlichen Systems Manager Manager-Ressourcen hinzuzufügen, einschließlich bestimmter Automations-Runbooks und SSM-Befehlsdokumente.	
AWSQuickSetupStart StopInstancesExecutionPolic y— Die verwaltete Richtlinie wurde aktualisiert	Systems ManagerDie verwaltete Richtlinie wurde aktualisiert, um die Berechtig ungen für die Quick Setup Scheduler-Konfiguration zu verfeinern. Die Richtlinie bietet jetzt spezifischere Berechtig ungen für das Starten und Stoppen von EC2 Amazon-Instances, den Zugriff auf Änderungskalender und die Ausführung von Automatis ierungsdokumenten mit verbesserten Sicherhei tsbedingungen.	12. September 2025

Änderungen	Beschreibung	Datum
AWSQuickSetupStartSSMAssociationsExecutionPolicy— Die verwaltete Richtlinie wurde aktualisiert	Systems Managerhat die verwaltete Richtlinie aktualisiert, um das Automatisierungsdokument von `AWSQuickSetupType-StartSSMAssociations` zu zu ändern`AWSQuickSetupType-Scheduler-ChangeCalendarState` . Mit diesem Update wird der Zweck der Richtlinie geändert: Statt SSM-Verknüpfungen zu starten, wird nun der Status des Änderungskalenders für geplante Operationen verwaltet.	12. September 2025
SSMAutomationAmazon-Rolle — Aktualisierung einer bestehenden Richtlinie	Systems Manager hat neue Berechtigungen hinzugefügt, damit Automation-Runbooks Kommunikationskanäle für sitzungsbasierte Vorgänge einrichten können. Die `ssmmessages:OpenDataChannel` -Berechtigung für die Ressource `arn:*:ssm:*:*:session/*` wurde hinzugefügt.	11. September 2025

Änderungen	Beschreibung	Datum
AWSSystemsManagerJustInTimeAccessServicePolicy— Die verwaltete Richtlinie wurde aktualisiert	Systems Manager hat die verwaltete Richtlinie aktualisiert, um Tagging-Berechtigungen für die Automatisierungsausführung hinzuzufügen. Der Service muss Automatisierungsausführungen mit einem `SystemsManagerJustInTimeNodeAccessManaged=true` -Tag kennzeichnen, damit Kunden die Bedienerberechtigungen auf bestimmte Tags beschränken können.	25. August 2025
AWSQuickSetupStartSSMAssociationsExecutionPolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, um Quick Setup die Ausführung des `AWSQuickSetupType-StartSSMAssociations` -Automation-Runbooks zu erlauben. Dieses Runbook wird verwendet, um State Manager-Zuordnungen zu starten, die durch Quick Setup-Konfigurationen erstellt werden.	12. August 2025

Änderungen	Beschreibung	Datum
AWSQuickSetupStartStopInstancesExecutionPolicy – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügt, die es ermöglichtQuick Setup, EC2 Amazon-Instances nach einem Zeitplan zu starten und zu beenden. Diese Richtlinie bietet die erforderlichen Berechtigungen für den Quick Setup-Scheduler-Konfigurationstyp, um den Instance-Status auf der Grundlage definierter Zeitpläne zu verwalten.	12. August 2025
AWSQuickSetupDeploymentRolePolicy— Aktualisierung der Dokumentation	Systems Manager hat die `AWSQuickSetupDeploymentRolePolicy` - verwaltete Richtlinie aktualisiert, um Berechtigungen für zusätzliche Ressourcen zu gewähren. Darüber hinaus wurde die Dokumentation für `AWSQuickSetupDeploymentRolePolicy` aktualisiert und enthält detailliertere Beschreibungen der Berechtigungen, die im Rahmen dieser Richtlinie für Quick Setup-Konfigurationsverwaltungsvorgänge gewährt werden.	12. August 2025

Änderungen	Beschreibung	Datum
AWS-SSM- RemediationAutomation - Execution RolePolicy — Aktualisierung einer bestehenden Richtlinie	Systems ManagerDie verwaltete Richtlinie wurde aktualisiert, um die Sicherheitslage der StartAutomationExecution SSM:API zu verbessern, indem Berechtigungen sowohl für die Ressourcentypen „Dokument“ als auch für „Automationsausführung“ erforderlich sind. Die aktualisierte Richtlinie bietet umfassendere und detailliertere Berechtigungen für die automatische Ausführung von Problembehebungsmaßnahmen, einschließlich verbesserter Beschreibungen für Netzwerkabhilfefunktionen, spezifischerer Berechtigungen zur Erstellung von Amazon VPC-Endpunkten, detailliertere Berechtigungen zur Verwaltung von Sicherheitsgruppen und verbesserte Kontrollen zur Ressourcenmarkierung für Behebungsvorgänge.	16. Juli 2025

Änderungen	Beschreibung	Datum
AWS-SSM- RemediationAutomation - AdministrationRolePolicy — Aktualisierung einer bestehenden Richtlinie	Systems Manager hat die verwaltete Richtlinie aktualisiert, um Verbesserungen der API-Autorisierung für automatische Behebungsvorgänge zu unterstützen. Die aktualisierte Richtlinie erweitert die Berechtigungen für die Ausführung von Aktivitäten, die in Automatisierungsdokumenten definiert sind, und bietet verbesserte Sicherheitskontrollen und Ressourcenzugriffsmuster für Korrektur-Workflows.	16. Juli 2025
AWS-SSM- DiagnosisAutomation - ExecutionRolePolicy — Aktualisierung einer bestehenden Richtlinie	Systems Manager hat die verwaltete Richtlinie aktualisiert, um detailliertere und genauere Berechtigungen für die automatische Ausführung von Diagnosen bereitzustellen. Die aktualisierte Richtlinie umfasst erweiterte Beschreibungen für den Zugriff auf Amazon EC2 - und Amazon VPC-Ressourcen, spezifischere SSM-Automatisierungsberechtigungen AWS KMS sowie verbesserte IAM-Berechtigungsbeschreibungen mit entsprechenden Ressourceneinschränkungen.	16. Juli 2025

Änderungen	Beschreibung	Datum
AWS-SSM- Diagnosis Automation - AdministrationRolePolicy — Aktualisierung einer bestehenden Richtlinie	Systems Manager hat die verwaltete Richtlinie aktualisiert, um spezifischere Berechtigungen und Sicherheitsbedingungen für Vorgänge zur Automatisierung von Diagnosen bereitzustellen. Die aktualisierte Richtlinie bietet erweiterte Sicherheitskontrollen für die AWS KMS Schlüsselnutzung, den Zugriff auf Amazon S3 S3-Buckets und Rollenannahmen mit strengeren ressourcenbasierten Bedingungen und Einschränkungen auf Kontoebene.	16. Juli 2025
AWSQuickSetupDeploymentRolePolicy— Aktualisierung einer Richtlinie	Systems Managerhat der verwalteten Richtlinie Berechtigungen `AWSQuickSetupDeploymentRolePolicy` für den Zugriff auf das Amazon-eigene Runbook AWSQuickSetupType-ManageInstanceProfilehinzugefügt. Diese Berechtigung ermöglicht es Quick Setup, Verknüpfungen mithilfe der verwalteten Richtlinie anstelle von Inline-Richtlinien zu erstellen.	14. Juli 2025

Änderungen	Beschreibung	Datum
Amazon SSMAutomation Role — Aktualisierung der Dokumentation	Systems Manager hat eine umfassende Dokumentation für die bestehende `AmazonSSMAutomationRole` -Richtlinie hinzugefügt, die dem Systems-Manager-Automation-Service Berechtigungen zum Ausführen der innerhalb von Automation-Runbooks definierten Aktivitäten hinzufügt.	15. Juli 2025
AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy— Aktualisierung einer Richtlinie	Systems ManagerEs wurden Berechtigungen hinzugefügt, um Systems Manager das Markieren einer Ressource zu ermöglichen, die AWS Resource Access Manager für den just-in-time Knotenzugriff gemeinsam genutzt wird.	30. April 2025
AWSQuickSetupManageJITNAResourcesExecutionPolicy— Aktualisierung einer Richtlinie	Systems ManagerEs wurden Berechtigungen hinzugefügt, um das Markieren von IAM-Rollen Systems Manager zu ermöglichen, die für den just-in-time Knotenzugriff erstellt wurden.	30. April 2025

Änderungen	Beschreibung	Datum
AWSSystemsManagerJustInTimeAccessTokenSessionPolicy – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügt, die es ermöglicht, Berechtigungen Systems Manager mit eingeschränktem Geltungsbereich auf ein just-in-time Knotenzugriffstoken anzuwenden.	30. April 2025
AWSSystemsManagerNotificationsServicePolicy – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügt, die es ermöglichtSystems Manager, E-Mail-Benachrichtigungen für just-in-time Knotenzugriffsanfragen an Genehmiger zu senden.	30. April 2025
AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, um Systems Manager zu erlauben, Genehmigungsrichtlinien für verschiedene Regionen zu replizieren.	30. April 2025
AWSSystemsManagerJustInTimeAccessTokenPolicy – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügt, die es ermöglicht, Zugriffstoken Systems Manager zu generieren, die für den just-in-time Knotenzugriff verwendet werden.	30. April 2025

Änderungen	Beschreibung	Datum
AWSSystemsManagerJustInTimeAccessServicePolicy – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügt, um Berechtigungen für AWS Ressourcen bereitzustellen, die von der just-in-time Knotenzugriffsfunktion von Systems Manager verwaltet oder verwendet werden.	30. April 2025
AWSQuickSetupManageJITNAResourcesExecutionPolicy – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügtQuick Setup, die es einem Tool in Systems Manager ermöglicht, die für den just-in-time Knotenzugriff erforderlichen IAM-Rollen zu erstellen.	30. April 2025
AWSQuickEinrichtungJITNADeployment RolePolicy — Neue Richtlinie	Systems ManagerEs wurde eine neue Richtlinie hinzugefügt, die Berechtigungen bereitstellt, mit denen Quick Setup der Konfigurationstyp bereitgestellt werden kann, der für die Einrichtung des just-in-time Knotenzugriffs erforderlich ist.	30. April 2025
AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy— Aktualisierung einer Richtlinie	Systems ManagerEs wurden Berechtigungen hinzugefügt, um Systems Manager das Markieren einer Ressource zu ermöglichen, die AWS Resource Access Manager für den just-in-time Knotenzugriff gemeinsam genutzt wird.	30. April 2025

Änderungen	Beschreibung	Datum
AWSQuickSetupManageJITNAResourcesExecutionPolicy— Aktualisierung einer Richtlinie	Systems ManagerEs wurden Berechtigungen hinzugefügt, um das Markieren von IAM-Rollen Systems Manager zu ermöglichen, die für den just-in-time Knotenzugriff erstellt wurden.	30. April 2025
AWSSystemsManagerJustInTimeAccessTokenSessionPolicy – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügt, die es ermöglicht, Berechtigungen Systems Manager mit eingeschränktem Geltungsbereich auf ein just-in-time Knotenzugriffstoken anzuwenden.	30. April 2025
AWSSystemsManagerNotificationsServicePolicy – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügt, die es ermöglichtSystems Manager, E-Mail-Benachrichtigungen für just-in-time Knotenzugriffsanfragen an Genehmiger zu senden.	30. April 2025
AWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, um Systems Manager zu erlauben, Genehmigungsrichtlinien für verschiedene Regionen zu replizieren.	30. April 2025

Änderungen	Beschreibung	Datum
AWSSystemsManagerJustInTimeAccessTokenPolicy – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügt, die es ermöglicht, Zugriffstoken Systems Manager zu generieren, die für den just-in-time Knotenzugriff verwendet werden.	30. April 2025
AWSSystemsManagerJustInTimeAccessServicePolicy – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügt, um Berechtigungen für AWS Ressourcen bereitzustellen, die von der just-in-time Knotenzugriffsfunktion von Systems Manager verwaltet oder verwendet werden.	30. April 2025
AWSQuickSetupManageJITNAResourcesExecutionPolicy – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügtQuick Setup, die es einem Tool in Systems Manager ermöglicht, die für den just-in-time Knotenzugriff erforderlichen IAM-Rollen zu erstellen.	30. April 2025
AWSQuickEinrichtungJITNADeployment RolePolicy — Neue Richtlinie	Systems ManagerEs wurde eine neue Richtlinie hinzugefügt, die Berechtigungen bereitstellt, mit denen Quick Setup der Konfigurationstyp bereitgestellt werden kann, der für die Einrichtung des just-in-time Knotenzugriffs erforderlich ist.	30. April 2025

Änderungen	Beschreibung	Datum
AWS-SSM-DiagnosisAutomation-OperationalAccountAdministrationRolePolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, die einem Betriebskonto Berechtigungen zur Diagnose von Problemen mit Knoten gewährt, indem organisationsspezifische Berechtigungen bereitgestellt werden.	21. November 2024
AWS-SSM-RemediationAutomation-OperationalAccountAdministrationRolePolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, die einem Betriebskonto Berechtigungen zur Diagnose von Problemen mit Knoten gewährt, indem organisationsspezifische Berechtigungen bereitgestellt werden.	21. November 2024
AWS-SSM-Automation-DiagnosisBucketPolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, um das Starten von Automation-Workflows zu unterstützen, die Probleme mit verwalteten Knoten in bestimmten Konten und Regionen diagnostizieren.	21. November 2024
AmazonSSMServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Systems Managerneue Berechtigungen hinzugefügt, um Details AWS Resource Explorer zu EC2 Amazon-Instances zu sammeln und die Ergebnisse in Widgets im neuen Systems Manager Dashboard anzuzeigen.	21. November 2024

Änderungen	Beschreibung	Datum
SSMQuickSetupRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Systems Manager hat die verwaltete Richtlinie SSMQuickSetupRolePolicy aktualisiert. Diese Aktualisierung ermöglicht es der zugehörigen, mit dem Service verknüpften Rolle AWSServiceRoleForSSMQuickSetup , Ressourcendatensynchronisationen zu verwalten.	21. November 2024
AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, um das Starten von Automation-Workflows zu unterstützen, die Probleme mit verwalteten Knoten in einem Zielkonto und einer Zielregion diagnostizieren.	21. November 2024
AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, um das Starten von Automation-Workflows zu unterstützen, die Probleme mit verwalteten Knoten in einem Zielkonto und einer Zielregion diagnostizieren.	21. November 2024

Änderungen	Beschreibung	Datum
AWS-SSM-RemediationAutomation-AdministrationRolePolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, um das Starten von Automation-Workflows zu unterstützen, die Probleme mit verwalteten Knoten in bestimmten Konten und Regionen diagnostizieren.	21. November 2024
AWS-SSM-RemediationAutomation-ExecutionRolePolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, um das Starten von Automation-Workflows zu unterstützen, die Probleme mit verwalteten Knoten in bestimmten Konten und Regionen diagnostizieren.	21. November 2024
AWSQuickEinrichtungSSMDeployment RolePolicy — Aktualisierung einer Richtlinie	Systems ManagerEs wurden Berechtigungen hinzugefügt, um das Systems Manager Taggen von IAM-Rollen und Lambda zu ermöglichen, die für die vereinheitlichte Konsole erstellt wurden.	7. Mai 2025
AWSQuickSetupSSMManageResourcesExecutionPolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, um die Ausführung eines Vorgangs in Quick Setup zu unterstützen, bei dem IAM-Rollen für Quick Setup-Zuordnungen erstellt werden, die wiederum durch eine `AWSQuickSetupType-SSM` -Bereitstellung erstellt werden.	21. November 2024

Änderungen	Beschreibung	Datum
AWSQuickSetupSSMLifecycleManagementExecutionPolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, um die Quick Setup Ausführung einer CloudFormation benutzerdefinierten Ressource bei Lebenszyklusereignissen während einer Quick Setup Bereitstellung zu unterstützen.	21. November 2024
AWSQuickSetupSSMDeploymentRolePolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, um die Gewährung von Administratorberechtigungen Quick Setup zu unterstützen, mit denen Ressourcen erstellt werden können, die während des Systems-Manager-Onboarding-Prozesses verwendet werden.	21. November 2024
AWSQuickSetupSSMDeploymentS3BucketRolePolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, um die Verwaltung und das Abrufen von Informationen zu bestimmten Buckets im Hauptkonto zu unterstützen, die über Vorlagen verwaltet werden. CloudFormation	21. November 2024

Änderungen	Beschreibung	Datum
AWSQuickSetupEnableDHMCExecutionPolicy – Neue Richtlinie	Systems Manager führt eine neue Richtlinie ein, die es Quick Setup ermöglicht, eine IAM-Rolle zu erstellen , die selbst die vorhandenen AmazonSSMManagedEC2InstanceDefaultPolicy verwendet. Diese Richtlinie enthält alle Berechtigungen, die für die Kommunikation von SSM Agent mit dem Systems-Manager-Service erforderlich sind. Die neue Richtlinie ermöglicht auch Änderungen an den Systems-Manager-Serviceeinstellungen .	21. November 2024
AWSQuickSetupEnableAREXExecutionPolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, mit der eine dienstverknüpfte Rolle für AWS Resource Explorer den Zugriff auf Resource Explorer-Ansichten und Aggregatorindizes erstellt werden kann. Quick Setup	21. November 2024

Änderungen	Beschreibung	Datum
AWSQuickSetupManagedInstanceProfileExecutionPolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, die es ermöglichtQuick Setup, ein Quick Setup Standard-Instance-Profil zu erstellen und es an alle EC2 Amazon-Instances anzuhängen, denen ein zugeordnetes Instance-Profil fehlt. Diese neue Richtlinie ermöglicht Quick Setup auch das Anhängen von Berechtigungen an bestehende Profile, um sicherzustellen, dass alle erforderlichen Systems-Manager-Berechtigungen erteilt wurden.	21. November 2024
SSMQuickSetupRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Systems Managerneue Berechtigungen hinzugefügt, um den Zustand zusätzlicher AWS CloudFormation Stack-Sets, die es erstellt hat, überprüfen zu könnenQuick Setup.	13. August 2024
AmazonSSMManagedEC2InstanceDefaultPolicy – Aktualisierung auf eine bestehende Richtlinie	Systems Managerhat der JSON-Richtlinie für `AmazonSSMManagedEC2InstanceDefaultPolicy` eine Anweisung IDs (Sids) hinzugefügt. Diese SIDs enthalten Inline-Beschreibungen des Zwecks der einzelnen Richtlinienerklärungen.	18. Juli 2024

Änderungen	Beschreibung	Datum
SSMQuickSetupRolePolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, die es Quick Setup ermöglicht, den Zustand der bereitgestellten Ressourcen zu überprüfen und Instances zu korrigieren, die von der ursprünglichen Konfiguration abweichen.	3. Juli 2024
AWSQuickSetupDeploymentRolePolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie zur Unterstützung mehrerer Quick-Setup-Konfigurationstypen hinzugefügt, die IAM-Rollen und Automatisierungen erstellen, die wiederum häufig verwendete Amazon-Web-Services-Services und -Funktionen mit empfohlenen bewährten Methoden konfigurieren.	3. Juli 2024
AWSQuickSetupPatchPolicyDeploymentRolePolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, mit der Quick Setup Ressourcen erstellen kann, die mit Patch Manager-Patch-Richtlinienkonfigurationen Quick Setup verknüpft sind.	3. Juli 2024

Änderungen	Beschreibung	Datum
AWSQuickSetupPatchPolicyBaselineAccess – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, die Quick Setup den Zugriff auf Patch-Baselines in Patch Manager mit schreibgeschützte Berechtigungen ermöglicht.	3. Juli 2024
AWSSystemsManagerEnableExplorerExecutionPolicy – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, mit der Quick Setup Administratorrechte für die Aktivierung von Explorer erteilen kann.	3. Juli 2024
AWSSystemsManagerEnableConfigRecordingExecutionPolicy – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügt, um die AWS Config Konfigurationsaufzeichnung Quick Setup zu aktivieren und zu konfigurieren.	3. Juli 2024
AWSQuickSetupDevOpsGuruPermissionsBoundary – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügt, die es ermöglichtQuick Setup, Amazon DevOps Guru zu aktivieren und zu konfigurieren.	3. Juli 2024
AWSQuickSetupDistributorPermissionsBoundary – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügt, Quick Setup um die Aktivierung und Konfiguration des Tools Distributor in zu ermöglichen AWS Systems Manager.	3. Juli 2024

Änderungen	Beschreibung	Datum
AWSQuickEinrichtungSSMHost MgmtPermissionsBoundary — Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügt, die es ermöglichtQuick Setup, Systems Manager Manager-Tools für die sichere Verwaltung von EC2 Amazon-Instances zu aktivieren und zu konfigurieren.	3. Juli 2024
AWSQuickSetupPatchPolicyPermissionsBoundary – Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, mit der Quick Setup Patch-Richtlinien in Patch Manager aktiviert und konfiguriert, einem Tool in AWS Systems Manager.	3. Juli 2024
AWSQuickSetupSchedulerPermissionsBoundary – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügt, die es Quick Setup ermöglicht, geplante Operationen auf EC2 Amazon-Instances und anderen Ressourcen zu aktivieren und zu konfigurieren.	3. Juli 2024
AWSQuickEinrichtungCFGCPacks PermissionsBoundary — Neue Richtlinie	Systems Manager hat eine neue Richtlinie hinzugefügt, mit der Quick Setup die Bereitstellung von Conformance Packs AWS Config ermöglicht.	3. Juli 2024

Änderungen	Beschreibung	Datum
AWSSystemsManagerOpsDataSyncServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	OpsCenterDie Richtlinie wurde aktualisiert, um die Sicherheit des Servicecodes innerhalb der OpsData dienstbezogenen Rolle für die Verwaltung verwandter Vorgänge Explorer zu verbessern.	03. Juli 2023
AmazonSSMManagedEC2InstanceDefaultPolicy – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügt, um Systems Manager Funktionen auf EC2 Amazon-Instances ohne die Verwendung eines IAM-Instance-Profils zu ermöglichen.	18. August 2022
Amazon SSMService RolePolicy — Aktualisierung einer bestehenden Richtlinie	Systems Managerneue Berechtigungen hinzugefügt, mit denen Explorer Sie eine verwaltete Regel erstellen können, wenn Sie Security Hub CSPM von oder aus Explorer aktivieren. OpsCenter Neue Berechtigungen wurden hinzugefügt, um zu überprüfen, ob die Konfiguration und der Compute-Optimizer die erforderlichen Anforderungen erfüllen, bevor sie zugelassen werden. OpsData	27. April 2021

Änderungen	Beschreibung	Datum
AWSSystemsManagerOpsDataSyncServiceRolePolicy – Neue Richtlinie	Systems Managerhat eine neue Richtlinie hinzugefügt, um die Ergebnisse OpsData von Security Hub CSPM in OpsItems Explorer und zu erstellen und zu aktualisieren. OpsCenter	27. April 2021
AmazonSSMServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Systems Managerneue Berechtigungen hinzugefügt, um die Anzeige von Aggregaten OpsData und OpsItems Details aus mehreren Konten und AWS-Regionen in zu ermöglichen. Explorer	24. März 2021
Systems Manager hat die Änderungsverfolgung gestartet	Systems Managerhat begonnen, Änderungen an den AWS verwalteten Richtlinien zu verfolgen.	12. März 2021

Zusätzliche verwaltete Richtlinien für Systems Manager

Zusätzlich zu den verwalteten Richtlinien, die weiter oben in diesem Thema beschrieben wurden, werden die folgenden Richtlinien auch von Systems Manager unterstützt.

- AmazonSSMAutomationApproverAccess – AWS -verwaltete Richtlinie, die den Zugriff ermöglicht, um Automatisierungsausführungen anzuzeigen und Genehmigungsentscheidungen an Automatisierungen zu senden, die auf Genehmigung warten.
- AmazonSSMDirectoryServiceAccess— AWS verwaltete Richtlinie, die SSM Agent den Zugriff im Namen des Benutzers Directory Service auf Anfragen zum Beitritt zur Domäne durch den verwalteten Knoten ermöglicht.
- AmazonSSMFullAccess— AWS verwaltete Richtlinie, die vollen Zugriff auf die Systems Manager API und Dokumente gewährt.

- AmazonSSMMaintenanceWindowRole – AWS -verwaltete Richtlinie, die Wartungsfenstern Berechtigungen für die Systems-Manager-API gewährt.
- AmazonSSMManagedInstanceCore – Von AWS verwaltete Richtlinie, die einem Knoten die Nutzung von Systems Manager-Servicekern-Funktionalität erlaubt.
- AmazonSSMPatchAssociation – AWS -verwaltete Richtlinie, die den Zugriff auf untergeordnete Instances für Patch-Zuordnungsvorgänge ermöglicht.
- AmazonSSMReadOnlyAccess – AWS -verwaltete Richtlinie, die Zugriff auf schreibgeschützte Systems Manager-API-Vorgänge wie `Get*` und `List*` gewährt.
- AWSSSMOpsInsightsServiceRolePolicy— AWS verwaltete Richtlinie, die Berechtigungen für die Erstellung und Aktualisierung betrieblicher Einblicke OpsItemsin gewährtSystems Manager. Wird verwendet, um Berechtigungen durch die serviceverknüpfte Rolle AWSServiceRoleForAmazonSSM_OpsInsights bereitzustellen.
- AWSSystemsManagerAccountDiscoveryServicePolicy— AWS verwaltete Richtlinie, die Systems Manager die Erlaubnis erteilt, AWS-Konto Informationen zu ermitteln.
- AmazonEC2RoleforSSM – Diese Richtlinie wird nicht mehr unterstützt und sollte nicht verwendet werden. Verwenden Sie stattdessen die `AmazonSSMManagedInstanceCore` Richtlinie, um die Kernfunktionen des Systems Manager Dienstes auf EC2 Instanzen zuzulassen. Informationen finden Sie unter Konfiguration von erforderliche Instance-Berechtigungen für Systems Manager.

Fehlerbehebung für AWS Systems Manager-Identität und -Zugriff

Verwenden Sie die folgenden Informationen, um häufige Probleme zu diagnostizieren und zu beheben, die beim Arbeiten mit AWS Systems Manager undAWS Identity and Access Management (IAM) auftreten könnten.

Themen

- Ich bin nicht autorisiert, eine Aktion in Systems Manager auszuführen.
- Ich bin nicht zur Ausführung von iam:PassRole autorisiert
- Ich möchte Personen außerhalb meines AWS-Konto Zugriff auf meine Systems Manager-Ressourcen gewähren

Ich bin nicht autorisiert, eine Aktion in Systems Manager auszuführen.

Wenn die AWS-Managementkonsole Ihnen mitteilt, dass Sie nicht zur Ausführung einer Aktion autorisiert sind, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson`-Benutzer versucht, die Konsole zum Anzeigen von Details zu einem Dokument zu verwenden, jedoch nicht über `ssm:GetDocument`-Berechtigungen verfügt.

```
User: arn:aws:ssm::123456789012:user/mateojackson isn't authorized to perform:
 ssm:GetDocument on resource: MyExampleDocument
```

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion `MyExampleDocument` auf die Ressource `ssm:GetDocument` zugreifen zu können.

Ich bin nicht zur Ausführung von iam:PassRole autorisiert

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zur Ausführung der Aktion `iam:PassRole` autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an Systems Manager übergeben zu können.

Einige AWS-Services erlauben die Übergabe einer vorhandenen Rolle an diesen Dienst, sodass keine neue Servicerolle oder serviceverknüpfte Rolle erstellt werden muss. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Systems Manager auszuführen. Um die Aktion ausführen zu können, benötigt der Service jedoch Berechtigungen, die von einer Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenden Sie sich an Ihren AWS-Administrator, falls Sie weitere Unterstützung benötigen. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meines AWS-Konto Zugriff auf meine Systems Manager-Ressourcen gewähren

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen dazu, ob Systems Manager diese Features unterstützt, finden Sie unter Funktionsweise von AWS Systems Manager mit IAM.
- Informationen zum Gewähren des Zugriffs auf Ihre Ressourcen für alle Ihre AWS-Konten finden Sie unter Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen Ihrer AWS-Konto im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie AWS-Konten-Drittanbieter Zugriff auf Ihre Ressourcen bereitstellen, finden Sie unter Gewähren des Zugriffs auf AWS-Konten von externen Benutzern im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwenden von serviceverknüpften Rollen für Systems Manager

AWS Systems Managerverwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Systems Manager verknüpft ist. Serviceverknüpfte Rollen werden von Systems Manager vordefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer AWS-Services in Ihrem Namen erfordert.

> **Note**
>
> Eine Servicerolle Rolle unterscheidet sich von einer servicegebundenen Rolle. Eine Servicerolle ist eine Art von AWS Identity and Access Management (IAM-) Rolle, die einer Person Berechtigungen erteilt, AWS-Service sodass der Dienst auf Ressourcen zugreifen kann. AWS Nur einige Systems Manager-Szenarien erfordern eine Servicerolle. Wenn Sie eine Servicerolle für Systems Manager erstellen, wählen Sie die dafür zu erteilenden Berechtigungen aus, damit ein Zugriff auf oder eine Interaktion mit anderen AWS -Ressourcen möglich ist.

Eine serviceverknüpfte Rolle vereinfacht die Einrichtung von Systems Manager, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Systems Manager definiert die Berechtigungen seiner serviceverknüpften Rollen. Sofern keine andere Konfiguration festgelegt wurde, kann nur Systems Manager die Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre Systems Manager-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

> **Note**
>
> Für EC2 Nicht-Knoten in einer Hybrid- und Multi-Cloud-Umgebung benötigen Sie eine zusätzliche IAM-Rolle, die es diesen Maschinen ermöglicht, mit dem Service zu kommunizieren. Systems Manager Dies ist die IAM-Servicerolle für Systems Manager. Diese Rolle gewährt AWS Security Token Service (AWS STS) AssumeRoleVertrauen in den Systems Manager Dienst. Die `AssumeRole`-Aktion gibt temporäre Sicherheitsanmeldeinformationen zurück (bestehend aus einer Zugriffsschlüssel-ID, einem geheimen Zugriffsschlüssel und einem Sicherheits-Token). Sie verwenden diese temporären Anmeldeinformationen, um auf AWS Ressourcen zuzugreifen, auf die Sie normalerweise keinen Zugriff haben. Weitere Informationen finden Sie unter Erstellen der für Systems Manager erforderlichen IAM-Servicerolle in Hybrid- und Multicloud-Umgebungen und AssumeRolein der AWS Security Token Service API-Referenz.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS-Services , die mit IAM arbeiten. Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Themen

- Verwenden von Rollen zum Sammeln und Anzeigen von Inventar OpsData
- Verwenden von Rollen zum Sammeln von AWS-Konto-Informationen für OpsCenter und Explorer
- Verwenden von Rollen zum Erstellen OpsData und OpsItems für Explorer
- Mithilfe von Rollen betriebliche Einblicke OpsItems in Systems Manager gewinnen OpsCenter
- Verwenden von Rollen zur Erhaltung des Zustands und der Konsistenz der von Quick Setup bereitgestellten Ressourcen
- Verwenden von Rollen zum Exportieren von Explorer OpsData
- Verwenden von Rollen zum Aktivieren des just-in-time Knotenzugriffs
- Verwenden von Rollen zum Senden von Benachrichtigungen über just-in-time Knotenzugriffsanfragen

Verwenden von Rollen zum Sammeln und Anzeigen von Inventar OpsData

Systems Managerverwendet die angegebene dienstbezogene Rolle. **`AWSServiceRoleForAmazonSSM`** AWS Systems Manager verwendet diese IAM-Servicerolle, um AWS Ressourcen in Ihrem Namen zu verwalten.

Dienstbezogene Rollenberechtigungen für Inventar, und OpsData OpsItems

Die serviceverknüpfte Rolle `AWSServiceRoleForAmazonSSM` vertraut nur darauf, dass `ssm.amazonaws.com` die Rolle annimmt.

Sie können die serviceverknüpfte Rolle `AWSServiceRoleForAmazonSSM` im Systems Manager für Folgendes verwenden:

- Das Inventory-Tool von Systems Manager verwendet die serviceverknüpfte Rolle `AWSServiceRoleForAmazonSSM` zum Erfassen von Bestand-Metadaten von Tags und Ressourcengruppen.
- Das Explorer Tool verwendet die dienstbezogene Rolle, `AWSServiceRoleForAmazonSSM` um die Anzeige OpsItems von OpsData und von mehreren Konten aus zu ermöglichen. Diese

dienstverknüpfte Rolle ermöglicht auch Explorer die Erstellung einer verwalteten Regel, wenn Sie Security Hub CSPM als Datenquelle von oder aktivieren. Explorer OpsCenter

> **Important**
>
> Bisher bot Ihnen die Systems Manager Manager-Konsole die Möglichkeit, die AWS verwaltete, mit dem IAM-Dienst verknüpfte Rolle auszuwählen`AWSServiceRoleForAmazonSSM`, die Sie als Wartungsrolle für Ihre Aufgaben verwenden möchten. Die Verwendung dieser Rolle und der zugehörigen Richtlinie, `AmazonSSMServiceRolePolicy`, für Wartungsfenster-Aufgaben wird nicht mehr empfohlen. Wenn Sie diese Rolle jetzt für Wartungsfenster-Aufgaben verwenden, empfehlen wir Ihnen, sie nicht mehr zu verwenden. Erstellen Sie stattdessen Ihre eigene IAM-Rolle, die die Kommunikation zwischen Systems Manager und anderen ermöglicht, AWS-Services wenn Ihre Wartungsfensteraufgaben ausgeführt werden.
> Weitere Informationen finden Sie unter Einrichten von Maintenance Windows.

Die verwaltete Richtlinie, die zum Bereitstellen von Berechtigungen für die `AWSServiceRoleForAmazonSSM`-Rolle verwendet wird, ist `AmazonSSMServiceRolePolicy`. Einzelheiten zu den Berechtigungen, gewährt werden, finden Sie unter AWS verwaltete Richtlinie: Amazon SSMService RolePolicy.

Erstellen einer **`AWSServiceRoleForAmazonSSM`**-serviceverknüpften Rolle für Systems Manager

Sie können die IAM-Konsole verwenden, um eine dienstbezogene Rolle mit dem EC2Anwendungsfall zu erstellen. Erstellen Sie mithilfe von Befehlen für IAM in AWS Command Line Interface (AWS CLI) oder mithilfe der IAM-API eine dienstverknüpfte Rolle mit dem `ssm.amazonaws.com`-Servicenamen. Weitere Informationen finden Sie unter Erstellen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen.

Bearbeiten einer **AWSServiceRoleForAmazonSSM**-serviceverknüpften Rolle für Systems Manager

Systems Manager lässt die Bearbeitung der serviceverknüpften Rolle namens `AWSServiceRoleForAmazonSSM` nicht zu. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung einer serviceverknüpften Rolle nicht bearbeitet werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer **AWSServiceRoleForAmazonSSM**-serviceverknüpften Rolle für Systems Manager

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise ist keine ungenutzte Entität vorhanden, die nicht aktiv überwacht oder verwaltet wird. Sie können die IAM-Konsole, die oder die IAM-API verwenden AWS CLI, um die serviceverknüpfte Rolle manuell zu löschen. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zuerst manuell bereinigen, bevor Sie diese manuell löschen können.

Da die serviceverknüpfte Rolle `AWSServiceRoleForAmazonSSM` von mehreren Tools verwendet werden kann, müssen Sie sicherstellen, dass keines der Tools diese Rolle verwendet, bevor Sie versuchen, diese zu löschen.

- Inventory: Wenn Sie die vom Bestands-Tool verwendete serviceverknüpfte Rolle löschen, werden die Bestands-Daten für Tags und Ressourcengruppen nicht mehr synchronisiert. Sie müssen die Ressourcen für Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.
- Explorer: Wenn Sie die vom Explorer Tool verwendete serviceverknüpfte Rolle löschen, sind die konto- und regionsübergreifenden Rollen nicht mehr OpsData sichtbar. OpsItems

> **Note**
>
> Wenn der Systems Manager-Service die Rolle verwendet, wenn Sie versuchen, Tags oder Ressourcengruppen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Löschen Sie die von **AWSServiceRoleForAmazonSSM** verwendeten Systems Manager-Ressourcen wie folgt:

1. Weitere Informationen zum Löschen von Tags finden Sie unter Hinzufügen und Löschen von Tags für einzelne Ressourcen.
2. Informationen zum Löschen von Ressourcengruppen finden Sie unter Gruppen löschen von. AWS -Ressourcengruppen

So löschen Sie die **AWSServiceRoleForAmazonSSM**-servicegebundene Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die IAM-API AWS CLI, um die `AWSServiceRoleForAmazonSSM` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für die Systems Manager **AWSServiceRoleForAmazonSSM**-serviceverknüpfte Rolle

Systems Managerunterstützt die Verwendung der `AWSServiceRoleForAmazonSSM` serviceverknüpften Rolle überall dort, AWS-Regionen wo der Service verfügbar ist. Weitere Informationen finden Sie unter AWS Systems Manager -Endpunkte und -Kontingente.

Verwenden von Rollen zum Sammeln von AWS-Konto-Informationen für OpsCenter und Explorer

Systems Manager verwendet die serviceverknüpfte Rolle namens **AWSServiceRoleForAmazonSSM_AccountDiscovery**. AWS Systems Manager verwendet diese IAM-Servicerolle, um andere AWS-Services aufzurufen, die AWS-Konto-Informationen ermitteln.

Berechtigungen von serviceverknüpften Rollen für Systems Manager-Kontoerkennung

Die serviceverknüpfte Rolle `AWSServiceRoleForAmazonSSM_AccountDiscovery` vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `accountdiscovery.ssm.amazonaws.com`

Die Rollenberechtigungsrichtlinie erlaubt Systems Manager die Durchführung der folgenden Aktionen für die angegebenen Ressourcen:

- `organizations:DescribeAccount`
- `organizations:DescribeOrganizationalUnit`
- `organizations:DescribeOrganization`
- `organizations:ListAccounts`
- `organizations:ListAWSServiceAccessForOrganization`
- `organizations:ListChildren`
- `organizations:ListParents`
- `organizations:ListDelegatedServicesForAccount`
- `organizations:ListDelegatedAdministrators`
- `organizations:ListRoots`

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer **`AWSServiceRoleForAmazonSSM_AccountDiscovery`**-serviceverknüpften Rolle für Systems Manager

Sie müssen eine serviceverknüpfte Rolle erstellen, wenn Sie Explorer- und OpsCenter-Tools von Systems Manager über mehrere AWS-Konten hinweg verwenden möchten. Für OpsCenter müssen Sie die serviceverknüpfte Rolle manuell erstellen. Weitere Informationen finden Sie unter (Optional) Manuelle Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems.

Wenn Sie für Explorer eine Ressourcendatensynchronisierung erstellen, indem Sie Systems Manager in der AWS-Managementkonsole verwenden, können Sie die serviceverknüpfte Rolle erstellen, indem Sie die Schaltfläche Create role (Rolle erstellen) auswählen. Wenn Sie eine Resource Data Sync programmgesteuert erstellen möchten, müssen Sie die Rolle erstellen, bevor Sie die Resource Data Sync erstellen. Sie können die Rolle mithilfe der API-Operation CreateServiceLinkedRole erstellen.

Bearbeiten einer **`AWSServiceRoleForAmazonSSM_AccountDiscovery`**-serviceverknüpften Rolle für Systems Manager

Systems Manager lässt die Bearbeitung der serviceverknüpften Rolle namens `AWSServiceRoleForAmazonSSM_AccountDiscovery` nicht zu. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung einer serviceverknüpften

Rolle nicht bearbeitet werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer **AWSServiceRoleForAmazonSSM_AccountDiscovery**-serviceverknüpften Rolle für Systems Manager

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise ist keine ungenutzte Entität vorhanden, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Bereinigen der **AWSServiceRoleForAmazonSSM_AccountDiscovery**-serviceverknüpften Rolle

Bevor Sie mit IAM eine `AWSServiceRoleForAmazonSSM_AccountDiscovery`-serviceverknüpfte Rolle löschen können, müssen Sie zunächst alle Explorer Resource Data Syncs löschen.

> **Note**
>
> Wenn der Systems Manager-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Manuelles Löschen der **AWSServiceRoleForAmazonSSM_AccountDiscovery**-serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, AWS CLI- oder AWS-API, um die `AWSServiceRoleForAmazonSSM_AccountDiscovery` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für die Systems Manager **AWSServiceRoleForAmazonSSM_AccountDiscovery**-serviceverknüpfte Rolle

Systems Manager unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter AWS Systems Manager-Endpunkte und -Kontingente.

Aktualisierungen der -serviceverknüpften Rolle AWSServiceRoleForAmazonSSM_AccountDiscovery

Details zu den Aktualisierungen der serviceverknüpften Rolle AWSServiceRoleForAmazonSSM_AccountDiscovery seit Beginn der Verfolgung dieser Änderungen durch diesen Service anzeigen. Um automatische Warnungen über Änderungen an dieser Seite erhalten, abonnieren Sie den RSS-Feed auf der Systems Manager Dokumentverlauf-Seite.

Änderung	Beschreibung	Datum
Neue Berechtigungen hinzugefügt	Diese serviceverknüpfte Rolle enthält jetzt `organizations:DescribeOrganizationalUnit` - und `organizations:ListRoots` -Berechtigungen. Diese Berechtigungen ermöglichen es einem AWS Organizations-Verwaltungskonto oder einem von Systems Manager delegierten Administratorkonto, kontenübergreifend mit OpsItems zu arbeiten. Weitere Informationen finden Sie unter (Optional) Manuelle Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems.	17. Oktober 2022

Verwenden von Rollen zum Erstellen OpsData und OpsItems für Explorer

Systems Managerverwendet die benannte dienstverknüpfte Rolle. **`AWSServiceRoleForSystemsManagerOpsDataSync`** AWS Systems Manager verwendet diese IAM-Dienstrolle Explorer zum Erstellen OpsData von und. OpsItems

Mit dem Dienst verknüpfte Rollenberechtigungen für die Synchronisierung Systems Manager OpsData

Die serviceverknüpfte Rolle `AWSServiceRoleForSystemsManagerOpsDataSync` vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `opsdatasync.ssm.amazonaws.com`

Die Rollenberechtigungsrichtlinie erlaubt Systems Manager die Durchführung der folgenden Aktionen für die angegebenen Ressourcen:

- Systems Manager Explorer erfordert, dass eine dienstverknüpfte Rolle die Erlaubnis erteilt, eine Sicherheitsfeststellung zu aktualisieren, wenn eine aktualisiert OpsItem wird, eine Sicherheitsfeststellung zu aktualisierenOpsItem, eine zu erstellen und zu aktualisieren und die Security Hub CSPM-Datenquelle zu deaktivieren, wenn eine von SSM verwaltete Regel von Kunden gelöscht wird.

Die verwaltete Richtlinie, die zum Bereitstellen von Berechtigungen für die `AWSServiceRoleForSystemsManagerOpsDataSync`-Rolle verwendet wird, ist `AWSSystemsManagerOpsDataSyncServiceRolePolicy`. Einzelheiten zu den Berechtigungen, gewährt werden, finden Sie unter AWS verwaltete Richtlinie: AWSSystems ManagerOpsDataSyncServiceRolePolicy.

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer **`AWSServiceRoleForSystemsManagerOpsDataSync`**-serviceverknüpften Rolle für Systems Manager

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie die Option aktivieren Explorer AWS-Managementkonsole, Systems Manager wird die dienstverknüpfte Rolle für Sie erstellt.

> **Important**
>
> Diese serviceverknüpfte Rolle kann in Ihrem Konto erscheinen, wenn Sie eine Aktion in einem anderen Service abgeschlossen haben, der die von dieser Rolle unterstützten

Features verwendet. Wenn Sie den Systems Manager-Service vor dem 1. Januar 2017 verwendet haben, als dieser begann, serviceverknüpfte Rollen zu unterstützen, dann hat Systems Manager die Rolle `AWSServiceRoleForSystemsManagerOpsDataSync` in Ihrem Konto erstellt. Weitere Informationen finden Sie unter In meinem IAM-Konto wird eine neue Rolle angezeigt.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie Explorer in aktivieren AWS-Managementkonsole, Systems Manager wird die dienstverknüpfte Rolle erneut für Sie erstellt.

Sie können auch die IAM-Konsole verwenden, um eine serviceverknüpfte Rolle mit der AWS Servicerolle zu erstellen, mit der Sie einen Fall erstellen OpsData und OpsItems verwenden können Explorer. Erstellen Sie in der AWS CLI oder der AWS API eine dienstbezogene Rolle mit dem `opsdatasync.ssm.amazonaws.com` Dienstnamen. Weitere Informationen finden Sie unter Erstellen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch. Wenn Sie diese serviceverknüpfte Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

Bearbeiten einer **`AWSServiceRoleForSystemsManagerOpsDataSync`**-serviceverknüpften Rolle für Systems Manager

Systems Manager lässt die Bearbeitung der serviceverknüpften Rolle namens `AWSServiceRoleForSystemsManagerOpsDataSync` nicht zu. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung einer serviceverknüpften Rolle nicht bearbeitet werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer **`AWSServiceRoleForSystemsManagerOpsDataSync`**-serviceverknüpften Rolle für Systems Manager

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise ist keine ungenutzte Entität vorhanden, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

> **Note**
>
> Wenn der Systems Manager-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Das Verfahren zum Löschen von Systems Manager Ressourcen, die von der `AWSServiceRoleForSystemsManagerOpsDataSync` Rolle verwendet werden, hängt davon ab, ob Sie Security Hub Explorer CSPM konfiguriert haben oder OpsCenter ob Sie es in Security Hub integrieren möchten.

Löschen Sie die von der **`AWSServiceRoleForSystemsManagerOpsDataSync`**-Rolle verwendeten Systems Manager-Ressourcen wie folgt:

- Informationen dazu, wie Sie Explorer verhindern können, neue CSPM-Ergebnisse OpsItems für Security Hub zu erstellen, finden Sie unter. Empfangen von Ergebnisse stoppen
- Um zu OpsCenter verhindern, dass neue CSPM-Ergebnisse OpsItems für Security Hub erstellt werden, siehe

So löschen Sie die **`AWSServiceRoleForSystemsManagerOpsDataSync`**-servicegebundene Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die serviceverknüpfte Rolle zu löschen. `AWSServiceRoleForSystemsManagerOpsDataSync` Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für die Systems Manager **`AWSServiceRoleForSystemsManagerOpsDataSync`**-serviceverknüpfte Rolle

Systems Manager unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter AWS Systems Manager - Endpunkte und -Kontingente.

Systems Manager unterstützt die Verwendung von serviceverknüpften Rollen nicht in allen Regionen, in denen der Service verfügbar ist. Sie können die Rolle `AWSServiceRoleForSystemsManagerOpsDataSync` in den folgenden Regionen verwenden.

AWS-Region Name	Regions-ID	Unterstützung in Systems Manager
USA Ost (Nord-Virginia)	us-east-1	Ja
USA Ost (Ohio)	us-east-2	Ja
USA West (Nordkalifornien)	us-west-1	Ja
USA West (Oregon)	us-west-2	Ja
Asien-Pazifik (Mumbai)	ap-south-1	Ja
Asien-Pazifik (Osaka)	ap-northeast-3	Ja
Asien-Pazifik (Seoul)	ap-northeast-2	Ja
Asien-Pazifik (Singapore)	ap-southeast-1	Ja
Asien-Pazifik (Sydney)	ap-southeast-2	Ja
Asien-Pazifik (Tokyo)	ap-northeast-1	Ja
Kanada (Zentral)	ca-central-1	Ja
Europa (Frankfurt)	eu-central-1	Ja
Europa (Ireland)	eu-west-1	Ja
Europa (London)	eu-west-2	Ja
Europa (Paris)	eu-west-3	Ja
Europa (Stockholm)	eu-north-1	Ja
Südamerika (São Paulo)	sa-east-1	Ja
AWS GovCloud (US)	us-gov-west-1	Nein

Mithilfe von Rollen betriebliche Einblicke OpsItems in Systems Manager gewinnen OpsCenter

Systems Managerverwendet die mit dem Dienst verknüpfte Rolle mit dem Namen. **`AWSServiceRoleForAmazonSSM_OpsInsights`** AWS Systems Manager verwendet diese IAM-Servicerolle, um betriebliche Einblicke OpsItems in Systems Manager OpsCenter zu erstellen und zu aktualisieren.

`AWSServiceRoleForAmazonSSM_OpsInsights`Mit dem Dienst verknüpfte Rollenberechtigungen für betriebliche Einblicke Systems Manager OpsItems

Die serviceverknüpfte Rolle `AWSServiceRoleForAmazonSSM_OpsInsights` vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `opsinsights.ssm.amazonaws.com`

Die Rollenberechtigungsrichtlinie erlaubt Systems Manager die Durchführung der folgenden Aktionen für die angegebenen Ressourcen:

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Sid": "AllowCreateOpsItem",
   "Effect": "Allow",
   "Action": [
    "ssm:CreateOpsItem",
    "ssm:AddTagsToResource"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AllowAccessOpsItem",
   "Effect": "Allow",
   "Action": [
    "ssm:UpdateOpsItem",
    "ssm:GetOpsItem"
   ],
```

```
    "Resource": "*",
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/SsmOperationalInsight": "true"
     }
    }
   }
  ]
}
```

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer **AWSServiceRoleForAmazonSSM_OpsInsights**-serviceverknüpften Rolle für Systems Manager

Sie müssen eine serviceverknüpfte Rolle erstellen. Wenn Sie Operational Insights mithilfe von Systems Manager in aktivieren AWS-Managementkonsole, können Sie die serviceverknüpfte Rolle erstellen, indem Sie auf die Schaltfläche Aktivieren klicken.

Bearbeiten einer **AWSServiceRoleForAmazonSSM_OpsInsights**-serviceverknüpften Rolle für Systems Manager

Systems Manager verhindert die Bearbeitung der serviceverknüpften Rolle `AWSServiceRoleForAmazonSSM_OpsInsights`. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer **AWSServiceRoleForAmazonSSM_OpsInsights**-serviceverknüpften Rolle für Systems Manager

Wenn Sie ein Feature oder einen Dienst, die bzw. der eine serviceverknüpften Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Bereinigen der **`AWSServiceRoleForAmazonSSM_OpsInsights`**-serviceverknüpften Rolle

Bevor Sie IAM zum Löschen einer serviceverknüpften `AWSServiceRoleForAmazonSSM_OpsInsights`-Rolle verwenden können, müssen Sie zunächst betriebliche Einblicke in Systems Manager OpsCenter deaktivieren. Weitere Informationen finden Sie unter Analyse betrieblicher Einblicke zur Reduzierung von OpsItems.

Manuelles Löschen der **`AWSServiceRoleForAmazonSSM_OpsInsights`**-serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die `AWSServiceRoleForAmazonSSM_OpsInsights` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für die Systems Manager **`AWSServiceRoleForAmazonSSM_OpsInsights`**-serviceverknüpfte Rolle

Systems Manager unterstützt die Verwendung von serviceverknüpften Rollen nicht in allen Regionen, in denen der Service verfügbar ist. Sie können die AWSServiceRoleForAmazonSSM_OpsInsights Rolle in den folgenden Regionen verwenden.

Name der Region	Regions-ID	Unterstützung in Systems Manager
USA Ost (Nord-Virginia)	us-east-1	Ja
USA Ost (Ohio)	us-east-2	Ja
USA West (Nordkalifornien)	us-west-1	Ja
USA West (Oregon)	us-west-2	Ja
Asien-Pazifik (Mumbai)	ap-south-1	Ja
Asien-Pazifik (Tokyo)	ap-northeast-1	Ja
Asien-Pazifik (Seoul)	ap-northeast-2	Ja
Asien-Pazifik (Singapore)	ap-southeast-1	Ja
Asien-Pazifik (Sydney)	ap-southeast-2	Ja

Name der Region	Regions-ID	Unterstützung in Systems Manager
Asien-Pazifik (Hongkong)	ap-east-1	Ja
Kanada (Zentral)	ca-central-1	Ja
Europa (Frankfurt)	eu-central-1	Ja
Europa (Ireland)	eu-west-1	Ja
Europa (London)	eu-west-2	Ja
Europa (Paris)	eu-west-3	Ja
Europa (Stockholm)	eu-north-1	Ja
Europa (Milan)	eu-south-1	Ja
Südamerika (São Paulo)	sa-east-1	Ja
Naher Osten (Bahrain)	me-south-1	Ja
Afrika (Kapstadt)	af-south-1	Ja
AWS GovCloud (US)	us-gov-west-1	Ja
AWS GovCloud (US)	us-gov-east-1	Ja

Verwenden von Rollen zur Erhaltung des Zustands und der Konsistenz der von Quick Setup bereitgestellten Ressourcen

Systems Manager verwendet die serviceverknüpfte Rolle namens **`AWSServiceRoleForSSMQuickSetup`**.

`AWSServiceRoleForSSMQuickSetup`-Berechtigungen von serviceverknüpften Rollen für Systems Manager

Die serviceverknüpfte Rolle `AWSServiceRoleForSSMQuickSetup` vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `ssm-quicksetup.amazonaws.com`

AWS Systems Manager verwendet diese IAM-Servicerolle, um den Zustand der Konfiguration zu überprüfen, die konsistente Verwendung von Parametern und bereitgestellten Ressourcen sicherzustellen und Ressourcen zu korrigieren, wenn Abweichungen festgestellt werden.

Die Rollenberechtigungsrichtlinie erlaubt Systems Manager die Durchführung der folgenden Aktionen für die angegebenen Ressourcen:

- `ssm` (Systems Manager) – Liest Informationen über den Status, in dem sich konfigurierte Ressourcen befinden sollen, einschließlich delegierter Administratorkonten.
- `iam` (AWS Identity and Access Management) – Dies ist erforderlich, damit Ressourcendaten organisationsweit in AWS Organizations synchronisiert werden können.
- `organizations` (AWS Organizations) – Liest Informationen über die Mitgliedskonten, die zu einer Organisation gehören, wie sie in Organizations konfiguriert ist.
- `cloudformation` (CloudFormation) – Liest Informationen über CloudFormation-Stacks, die zur Verwaltung des Status von Ressourcen und CloudFormation-Stackset-Vorgängen verwendet werden.

Die verwaltete Richtlinie, die zum Bereitstellen von Berechtigungen für die `AWSServiceRoleForSSMQuickSetup`-Rolle verwendet wird, ist `SSMQuickSetupRolePolicy`. Einzelheiten zu den Berechtigungen, gewährt werden, finden Sie unter AWS verwaltete Richtlinie: SSMQuick SetupRolePolicy.

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer **`AWSServiceRoleForSSMQuickSetup`**-serviceverknüpften Rolle für Systems Manager

Sie müssen die serviceverknüpfte Rolle AWSServiceRoleForSSMQuickSetup nicht manuell erstellen. Wenn Sie eine Quick Setup-Konfiguration in der AWS-Managementkonsole erstellen, erstellt Systems Manager die serviceverknüpfte Rolle für Sie.

Bearbeiten einer **AWSServiceRoleForSSMQuickSetup**-serviceverknüpften Rolle für Systems Manager

Systems Manager verhindert die Bearbeitung der serviceverknüpften Rolle `AWSServiceRoleForSSMQuickSetup`. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer **AWSServiceRoleForSSMQuickSetup**-serviceverknüpften Rolle für Systems Manager

Wenn Sie ein Feature oder einen Dienst, die bzw. der eine serviceverknüpften Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Bereinigen der **AWSServiceRoleForSSMQuickSetup**-serviceverknüpften Rolle

Bevor Sie die serviceverknüpfte Rolle mit `AWSServiceRoleForSSMQuickSetup` löschen können, müssen Sie zunächst die Konfigurationen Quick Setup, die die Rollen nutzen über die Konsole deaktivieren. Weitere Informationen finden Sie unter Bearbeiten und Löschen Ihrer Konfiguration.

Manuelles Löschen der **AWSServiceRoleForSSMQuickSetup**-serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, AWS CLI- oder AWS-API, um die `AWSServiceRoleForSSMQuickSetup` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter den folgenden Themen:

- Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch
- delete-configuration-manager im Abschnitt Quick Setup der AWS CLI-Referenz
- DeleteConfigurationManager in der Quick Setup-API-Referenz

Unterstützte Regionen für die Systems Manager **AWSServiceRoleForSSMQuickSetup**-serviceverknüpfte Rolle

Systems Manager unterstützt die Verwendung von serviceverknüpften Rollen nicht in allen Regionen, in denen der Service verfügbar ist. Sie können die Rolle AWSServiceRoleForSSMQuickSetup in den folgenden Regionen verwenden.

- US East (Ohio)
- USA Ost (Nord-Virginia)
- USA West (Nordkalifornien)
- USA West (Oregon)
- Asia Pacific (Mumbai)
- Asia Pacific (Seoul)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)
- Canada (Central)
- Europa (Frankfurt)
- Europa (Stockholm)
- Europa (Irland)
- Europa (London)
- Europe (Paris)
- Südamerika (São Paulo)

Verwenden von Rollen zum Exportieren von Explorer OpsData

AWS Systems Manager Explorer verwendet die Servicerolle AmazonSSMExplorerExportRole, um Betriebsdaten (OpsData) mithilfe des Automation-Runbooks `AWS-ExportOpsDataToS3` zu exportieren.

Berechtigungen von serviceverknüpften Rollen für Explorer

Die serviceverknüpfte Rolle `AmazonSSMExplorerExportRole` vertraut nur darauf, dass `ssm.amazonaws.com` die Rolle annimmt.

Sie können die `AmazonSSMExplorerExportRole`-serviceverknüpfte Rolle verwenden, um Betriebsdaten (OpsData) mithilfe des `AWS-ExportOpsDataToS3`-Automation-Runbooks zu exportieren. Sie können 5 000 OpsData-Elemente von Explorer als kommagetrennte Datei (.csv) in einen Amazon Simple Storage Service (Amazon S3)-Bucket exportieren.

Die Rollenberechtigungsrichtlinie erlaubt Systems Manager die Durchführung der folgenden Aktionen für die angegebenen Ressourcen:

- `s3:PutObject`
- `s3:GetBucketAcl`
- `s3:GetBucketLocation`
- `sns:Publish`
- `logs:DescribeLogGroups`
- `logs:DescribeLogStreams`
- `logs:CreateLogGroup`
- `logs:PutLogEvents`
- `logs:CreateLogStream`
- `ssm:GetOpsSummary`

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer **`AmazonSSMExplorerExportRole`**-serviceverknüpften Rolle für Systems Manager

Systems Manager erstellt die `AmazonSSMExplorerExportRole`-servicebezogene Rolle, wenn Sie OpsData mit Explorer die Systems-Manager-Konsole exportieren. Weitere Informationen finden Sie unter OpsData Aus Systems Manager exportieren Explorer.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen.

Bearbeiten einer **`AmazonSSMExplorerExportRole`**-serviceverknüpften Rolle für Systems Manager

Systems Manager lässt die Bearbeitung der serviceverknüpften Rolle namens `AmazonSSMExplorerExportRole` nicht zu. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung einer serviceverknüpften Rolle nicht bearbeitet werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer **AmazonSSMExplorerExportRole**-serviceverknüpften Rolle für Systems Manager

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise ist keine ungenutzte Entität vorhanden, die nicht aktiv überwacht oder verwaltet wird. Sie können jedoch die IAM-Konsole, AWS CLI oder IAM-API verwenden, um eine serviceverknüpfte Rolle manuell zu löschen. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zuerst manuell bereinigen, bevor Sie diese manuell löschen können.

> **Note**
>
> Wenn der Systems Manager-Service die Rolle verwendet, wenn Sie versuchen, Tags oder Ressourcengruppen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Löschen Sie die von **AmazonSSMExplorerExportRole** verwendeten Systems Manager-Ressourcen wie folgt:

1. Weitere Informationen zum Löschen von Tags finden Sie unter Hinzufügen und Löschen von Tags für einzelne Ressourcen.
2. Weitere Informationen zum Löschen von Ressourcengruppen finden Sie unter Löschen von Gruppen aus AWS -Ressourcengruppen.

To manually delete the **AmazonSSMExplorerExportRole** service-linked role using IAM (So löschen Sie die servicegebundene Rolle mit IAM)

Verwenden Sie die IAM-Konsole, AWS CLI oder IAM-API, um die `AmazonSSMExplorerExportRole` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für die Systems Manager **AmazonSSMExplorerExportRole**-serviceverknüpfte Rolle

Systems Manager unterstützt die Verwendung der `AmazonSSMExplorerExportRole`-serviceverknüpften Rolle in allen AWS-Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter AWS Systems Manager-Endpunkte und -Kontingente.

Verwenden von Rollen zum Aktivieren des just-in-time Knotenzugriffs

Systems Managerverwendet die benannte dienstverknüpfte Rolle. **`AWSServiceRoleForSystemsManagerJustInTimeAccess`** AWS Systems Manager verwendet diese IAM-Dienstrolle, um den just-in-time Knotenzugriff zu ermöglichen.

Dienstbezogene Rollenberechtigungen für Systems Manager just-in-time den Knotenzugriff

Die serviceverknüpfte Rolle `AWSServiceRoleForSystemsManagerJustInTimeAccess` vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `ssm.amazonaws.com`

Die Rollenberechtigungsrichtlinie erlaubt Systems Manager die Durchführung der folgenden Aktionen für die angegebenen Ressourcen:

- `ssm:CreateOpsItem`
- `ssm:GetOpsItem`
- `ssm:UpdateOpsItem`
- `ssm:DescribeOpsItems`
- `ssm:DescribeSessions`
- `ssm:ListTagsForResource`
- `ssm-guiconnect:ListConnections`
- `identitystore:ListGroupMembershipsForMember`
- `identitystore:DescribeUser`
- `identitystore:GetGroupId`
- `identitystore:GetUserId`
- `sso-directory:DescribeUsers`
- `sso-directory:IsMemberInGroup`
- `sso:ListInstances`
- `sso:DescribeRegisteredRegions`
- `sso:ListDirectoryAssociations`
- `ec2:DescribeTags`

Die verwaltete Richtlinie, die zum Bereitstellen von Berechtigungen für die `AWSServiceRoleForSystemsManagerJustInTimeAccess`-Rolle verwendet wird, ist `AWSSystemsManagerEnableJustInTimeAccessPolicy`. Einzelheiten zu den Berechtigungen, gewährt werden, finden Sie unter AWS verwaltete Richtlinie: AWSSystems ManagerJustInTimeAccessServicePolicy.

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer **`AWSServiceRoleForSystemsManagerJustInTimeAccess`**-serviceverknüpften Rolle für Systems Manager

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie den just-in-time Knotenzugriff in aktivieren AWS-Managementkonsole, Systems Manager wird die dienstverknüpfte Rolle für Sie erstellt.

> **Important**
>
> Diese serviceverknüpfte Rolle kann in Ihrem Konto erscheinen, wenn Sie eine Aktion in einem anderen Service abgeschlossen haben, der die von dieser Rolle unterstützten Features verwendet. Wenn Sie den Systems Manager-Service bereits seit der Einführung von serviceverknüpften Rollen am 19. November 2024 nutzen, hat Systems Manager die Rolle `AWSServiceRoleForSystemsManagerJustInTimeAccess` in Ihrem Konto erstellt. Weitere Informationen finden Sie unter In meinem IAM-Konto wird eine neue Rolle angezeigt.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie den just-in-time Knotenzugriff in aktivieren AWS-Managementkonsole, Systems Manager erstellt die dienstverknüpfte Rolle erneut für Sie.

Sie können die IAM-Konsole auch verwenden, um eine serviceverknüpfte Rolle mit der AWS Servicerolle zu erstellen, die es Systems Manager ermöglicht, den just-in-time Knotenzugriff zu aktivieren. Anwendungsfall. Erstellen Sie in der AWS CLI oder der AWS API eine dienstverknüpfte Rolle mit dem `ssm.amazonaws.com` Dienstnamen. Weitere Informationen finden Sie unter Erstellen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch. Wenn Sie diese serviceverknüpfte Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

Bearbeiten einer **AWSServiceRoleForSystemsManagerJustInTimeAccess**-serviceverknüpften Rolle für Systems Manager

Systems Manager lässt die Bearbeitung der serviceverknüpften Rolle namens `AWSServiceRoleForSystemsManagerJustInTimeAccess` nicht zu. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung einer serviceverknüpften Rolle nicht bearbeitet werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer **AWSServiceRoleForSystemsManagerJustInTimeAccess**-serviceverknüpften Rolle für Systems Manager

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise ist keine ungenutzte Entität vorhanden, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

> **Note**
>
> Wenn der Systems Manager-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

So löschen Sie die **AWSServiceRoleForSystemsManagerJustInTimeAccess**-servicegebundene Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die `AWSServiceRoleForSystemsManagerJustInTimeAccess` dienstverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für die Systems Manager **AWSServiceRoleForSystemsManagerJustInTimeAccess**-serviceverknüpfte Rolle

AWS-Region Name	Regions-ID	Unterstützung in Systems Manager
USA Ost (Nord-Virginia)	us-east-1	Ja
USA Ost (Ohio)	us-east-2	Ja
USA West (Nordkalifornien)	us-west-1	Ja
USA West (Oregon)	us-west-2	Ja
Asien-Pazifik (Mumbai)	ap-south-1	Ja
Asien-Pazifik (Osaka)	ap-northeast-3	Ja
Asien-Pazifik (Seoul)	ap-northeast-2	Ja
Asien-Pazifik (Singapore)	ap-southeast-1	Ja
Asien-Pazifik (Sydney)	ap-southeast-2	Ja
Asien-Pazifik (Tokyo)	ap-northeast-1	Ja
Kanada (Zentral)	ca-central-1	Ja
Europa (Frankfurt)	eu-central-1	Ja
Europa (Ireland)	eu-west-1	Ja
Europa (London)	eu-west-2	Ja
Europa (Paris)	eu-west-3	Ja
Europa (Stockholm)	eu-north-1	Ja
Südamerika (São Paulo)	sa-east-1	Ja

AWS-Region Name	Regions-ID	Unterstützung in Systems Manager
AWS GovCloud (US)	us-gov-west-1	Nein

Verwenden von Rollen zum Senden von Benachrichtigungen über just-in-time Knotenzugriffsanfragen

Systems Managerverwendet die benannte dienstverknüpfte Rolle. **`AWSServiceRoleForSystemsManagerNotifications`** AWS Systems Manager verwendet diese IAM-Dienstrolle, um Benachrichtigungen an Genehmigungsberechtigte für Zugriffsanfragen zu senden.

Mit dem Dienst verknüpfte Rollenberechtigungen für Systems Manager just-in-time Benachrichtigungen über den Knotenzugriff

Die serviceverknüpfte Rolle `AWSServiceRoleForSystemsManagerNotifications` vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `ssm.amazonaws.com`

Die Rollenberechtigungsrichtlinie erlaubt Systems Manager die Durchführung der folgenden Aktionen für die angegebenen Ressourcen:

- `identitystore:ListGroupMembershipsForMember`
- `identitystore:ListGroupMemberships`
- `identitystore:DescribeUser`
- `sso:ListInstances`
- `sso:DescribeRegisteredRegions`
- `sso:ListDirectoryAssociations`
- `sso-directory:DescribeUser`
- `sso-directory:ListMembersInGroup`
- `iam:GetRole`

Die verwaltete Richtlinie, die zum Bereitstellen von Berechtigungen für die `AWSServiceRoleForSystemsManagerNotifications`-Rolle verwendet wird, ist `AWSSystemsManagerNotificationsServicePolicy`. Einzelheiten zu den Berechtigungen, gewährt werden, finden Sie unter AWS verwaltete Richtlinie: AWSSystems ManagerNotificationsServicePolicy.

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer **`AWSServiceRoleForSystemsManagerNotifications`**-serviceverknüpften Rolle für Systems Manager

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie den just-in-time Knotenzugriff in aktivieren AWS-Managementkonsole, Systems Manager wird die dienstverknüpfte Rolle für Sie erstellt.

> **Important**
>
> Diese serviceverknüpfte Rolle kann in Ihrem Konto erscheinen, wenn Sie eine Aktion in einem anderen Service abgeschlossen haben, der die von dieser Rolle unterstützten Features verwendet. Wenn Sie den Systems Manager-Service bereits seit der Einführung von serviceverknüpften Rollen am 19. November 2024 nutzen, hat Systems Manager die Rolle `AWSServiceRoleForSystemsManagerNotifications` in Ihrem Konto erstellt. Weitere Informationen finden Sie unter In meinem IAM-Konto wird eine neue Rolle angezeigt.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie den just-in-time Knotenzugriff in aktivieren AWS-Managementkonsole, Systems Manager erstellt die dienstverknüpfte Rolle erneut für Sie.

Sie können auch den Anwendungsfall AWS -Servicerolle, die es Systems Manager ermöglicht, Benachrichtigungen an Genehmigungsberechtigte für Zugriffsanforderungen zu senden in der IAM-Konsole verwenden, um eine serviceverknüpfte Rolle zu erstellen. Erstellen Sie in der AWS CLI oder der AWS API eine dienstverknüpfte Rolle mit dem `ssm.amazonaws.com` Dienstnamen. Weitere Informationen finden Sie unter Erstellen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch. Wenn Sie diese serviceverknüpfte Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

Bearbeiten einer **AWSServiceRoleForSystemsManagerNotifications**-serviceverknüpften Rolle für Systems Manager

Systems Manager lässt die Bearbeitung der serviceverknüpften Rolle namens `AWSServiceRoleForSystemsManagerNotifications` nicht zu. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung einer serviceverknüpften Rolle nicht bearbeitet werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer **AWSServiceRoleForSystemsManagerNotifications**-serviceverknüpften Rolle für Systems Manager

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise ist keine ungenutzte Entität vorhanden, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

> **Note**
>
> Wenn der Systems Manager-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

So löschen Sie die **AWSServiceRoleForSystemsManagerNotifications**-servicegebundene Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die `AWSServiceRoleForSystemsManagerNotifications` dienstverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für die Systems Manager **AWSServiceRoleForSystemsManagerNotifications**-serviceverknüpfte Rolle

AWS-Region Name	Regions-ID	Unterstützung in Systems Manager
USA Ost (Nord-Virginia)	us-east-1	Ja
USA Ost (Ohio)	us-east-2	Ja
USA West (Nordkalifornien)	us-west-1	Ja
USA West (Oregon)	us-west-2	Ja
Asien-Pazifik (Mumbai)	ap-south-1	Ja
Asien-Pazifik (Osaka)	ap-northeast-3	Ja
Asien-Pazifik (Seoul)	ap-northeast-2	Ja
Asien-Pazifik (Singapore)	ap-southeast-1	Ja
Asien-Pazifik (Sydney)	ap-southeast-2	Ja
Asien-Pazifik (Tokyo)	ap-northeast-1	Ja
Kanada (Zentral)	ca-central-1	Ja
Europa (Frankfurt)	eu-central-1	Ja
Europa (Ireland)	eu-west-1	Ja
Europa (London)	eu-west-2	Ja
Europa (Paris)	eu-west-3	Ja
Europa (Stockholm)	eu-north-1	Ja
Südamerika (São Paulo)	sa-east-1	Ja

AWS-Region Name	Regions-ID	Unterstützung in Systems Manager
AWS GovCloud (US)	us-gov-west-1	Nein

Protokollieren und Überwachen in AWS Systems Manager

Die Überwachung ist ein wichtiger Teil der Wahrung von Zuverlässigkeit, Verfügbarkeit und Performance von AWS Systems Manager und Ihren AWS-Lösungen. Sie sollten Überwachungsdaten von allen Teilen Ihrer AWS-Lösung sammeln, damit Sie einen Multi-Point-Fehler debuggen können, falls ein solcher auftritt. AWS bietet verschiedene Tools zur Überwachung von Systems Manager und anderer Ressourcen und zur Reaktion auf potenzielle Vorfälle.

AWS CloudTrail-Protokolle

CloudTrail stellt eine Aufzeichnung der Aktionen bereit, die von einem Benutzer, einer Rolle oder einem AWS-Service in Systems Manager durchgeführt wurden. Mit den von CloudTrail erfassten Informationen können Sie die an Systems Manager gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und zusätzliche Details bestimmen. Weitere Informationen finden Sie unter Protokollierung von AWS Systems Manager-API-Aufrufen mit AWS CloudTrail.

Amazon CloudWatch-Alarme

Mit Amazon CloudWatch-Alarmen überwachen Sie eine einzelne Metrik über einen Zeitraum, den Sie für Ihre Amazon Elastic Compute Cloud (Amazon EC2)-Instances und andere Ressourcen festlegen. Wenn die Metrik einen bestimmten Schwellenwert überschreitet, wird eine Benachrichtigung an ein Thema von Amazon Simple Notification Service (Amazon SNS) oder eine AWS Auto Scaling-Richtlinie gesendet. CloudWatch-Alarme rufen keine Aktionen auf, nur weil sie sich in einem bestimmten Status befinden. Der Status muss sich stattdessen geändert haben und für eine festgelegte Anzahl an Zeiträumen aufrechterhalten worden sein. Weitere Informationen finden Sie unter Verwenden von Amazon CloudWatch-Alarmen im Amazon CloudWatch-Benutzerhandbuch.

Amazon CloudWatch Dashboards

CloudWatch-Dashboards sind anpassbare Startseiten in der CloudWatch-Konsole, mit denen Sie Ihre Ressourcen in einer einzigen Ansicht überwachen können, auch solche, die über verschiedene AWS-Regionen verteilt sind. Sie können CloudWatch-Dashboards verwenden, um

benutzerdefinierte Ansichten der Metriken und Alarme für Ihre AWS Ressourcen zu erstellen. Weitere Informationen finden Sie unter Amazon-CloudWatch-Dashboards, die von Systems Manager gehostet werden.

Amazon EventBridge

Mit Amazon EventBridge können Sie Regeln konfigurieren, um Sie auf Änderungen in Systems Manager-Ressourcen aufmerksam zu machen und EventBridge anzuweisen, Aktionen basierend auf dem Inhalt dieser Ereignisse durchzuführen. EventBridge bietet Unterstützung für eine Reihe von Ereignissen, die von verschiedenen Systems Manager-Tools ausgegeben werden. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge.

Amazon CloudWatch Logs und SSM Agent-Protokolle

SSM Agent schreibt Informationen zu Ausführungen, geplanten Aktionen, Fehlern und dem Zustandsstatus in Protokolldateien auf jedem Knoten. Sie können Protokolldateien anzeigen, indem Sie sich manuell mit einem Knoten verbinden. Es wird empfohlen, Agentprotokolldaten zur Analyse automatisch an eine Protokollgruppe in CloudWatch Logs zu senden. Weitere Informationen erhalten Sie unter Senden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent) und Anzeigen von SSM Agent-Protokollen.

AWS Systems Manager-Compliance

Mit Compliance, einem Tool in AWS Systems Manager, können Sie Ihre Flotte verwalteter Knoten auf Patch-Compliance und Konfigurations-Inkonsistenzen prüfen. Sie können Daten aus mehreren AWS-Konten und AWS-Regionen erfassen und zusammenfassen und anschließend Details zu bestimmten Ressourcen anzeigen, die nicht konform sind. Standardmäßig zeigt Compliance aktuelle Compliance-Daten zum Patchen in Patch Manager, einem Tool in AWS Systems Manager, und Verknüpfungen in State Manager an, einem Tool in AWS Systems Manager. Weitere Informationen finden Sie unter AWS Systems Manager-Compliance.

AWS Systems Manager Explorer

Explorer, ein Tool in AWS Systems Manager, ist ein anpassbares Betriebs-Dashboard, in dem Informationen zu Ihren AWS-Ressourcen angezeigt werden. Explorer bietet eine aggregierte Ansicht der Betriebsdaten (OpsData) für Ihre AWS-Konten und über AWS-Regionen hinweg. In Explorer enthalten OpsData Metadaten zu Ihren EC2-Instances, Details zur Patch-Compliance und operative Arbeitselemente (OpsItems). Explorer macht deutlich, wie OpsItems über Ihre Geschäftsbereiche oder Anwendungen verteilt sind, welche Trends im Zeitverlauf bestehen und wie sie je nach Kategorie variieren. Sie können Informationen in Explorer gruppieren und filtern,

um sich auf die Elemente zu konzentrieren, die für Sie relevant sind und eine Aktion erfordern. Weitere Informationen finden Sie unter AWS Systems Manager Explorer.

AWS Systems Manager OpsCenter

OpsCenter, ein Tool in AWS Systems Manager, bietet einen zentralen Ort, an dem Betriebstechniker und IT-Experten operative Arbeitselemente (OpsItems) im Zusammenhang mit AWS-Ressourcen anzeigen, untersuchen und lösen können. OpsCenter aggregiert und standardisiert OpsItems serviceübergreifend und stellt gleichzeitig kontextbezogene Untersuchungsdaten zu jedem OpsItem, zugehörigen OpsItems und zugehörigen Ressourcen bereit. OpsCenter bietet auch Automation-Dokumente, ein Tool in AWS Systems Manager, mit dem Sie Probleme schnell lösen können. OpsCenter ist in Amazon EventBridge integriert. Dadurch können Sie EventBridge-Regeln erstellen, die automatisch OpsItems für alle AWS-Service erzeugen, die Ereignisse an EventBridge veröffentlichen. Weitere Informationen finden Sie unter AWS Systems Manager OpsCenter.

Amazon Simple Notification Service

Sie können Amazon Simple Notification Service (Amazon SNS) zum Senden von Benachrichtigungen über den Status der Befehle konfigurieren, die Sie mithilfe von Run Command oder Maintenance Windows, Tools in AWS Systems Manager, senden. Amazon SNS koordiniert und verwaltet das Senden und Zustellen von Benachrichtigungen an Clients oder Endpunkte, die Amazon-SNS-Themen abonniert haben. Sie können eine Benachrichtigung erhalten, wenn ein Befehl in einen neuen Status oder in einen bestimmten Status wechselt, z. B. `Failed` oder `Timed Out`. In Fällen, in denen Sie einen Befehl an mehrere Knoten senden, können Sie eine Benachrichtigung für jede Kopie des Befehls abrufen, die an einen bestimmten Knoten gesendet wurde. Weitere Informationen finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

AWS Trusted Advisor und AWS Health Dashboard

Trusted Advisor stützt sich auf bewährte Methoden, die sich während der gesamten Betriebsgeschichte der Betreuung vieler Hunderttausend AWS-Kunden ergeben haben. Trusted Advisor überprüft Ihre AWS-Umgebung und gibt dann Empfehlungen, sobald sich Möglichkeiten ergeben, Kosten zu senken, die Systemleistung zu verbessern oder Schwachstellen zu schließen. Alle AWS-Kunden haben Zugriff auf fünf Trusted Advisor-Prüfungen. Kunden mit einem AWS -Support Business- oder Enterprise-Plan können alle Trusted Advisor-Prüfungen anzeigen. Weitere Informationen finden Sie unter AWS Trusted Advisor im AWS -Support-Benutzerhandbuch und im AWS Health-Benutzerhandbuch.

Weitere Informationen

- Einloggen und Überwachen AWS Systems Manager

Compliance-Validierung für AWS Systems Manager

Dieses Thema befasst sich mit der AWS Systems Manager-Compliance mit Assurance-Programmen von Drittanbietern. Informationen zum Anzeigen von Compliance-Daten für Ihre verwalteten Knoten finden Sie unter AWS Systems Manager-Compliance.

Externe Prüfer bewerten im Rahmen verschiedener AWS-Compliance-Programme die Sicherheit und Compliance von Systems Manager. Hierzu zählen unter anderem SOC, PCI, FedRAMP und HIPAA.

Eine Liste der AWS-Services, die in den Anwendungsbereich bestimmter Compliance-Programme fallen, finden Sie unter AWS-Services im Anwendungsbereich nach Compliance-Programm . Allgemeine Informationen finden Sie unter AWS-Compliance-Programme.

Sie können Auditberichte von Drittanbietern unter AWS Artifact herunterladen. Weitere Informationen finden Sie unter Berichte herunterladen in AWS Artifact.

Ihre Compliance-Verantwortung bei der Verwendung von Systems Manager ist von der Sensibilität Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften abhängig. AWS stellt die folgenden Ressourcen zur Unterstützung der Compliance bereit:

- Kurzanleitungen für Sicherheit und Compliance – In diesen Bereitstellungsleitfäden finden Sie wichtige Überlegungen zur Architektur sowie die einzelnen Schritte zur Bereitstellung von sicherheits- und Compliance-orientierten Basisumgebungen in AWS.
- Whitepaper zur Erstellung einer Architektur mit HIPAA-konformer Sicherheit und Compliance – In diesem Whitepaper wird beschrieben, wie Unternehmen mithilfe von AWS HIPAA-konforme Anwendungen erstellen können.
- AWS-Compliance-Ressourcen – Diese Arbeitsbücher und Leitfäden könnten für Ihre Branche und Ihren Standort relevant sein.
- Auswertung von Ressourcen mit Regeln im AWS ConfigEntwicklerhandbuch – Der AWS Config-Service bewertet, wie gut Ihre Ressourcenkonfigurationen mit internen Praktiken, Branchenrichtlinien und Vorschriften übereinstimmen.

- AWS Security Hub CSPM – Dieser AWS-Service liefert einen umfassenden Überblick über den Sicherheitsstatus in AWS. So können Sie die Compliance mit den Sicherheitsstandards in der Branche und den bewährten Methoden abgleichen.

Ausfallsicherheit in AWS Systems Manager

Die globale AWS-Infrastruktur ist um AWS-Regionen und Availability Zones herum aufgebaut. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die mit einem Netzwerk mit geringer Latenz, hohem Durchsatz und hoher Redundanz verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen über AWS-Regionen und Availability Zones finden Sie unter Globale AWS-Infrastruktur.

Sicherheit der Infrastruktur in AWS Systems Manager

Als verwalteter Dienst AWS Systems Manager ist er durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter AWS Cloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe für den Zugriff Systems Manager über das Netzwerk. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Konfigurations- und Schwachstellenanalyse in AWS Systems Manager

AWS übernimmt grundlegende Sicherheitsaufgaben wie Firewall-Konfiguration und Notfallwiederherstellung. Diese Verfahren wurden von qualifizierten Dritten überprüft und zertifiziert. Weitere Informationen finden Sie in den folgenden Ressourcen:

- Compliance-Validierung für AWS Systems Manager
- Modell der übergreifenden Verantwortlichkeit
- Bewährte Methoden für Sicherheit, Identität und Compliance

Bewährte Methoden für die Sicherheit für Systems Manager

AWS Systems Manager bietet eine Reihe von Sicherheitsfunktionen, die Sie bei der Entwicklung und Implementierung Ihrer eigenen Sicherheitsrichtlinien berücksichtigen sollten. Die folgenden bewährten Methoden sind allgemeine Richtlinien und keine vollständige Sicherheitslösung. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Überlegungen und nicht als bindend ansehen.

Themen

- Bewährte Methoden für vorbeugende Systems Manager-Sicherheitsmaßnahmen
- Bewährte Methoden für die Installation von SSM Agent
- Bewährte Methoden zur Überwachung und Prüfung von Systems Manager

Bewährte Methoden für vorbeugende Systems Manager-Sicherheitsmaßnahmen

Die folgenden bewährten Methoden für Systems Manager können dabei helfen, Sicherheitsvorfälle zu verhindern.

Implementieren des Zugriffs mit geringsten Berechtigungen

> Beim Erteilen von Berechtigungen entscheiden Sie, wer welche Berechtigungen für welche Systems Manager-Ressourcen erhält. Sie aktivieren die spezifischen Aktionen, die daraufhin für die betreffenden Ressourcen erlaubt sein sollen. Aus diesem Grund sollten

Sie nur Berechtigungen gewähren, die zum Ausführen einer Aufgabe erforderlich sind. Die Implementierung der geringstmöglichen Zugriffsrechte ist eine grundlegende Voraussetzung zum Reduzieren des Sicherheitsrisikos und der Auswirkungen, die aufgrund von Fehlern oder böswilligen Absichten entstehen könnten.

Die folgenden Tools stehen zur Implementierung der geringstmöglichen Zugriffsrechte zur Verfügung:

- IAM-Richtlinien und Berechtigungsgrenzen für IAM-Entitäten
- Service-Kontrollrichtlinien

Verwenden Sie die empfohlenen Einstellungen für SSM Agent, wenn Sie für die Verwendung eines Proxys konfiguriert sind

Wenn Sie SSM Agent eines Proxys konfigurieren, verwenden Sie die `no_proxy`-Variable mit der IP-Adresse des Metadaten-Services der Systems-Manager-Instance, um sicherzustellen, dass Aufrufe von Systems Manager nicht die Identität des Proxyservices annehmen.

Weitere Informationen erhalten Sie unter Konfigurieren Sie SSM Agent, um einen Proxy in Linux-Knoten zu verwenden und Konfigurieren des SSM Agent zur Nutzung eines Proxys für Windows Server-Instances.

Verwenden Sie SecureString Parameter, um geheime Daten zu verschlüsseln und zu schützen

In Parameter Store, ein Tool in AWS Systems Manager, kann ein `SecureString`-Parameter aus beliebigen vertraulichen Daten bestehen, die auf sichere Weise gespeichert und referenziert werden müssen. Wenn Sie Daten haben, die Benutzer nicht ändern oder im Klartext referenzieren sollen, z. B. Passwörter oder Lizenzschlüssel, erstellen Sie diese Parameter mithilfe des `SecureString` Datentyps. Parameter Storeverwendet ein AWS KMS key in AWS Key Management Service (AWS KMS), um den Parameterwert zu verschlüsseln. AWS KMS verwendet Von AWS verwalteter Schlüssel beim Verschlüsseln des Parameterwerts entweder einen vom Kunden verwalteten Schlüssel oder einen. Für maximale Sicherheit empfehlen wir die Verwendung eines eigenen KMS-Schlüssel. Wenn Sie den verwenden Von AWS verwalteter Schlüssel, kann jeder Benutzer, der berechtigt ist, die GetParameterGetParametersAND-Aktionen in Ihrem Konto auszuführen, den Inhalt aller `SecureString` Parameter anzeigen oder abrufen. Wenn Sie vom Kunden verwaltete Schlüssel zur Verschlüsselung Ihrer sicheren `SecureString`-Werte verwenden, können Sie IAM-Richtlinien und -Schlüsselrichtlinien verwenden, um die Berechtigungen für die Ver- und Entschlüsselung von Parametern zu verwalten.

Es ist schwieriger, Richtlinien für die Zugriffssteuerung für diese Vorgänge zu erstellen, wenn Sie Von AWS verwalteter Schlüssel verwenden. Wenn Sie beispielsweise einen Von AWS

verwalteter Schlüssel zum Verschlüsseln von `SecureString` Parametern verwenden und nicht möchten, dass Benutzer mit `SecureString` Parametern arbeiten, müssen die IAM-Richtlinien des Benutzers den Zugriff auf den Standardschlüssel ausdrücklich verweigern.

Weitere Informationen finden Sie unter Einschränken des Zugriffs auf Parameter Store-Parameter mithilfe von IAM-Richtlinien und How AWS Systems ManagerParameter Store Uses AWS KMS in the AWS Key Management Service Developer Guide.

Definieren von allowedValues und allowedPattern für Dokumentparameter

Sie können Benutzereingaben für Parameter in Systems Manager-Dokumenten (SSM-Dokumenten) validieren, indem Sie `allowedValues` und `allowedPattern` definieren. Für `allowedValues` definieren Sie ein Array von Werten, die für den Parameter zulässig sind. Wenn ein Benutzer einen Wert eingibt, der nicht zulässig ist, kann die Ausführung nicht gestartet werden. Für `allowedPattern` definieren Sie einen regulären Ausdruck, der überprüft, ob die Benutzereingabe mit dem definierten Muster für den Parameter übereinstimmt. Wenn die Benutzereingabe nicht mit dem zulässigen Muster übereinstimmt, kann die Ausführung nicht gestartet werden.

Weitere Informationen zu `allowedValues` und `allowedPattern` finden Sie unter Datenelemente und Parameter.

Öffentliche Freigabe für Dokumente blockieren

Sofern für Ihren Anwendungsfall keine öffentliche Freigabe erforderlich ist, empfehlen wir Ihnen, die Einstellung zum Blockieren der öffentlichen Freigabe für Ihre SSM-Dokumente im Abschnitt Preferences (Einstellungen) der Systems Manager-Dokumentenkonsole zu aktivieren.

Amazon Virtual Private Cloud (Amazon VPC) und VPC-Endpunkte verwenden

Sie können Amazon VPC verwenden, um AWS Ressourcen in einem von Ihnen definierten virtuellen Netzwerk bereitzustellen. Dieses virtuelle Netzwerk entspricht weitgehend einem herkömmlichen Netzwerk, wie Sie es in Ihrem Rechenzentrum betreiben, kann jedoch die Vorzüge der skalierbaren Infrastruktur von AWS nutzen.

Durch die Implementierung eines VPC-Endpunkts können Sie Ihre VPC privat mit unterstützten AWS-Services und unterstützten VPC-Endpunktdiensten verbinden, AWS PrivateLink ohne dass ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder eine Verbindung erforderlich ist. AWS Direct Connect Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um mit Ressourcen im Service zu kommunizieren. Der Datenverkehr zwischen Ihrer VPC und dem anderen Service verlässt das Amazon-Netzwerk nicht.

Weitere Informationen zur Sicherheit von Amazon VPC finden Sie unter Verbessern der Sicherheit von EC2 Instances durch die Verwendung von VPC-Endpunkten für Systems Manager und Schutz des Netzwerkverkehrs in Amazon VPC im Amazon VPC-Benutzerhandbuch.

Beschränken Sie Session Manager-Benutzer auf Sitzungen mit interaktiven Befehlen und bestimmten SSM-Sitzungsdokumenten

Session Manager, ein Tool in AWS Systems Manager, bietet mehrere Methoden zum Starten von Sitzungen für Ihre verwalteten Knoten. Für die sichersten Verbindungen können Sie von den Benutzern verlangen, dass sie sich mit der Methode interaktive Befehle verbinden, um die Benutzerinteraktion auf einen bestimmten Befehl oder eine bestimmte Befehlssequenz zu beschränken. Dies hilft Ihnen bei der Verwaltung der interaktiven Aktionen, die ein Benutzer durchführen kann. Weitere Informationen finden Sie unter Starten einer Sitzung (interaktive und nicht interaktive Befehle).

Für zusätzliche Sicherheit können Sie den Session Manager Zugriff auf bestimmte EC2 Amazon-Instances und bestimmte Session Manager Sitzungsdokumente einschränken. Sie gewähren oder widerrufen Session Manager den Zugriff auf diese Weise mithilfe von AWS Identity and Access Management (IAM-) Richtlinien. Weitere Informationen finden Sie unter Schritt 3: Steuern des Sitzungs-Zugriffs auf verwaltete Knoten.

Bereitstellen von temporären Knoten-Berechtigungen für Automatisierungs-Workflows

Während eines Workflows in Automation, ein Tool in AWS Systems Manager, benötigen Ihre Knoten möglicherweise Berechtigungen, die nur für diese Ausführung, nicht aber für andere Systems Manager-Vorgänge benötigt werden. Für einen Automatisierungs-Workflow kann es beispielsweise erforderlich sein, dass ein Knoten während des Workflows eine bestimmte API-Operation aufruft oder auf eine AWS Ressource zugreift. Wenn diese Aufrufe oder Ressourcen solche sind, auf die Sie den Zugriff beschränken möchten, können Sie temporäre, zusätzliche Berechtigungen für Ihre Knoten im Automatisierungs-Runbook selbst bereitstellen, anstatt die Berechtigungen zu Ihrem IAM-Instance-Profil hinzuzufügen. Am Ende des Automation-Workflows werden die temporären Berechtigungen entfernt. Weitere Informationen finden Sie unter Bereitstellung temporärer Instance-Berechtigungen mit AWS Systems Manager Automations im AWS Management- und Governance-Blog.

Halten Sie AWS die Systems Manager Tools auf dem neuesten Stand

AWS veröffentlicht regelmäßig aktualisierte Versionen von Tools und Plugins, die Sie in Ihren AWS Systems Manager Betriebsabläufen verwenden können. Wenn Sie diese Ressourcen auf dem neuesten Stand halten, wird sichergestellt, dass Benutzer und Knoten in Ihrem Konto Zugriff auf die neueste Funktion und Sicherheitsfeatures dieser Tools haben.

- SSM Agent— AWS Systems Manager Agent (SSM Agent) ist Amazon-Software, die auf einer Amazon Elastic Compute Cloud (Amazon EC2) -Instance, einem lokalen Server oder einer virtuellen Maschine (VM) installiert und konfiguriert werden kann. SSM Agentermöglicht die AktualisierungSystems Manager, Verwaltung und Konfiguration dieser Ressourcen. Es wird empfohlen, mindestens alle zwei Wochen nach neuen Versionen zu suchen oder Aktualisierungen des Agenten zu automatisieren. Weitere Informationen finden Sie unter Automatisieren von Updates für SSM Agent. Wir empfehlen außerdem, die Signatur von SSM Agent im Rahmen Ihres Aktualisierungsprozesses zu überprüfen. Weitere Informationen finden Sie unter Verifizieren der Signatur von SSM Agent.
- AWS CLI — The AWS Command Line Interface (AWS CLI) ist ein Open-Source-Tool, mit dem Sie AWS-Services mithilfe von Befehlen in Ihrer Befehlszeilen-Shell interagieren können. Um das zu aktualisieren AWS CLI, führen Sie denselben Befehl aus, mit dem Sie das installiert haben. AWS CLI Wir empfehlen, mindestens alle zwei Wochen eine geplante Aufgabe auf Ihrem lokalen Rechner zu erstellen, um den für Ihr Betriebssystem geeigneten Befehl auszuführen. Informationen zu Installationsbefehlen finden Sie im AWS Command Line Interface Benutzerhandbuch unter Installation der AWS CLI Version 2.
- AWS Tools for Windows PowerShell — Die Tools für Windows PowerShell sind eine Reihe von PowerShell Modulen, die auf der Funktionalität aufbauen, die das AWS SDK for .NET. Sie AWS Tools for Windows PowerShell ermöglichen es Ihnen, Operationen auf Ihren AWS Ressourcen von der PowerShell Befehlszeile aus per Skript auszuführen. Wenn aktualisierte Versionen der Tools für Windows veröffentlicht PowerShell werden, sollten Sie regelmäßig die Version aktualisieren, die Sie lokal ausführen. Weitere Informationen finden Sie unter Aktualisieren von AWS Tools for Windows PowerShell unter Windows oder Aktualisieren von unter Linux oder macOS im IAM Policy Simulator-Benutzerhandbuch. AWS Tools for Windows PowerShell
- Session Manager-Plugin – Wenn Benutzer in Ihrer Organisation mit den Berechtigungen für die Verwendung von Session Manager eine Verbindung zu einem Knoten mit AWS CLI herstellen möchten, müssen sie zuerst Session Manager in ihren lokalen Maschinen installieren. Um das Plugin zu aktualisieren, führen Sie denselben Befehl aus, der für die Installation des Plugins verwendet wird. Wir empfehlen, mindestens alle zwei Wochen eine geplante Aufgabe auf Ihrem lokalen Rechner zu erstellen, um den für Ihr Betriebssystem geeigneten Befehl auszuführen. Weitere Informationen finden Sie unter Installieren des Session Manager-Plugin für die AWS CLI.
- CloudWatch Agent — Sie können den CloudWatch Agenten konfigurieren und verwenden, um Metriken und Protokolle von Ihren EC2 Instanzen, lokalen Instanzen und virtuellen Maschinen zu sammeln ()VMs. Diese Protokolle können zur Überwachung und Analyse an Amazon CloudWatch Logs gesendet werden. Es wird empfohlen, mindestens alle zwei Wochen nach

neuen Versionen zu suchen oder Aktualisierungen des Agenten zu automatisieren. Für die einfachsten Aktualisierungen verwenden Sie AWS Systems Manager Quick Setup. Weitere Informationen finden Sie unter AWS Systems Manager Quick Setup.

Bewährte Methoden für die Installation von SSM Agent

Verwenden Sie bei der Installation von SSM Agent die für Ihren Maschinentyp geeignete Installationsmethode. Verwenden Sie das `ssm-setup-cli` Tool insbesondere für alle EC2 Nichtinstallationen in einer Hybrid- und Multi-Cloud-Umgebung. Dieses Tool bietet zusätzlichen Sicherheitsschutz für Computer, die keine Computer sind. EC2

Verwenden Sie das `ssm-setup-cli` Tool, um den Agenten auf On-PremisesServern und virtuellen Maschinen zu installieren, wie in den folgenden Themen beschrieben:

- the section called “Installation von SSM Agent auf hybriden Linux-Knoten”
- the section called “Installation von SSM Agent auf hybriden Windows Server-Knoten”

Um den Agenten auf EC2 Instanzen zu installieren, verwenden Sie das für Ihren Betriebssystemtyp geeignete Installationsverfahren:

- the section called “Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux”
- the section called “Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für macOS”
- the section called “Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Windows Server”

Bewährte Methoden zur Überwachung und Prüfung von Systems Manager

Mithilfe der folgenden bewährten Methoden für Systems Manager können Sie potenzielle Sicherheitsschwächen und Vorfälle erkennen.

Identifizieren und prüfen all Ihrer Systems Manager-Ressourcen

Die Identifikation Ihrer IT-Assets ist ein wichtiger Aspekt von Governance und Sicherheit. Sie müssen alle Ihre Systems Manager-Ressourcen identifizieren, um ihre Sicherheitssituation zu bewerten und Maßnahmen in potentiellen Schwachstellenbereichen zu ergreifen.

Verwenden Sie den Tag-Editor, um sicherheits- und prüfungsrelevante Ressourcen zu identifizieren. Verwenden Sie dann diese Markierungen zur Suche nach den entsprechenden Ressourcen. Weitere Informationen finden Sie unter Suchen nach zu markierenden Ressourcen im AWS -Ressourcengruppen -Benutzerhandbuch.

Erstellen Sie Ressourcengruppen für Ihre Systems Manager-Ressourcen. Weitere Informationen finden Sie unter Was sind Ressourcengruppen?

Implementieren Sie die Überwachung mithilfe der CloudWatch Amazon-Überwachungstools

Die Überwachung ist ein wichtiger Teil der Aufrechterhaltung von Zuverlässigkeit, Sicherheit, Verfügbarkeit und Performance von Systems Manager und Ihren AWS -Lösungen. Amazon CloudWatch bietet verschiedene Tools und Dienste, die Sie bei der Überwachung Systems Manager und Ihrer anderen unterstützen AWS-Services. Weitere Informationen erhalten Sie unter Senden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent) und Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge.

Benutzen CloudTrail

AWS CloudTrail bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS-Service Mitglied ausgeführt wurdenSystems Manager. Anhand der von gesammelten Informationen können Sie die Anfrage ermitteln CloudTrail, an die die Anfrage gestellt wurdeSystems Manager, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Informationen. Weitere Informationen finden Sie unter Protokollierung von AWS Systems Manager-API-Aufrufen mit AWS CloudTrail.

Einschalten AWS Config

AWS Config ermöglicht es Ihnen, die Konfigurationen Ihrer AWS Ressourcen zu bewerten, zu prüfen und zu bewerten. AWS Config überwacht die Ressourcenkonfigurationen, sodass Sie die aufgezeichneten Konfigurationen mit den erforderlichen sicheren Konfigurationen vergleichen können. Mithilfe AWS Config dieser Funktion können Sie Änderungen an Konfigurationen und Beziehungen zwischen AWS Ressourcen überprüfen, den detaillierten Verlauf der Ressourcenkonfigurationen untersuchen und die allgemeine Konformität mit den in Ihren internen Richtlinien festgelegten Konfigurationen ermitteln. Dadurch können Sie die Compliance-Prüfung, die Sicherheitsanalyse, das Änderungsmanagement und die Fehlerbehebung bei Betriebsabläufen vereinfachen. Weitere Informationen finden Sie unter Einrichten von AWS Config mit der Konsole im AWS Config -Entwicklerhandbuch. Achten Sie bei der Angabe der Ressourcentypen, die aufgezeichnet werden sollen, darauf, dass Systems Manager-Ressourcen enthalten sind.

Überwachen Sie die AWS Sicherheitsempfehlungen

Sie sollten regelmäßig die Trusted Advisor für Sie veröffentlichten Sicherheitshinweise überprüfen. AWS-Konto Sie können dies programmgesteuert tun mit. describe-trusted-advisor-checks

Überwachen Sie außerdem aktiv die primäre E-Mail-Adresse, die für jeden von Ihnen registriert ist. AWS-Konten AWS wird Sie unter Verwendung dieser E-Mail-Adresse über neu auftretende Sicherheitsprobleme kontaktieren, die Sie betreffen könnten.

AWS Betriebsprobleme mit weitreichenden Auswirkungen werden im AWS Service Health Dashboard veröffentlicht. Operative Probleme werden ebenfalls über das Personal Health Dashboard in den einzelnen Konten gepostet. Weitere Informationen finden Sie in der AWS Health Dokumentation.

Weitere Informationen

- Bewährte Methoden für Sicherheit, Identität und Compliance
- Erste Schritte: Halten Sie sich bei der Konfiguration Ihrer AWS Ressourcen an bewährte Sicherheitsmethoden (AWS Sicherheitsblog)
- Bewährte Methoden für die Sicherheit in IAM
- Bewährte Sicherheitsmethoden in AWS CloudTrail
- Bewährte Methoden für die Sicherheit in Simple Storage Service (Amazon S3)
- Bewährte Sicherheitsmethoden für AWS Key Management Service

Codebeispiele für Systems Manager mit AWS SDKs

Die folgenden Codebeispiele zeigen, wie Systems Manager mit einem AWS Software Development Kit (SDK) verwendet wird.

Bei Grundlagen handelt es sich um Codebeispiele, die Ihnen zeigen, wie Sie die wesentlichen Vorgänge innerhalb eines Services ausführen.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Während Aktionen Ihnen zeigen, wie Sie einzelne Service-Funktionen aufrufen, können Sie Aktionen im Kontext der zugehörigen Szenarien anzeigen.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden dieses Dienstes mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erste Schritte

Hallo Systems Manager

Die folgenden Codebeispiele veranschaulichen, wie Sie mit der Verwendung von Systems Manager beginnen.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.ssm.SsmClient;
import software.amazon.awssdk.services.ssm.model.DocumentFilter;
import software.amazon.awssdk.services.ssm.model.ListDocumentsRequest;
import software.amazon.awssdk.services.ssm.model.ListDocumentsResponse;

public class HelloSSM {
```

```
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <awsAccount>

                Where:
                    awsAccount - Your AWS Account number.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String awsAccount = args[0] ;
        Region region = Region.US_EAST_1;
        SsmClient ssmClient = SsmClient.builder()
            .region(region)
            .build();

        listDocuments(ssmClient, awsAccount);
    }

    /*
    This code automatically fetches the next set of results using the `nextToken`
and
    stops once the desired maxResults (20 in this case) have been reached.
    */
    public static void listDocuments(SsmClient ssmClient, String awsAccount) {
        String nextToken = null;
        int totalDocumentsReturned = 0;
        int maxResults = 20;
        do {
            ListDocumentsRequest request = ListDocumentsRequest.builder()
                .documentFilterList(
                    DocumentFilter.builder()
                        .key("Owner")
                        .value(awsAccount)
                        .build()
                    )
                .maxResults(maxResults)
                .nextToken(nextToken)
```

```
                .build();

            ListDocumentsResponse response = ssmClient.listDocuments(request);
            response.documentIdentifiers().forEach(identifier ->
 System.out.println("Document Name: " + identifier.name()));
            nextToken = response.nextToken();
            totalDocumentsReturned += response.documentIdentifiers().size();
        } while (nextToken != null && totalDocumentsReturned < maxResults);
    }
}
```

- Einzelheiten zur API finden Sie ListDocumentsin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import { paginateListDocuments, SSMClient } from "@aws-sdk/client-ssm";

// Call ListDocuments and display the result.
export const main = async () => {
  const client = new SSMClient();
  const listDocumentsPaginated = [];
  console.log(
    "Hello, AWS Systems Manager! Let's list some of your documents:\n",
  );
  try {
    // The paginate function is a wrapper around the base command.
    const paginator = paginateListDocuments({ client }, { MaxResults: 5 });
    for await (const page of paginator) {
      listDocumentsPaginated.push(...page.DocumentIdentifiers);
    }
  } catch (caught) {
    console.error(`There was a problem saying hello: ${caught.message}`);
    throw caught;
```

```
  }

  for (const { Name, DocumentFormat, CreatedDate } of listDocumentsPaginated) {
    console.log(`${Name} - ${DocumentFormat} - ${CreatedDate}`);
  }
};

// Call function if run directly.
import { fileURLToPath } from "node:url";
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main();
}
```

- Einzelheiten zur API finden Sie ListDocumentsin der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import boto3
from botocore.exceptions import ClientError


def hello_systems_manager(ssm_client):
    """
    Use the AWS SDK for Python (Boto3) to create an AWS Systems Manager
    client and list the first 5 documents in your account.
    This example uses the default settings specified in your shared credentials
    and config files.

    :param ssm_client: A Boto3 AWS Systems Manager Client object. This object
 wraps
                           the low-level AWS Systems Manager service API.
    """
    print("Hello, AWS Systems Manager! Let's list some of your documents:\n")
```

```
    paginator = ssm_client.get_paginator("list_documents")
    page_iterator = paginator.paginate(PaginationConfig={"MaxItems": 5})
    for page in page_iterator:
        for document in page["DocumentIdentifiers"]:
            print(f"  {document['Name']}")


if __name__ == "__main__":
    try:
        hello_systems_manager(boto3.client("ssm"))
    except ClientError as err:
        print("Hello systems manager had an error.")
        print(err.response["Error"]["Code"])
        print(err.response["Error"]["Message"])
```

- Einzelheiten zur API finden Sie ListDocumentsin AWS SDK for Python (Boto3) API Reference.

Codebeispiele

- Grundlegende Beispiele für die Verwendung von Systems Manager AWS SDKs
  - Hallo Systems Manager
  - Erlernen Sie die Grundlagen von Systems Manager mit einem AWS SDK
  - Aktionen für Systems Manager mit AWS SDKs
    - Verwendung von AddTagsToResource mit einer CLI
    - Verwendung von CancelCommand mit einer CLI
    - Verwendung von CreateActivation mit einer CLI
    - Verwendung von CreateAssociation mit einer CLI
    - Verwendung von CreateAssociationBatch mit einer CLI
    - Verwendung CreateDocument mit einem AWS SDK oder CLI
    - Verwendung CreateMaintenanceWindow mit einem AWS SDK oder CLI
    - Verwendung CreateOpsItem mit einem AWS SDK oder CLI
    - Verwendung von CreatePatchBaseline mit einer CLI
    - Verwendung von DeleteActivation mit einer CLI
    - Verwendung von DeleteAssociation mit einer CLI

- Verwendung DeleteDocument mit einem AWS SDK oder CLI
- Verwendung DeleteMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung DeleteOpsItem mit einem AWS SDK
- Verwendung von DeleteParameter mit einer CLI
- Verwendung von DeletePatchBaseline mit einer CLI
- Verwendung von DeregisterManagedInstance mit einer CLI
- Verwendung von DeregisterPatchBaselineForPatchGroup mit einer CLI
- Verwendung von DeregisterTargetFromMaintenanceWindow mit einer CLI
- Verwendung von DeregisterTaskFromMaintenanceWindow mit einer CLI
- Verwendung von DescribeActivations mit einer CLI
- Verwendung von DescribeAssociation mit einer CLI
- Verwendung von DescribeAssociationExecutionTargets mit einer CLI
- Verwendung von DescribeAssociationExecutions mit einer CLI
- Verwendung von DescribeAutomationExecutions mit einer CLI
- Verwendung von DescribeAutomationStepExecutions mit einer CLI
- Verwendung von DescribeAvailablePatches mit einer CLI
- Verwendung von DescribeDocument mit einer CLI
- Verwendung von DescribeDocumentPermission mit einer CLI
- Verwendung von DescribeEffectiveInstanceAssociations mit einer CLI
- Verwendung von DescribeEffectivePatchesForPatchBaseline mit einer CLI
- Verwendung von DescribeInstanceAssociationsStatus mit einer CLI
- Verwendung von DescribeInstanceInformation mit einer CLI
- Verwendung von DescribeInstancePatchStates mit einer CLI
- Verwendung von DescribeInstancePatchStatesForPatchGroup mit einer CLI
- Verwendung von DescribeInstancePatches mit einer CLI
- Verwendung von DescribeMaintenanceWindowExecutionTaskInvocations mit einer CLI
- Verwendung von DescribeMaintenanceWindowExecutionTasks mit einer CLI
- Verwendung von DescribeMaintenanceWindowExecutions mit einer CLI
- Verwendung von DescribeMaintenanceWindowTargets mit einer CLI

- Verwendung von DescribeMaintenanceWindowTasks mit einer CLI

- Verwendung von DescribeMaintenanceWindows mit einer CLI
- Verwendung DescribeOpsItems mit einem AWS SDK oder CLI
- Verwendung DescribeParameters mit einem AWS SDK oder CLI
- Verwendung von DescribePatchBaselines mit einer CLI
- Verwendung von DescribePatchGroupState mit einer CLI
- Verwendung von DescribePatchGroups mit einer CLI
- Verwendung von GetAutomationExecution mit einer CLI
- Verwendung von GetCommandInvocation mit einer CLI
- Verwendung von GetConnectionStatus mit einer CLI
- Verwendung von GetDefaultPatchBaseline mit einer CLI
- Verwendung von GetDeployablePatchSnapshotForInstance mit einer CLI
- Verwendung von GetDocument mit einer CLI
- Verwendung von GetInventory mit einer CLI
- Verwendung von GetInventorySchema mit einer CLI
- Verwendung von GetMaintenanceWindow mit einer CLI
- Verwendung von GetMaintenanceWindowExecution mit einer CLI
- Verwendung von GetMaintenanceWindowExecutionTask mit einer CLI
- Verwendung GetParameter mit einem AWS SDK oder CLI
- Verwendung von GetParameterHistory mit einer CLI
- Verwendung von GetParameters mit einer CLI
- Verwendung von GetPatchBaseline mit einer CLI
- Verwendung von GetPatchBaselineForPatchGroup mit einer CLI
- Verwendung von ListAssociationVersions mit einer CLI
- Verwendung von ListAssociations mit einer CLI
- Verwendung ListCommandInvocations mit einem AWS SDK oder CLI
- Verwendung von ListCommands mit einer CLI
- Verwendung von ListComplianceItems mit einer CLI
- Verwendung von ListComplianceSummaries mit einer CLI
- Verwendung von ListDocumentVersions mit einer CLI

- Verwendung von ListDocuments mit einer CLI

- Verwendung von ListInventoryEntries mit einer CLI
- Verwendung von ListResourceComplianceSummaries mit einer CLI
- Verwendung von ListTagsForResource mit einer CLI
- Verwendung von ModifyDocumentPermission mit einer CLI
- Verwendung von PutComplianceItems mit einer CLI
- Verwendung von PutInventory mit einer CLI
- Verwendung PutParameter mit einem AWS SDK oder CLI
- Verwendung von RegisterDefaultPatchBaseline mit einer CLI
- Verwendung von RegisterPatchBaselineForPatchGroup mit einer CLI
- Verwendung von RegisterTargetWithMaintenanceWindow mit einer CLI
- Verwendung von RegisterTaskWithMaintenanceWindow mit einer CLI
- Verwendung von RemoveTagsFromResource mit einer CLI
- Verwendung SendCommand mit einem AWS SDK oder CLI
- Verwendung von StartAutomationExecution mit einer CLI
- Verwendung von StartSession mit einer CLI
- Verwendung von StopAutomationExecution mit einer CLI
- Verwendung von UpdateAssociation mit einer CLI
- Verwendung von UpdateAssociationStatus mit einer CLI
- Verwendung von UpdateDocument mit einer CLI
- Verwendung von UpdateDocumentDefaultVersion mit einer CLI
- Verwendung UpdateMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung von UpdateManagedInstanceRole mit einer CLI
- Verwendung UpdateOpsItem mit einem AWS SDK oder CLI
- Verwendung von UpdatePatchBaseline mit einer CLI

Grundlegende Beispiele für die Verwendung von Systems Manager AWS SDKs

Die folgenden Codebeispiele zeigen, wie die Grundlagen von AWS Systems Manager with verwendet AWS SDKs werden.

Beispiele

- Hallo Systems Manager
- Erlernen Sie die Grundlagen von Systems Manager mit einem AWS SDK
- Aktionen für Systems Manager mit AWS SDKs
  - Verwendung von AddTagsToResource mit einer CLI
  - Verwendung von CancelCommand mit einer CLI
  - Verwendung von CreateActivation mit einer CLI
  - Verwendung von CreateAssociation mit einer CLI
  - Verwendung von CreateAssociationBatch mit einer CLI
  - Verwendung CreateDocument mit einem AWS SDK oder CLI
  - Verwendung CreateMaintenanceWindow mit einem AWS SDK oder CLI
  - Verwendung CreateOpsItem mit einem AWS SDK oder CLI
  - Verwendung von CreatePatchBaseline mit einer CLI
  - Verwendung von DeleteActivation mit einer CLI
  - Verwendung von DeleteAssociation mit einer CLI
  - Verwendung DeleteDocument mit einem AWS SDK oder CLI
  - Verwendung DeleteMaintenanceWindow mit einem AWS SDK oder CLI
  - Verwendung DeleteOpsItem mit einem AWS SDK
  - Verwendung von DeleteParameter mit einer CLI
  - Verwendung von DeletePatchBaseline mit einer CLI
  - Verwendung von DeregisterManagedInstance mit einer CLI
  - Verwendung von DeregisterPatchBaselineForPatchGroup mit einer CLI
  - Verwendung von DeregisterTargetFromMaintenanceWindow mit einer CLI
  - Verwendung von DeregisterTaskFromMaintenanceWindow mit einer CLI
  - Verwendung von DescribeActivations mit einer CLI
  - Verwendung von DescribeAssociation mit einer CLI
  - Verwendung von DescribeAssociationExecutionTargets mit einer CLI
  - Verwendung von DescribeAssociationExecutions mit einer CLI
  - Verwendung von DescribeAutomationExecutions mit einer CLI
  - Verwendung von DescribeAutomationStepExecutions mit einer CLI

- Verwendung von DescribeAvailablePatches mit einer CLI
- Verwendung von DescribeDocument mit einer CLI
- Verwendung von DescribeDocumentPermission mit einer CLI
- Verwendung von DescribeEffectiveInstanceAssociations mit einer CLI
- Verwendung von DescribeEffectivePatchesForPatchBaseline mit einer CLI
- Verwendung von DescribeInstanceAssociationsStatus mit einer CLI
- Verwendung von DescribeInstanceInformation mit einer CLI
- Verwendung von DescribeInstancePatchStates mit einer CLI
- Verwendung von DescribeInstancePatchStatesForPatchGroup mit einer CLI
- Verwendung von DescribeInstancePatches mit einer CLI
- Verwendung von DescribeMaintenanceWindowExecutionTaskInvocations mit einer CLI
- Verwendung von DescribeMaintenanceWindowExecutionTasks mit einer CLI
- Verwendung von DescribeMaintenanceWindowExecutions mit einer CLI
- Verwendung von DescribeMaintenanceWindowTargets mit einer CLI
- Verwendung von DescribeMaintenanceWindowTasks mit einer CLI
- Verwendung von DescribeMaintenanceWindows mit einer CLI
- Verwendung DescribeOpsItems mit einem AWS SDK oder CLI
- Verwendung DescribeParameters mit einem AWS SDK oder CLI
- Verwendung von DescribePatchBaselines mit einer CLI
- Verwendung von DescribePatchGroupState mit einer CLI
- Verwendung von DescribePatchGroups mit einer CLI
- Verwendung von GetAutomationExecution mit einer CLI
- Verwendung von GetCommandInvocation mit einer CLI
- Verwendung von GetConnectionStatus mit einer CLI
- Verwendung von GetDefaultPatchBaseline mit einer CLI
- Verwendung von GetDeployablePatchSnapshotForInstance mit einer CLI
- Verwendung von GetDocument mit einer CLI
- Verwendung von GetInventory mit einer CLI
- Verwendung von GetInventorySchema mit einer CLI
- Verwendung von GetMaintenanceWindow mit einer CLI

- Verwendung von GetMaintenanceWindowExecution mit einer CLI
- Verwendung von GetMaintenanceWindowExecutionTask mit einer CLI
- Verwendung GetParameter mit einem AWS SDK oder CLI
- Verwendung von GetParameterHistory mit einer CLI
- Verwendung von GetParameters mit einer CLI
- Verwendung von GetPatchBaseline mit einer CLI
- Verwendung von GetPatchBaselineForPatchGroup mit einer CLI
- Verwendung von ListAssociationVersions mit einer CLI
- Verwendung von ListAssociations mit einer CLI
- Verwendung ListCommandInvocations mit einem AWS SDK oder CLI
- Verwendung von ListCommands mit einer CLI
- Verwendung von ListComplianceItems mit einer CLI
- Verwendung von ListComplianceSummaries mit einer CLI
- Verwendung von ListDocumentVersions mit einer CLI
- Verwendung von ListDocuments mit einer CLI
- Verwendung von ListInventoryEntries mit einer CLI
- Verwendung von ListResourceComplianceSummaries mit einer CLI
- Verwendung von ListTagsForResource mit einer CLI
- Verwendung von ModifyDocumentPermission mit einer CLI
- Verwendung von PutComplianceItems mit einer CLI
- Verwendung von PutInventory mit einer CLI
- Verwendung PutParameter mit einem AWS SDK oder CLI
- Verwendung von RegisterDefaultPatchBaseline mit einer CLI
- Verwendung von RegisterPatchBaselineForPatchGroup mit einer CLI
- Verwendung von RegisterTargetWithMaintenanceWindow mit einer CLI
- Verwendung von RegisterTaskWithMaintenanceWindow mit einer CLI
- Verwendung von RemoveTagsFromResource mit einer CLI
- Verwendung SendCommand mit einem AWS SDK oder CLI
- Verwendung von StartAutomationExecution mit einer CLI

- Verwendung von StartSession mit einer CLI

- Verwendung von StopAutomationExecution mit einer CLI
- Verwendung von UpdateAssociation mit einer CLI
- Verwendung von UpdateAssociationStatus mit einer CLI
- Verwendung von UpdateDocument mit einer CLI
- Verwendung von UpdateDocumentDefaultVersion mit einer CLI
- Verwendung UpdateMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung von UpdateManagedInstanceRole mit einer CLI
- Verwendung UpdateOpsItem mit einem AWS SDK oder CLI
- Verwendung von UpdatePatchBaseline mit einer CLI

Hallo Systems Manager

Die folgenden Codebeispiele zeigen, wie Sie mit der Verwendung von Systems Manager beginnen.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.ssm.SsmClient;
import software.amazon.awssdk.services.ssm.model.DocumentFilter;
import software.amazon.awssdk.services.ssm.model.ListDocumentsRequest;
import software.amazon.awssdk.services.ssm.model.ListDocumentsResponse;

public class HelloSSM {

    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <awsAccount>
```

```
                Where:
                    awsAccount - Your AWS Account number.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String awsAccount = args[0] ;
        Region region = Region.US_EAST_1;
        SsmClient ssmClient = SsmClient.builder()
            .region(region)
            .build();

        listDocuments(ssmClient, awsAccount);
    }

    /*
    This code automatically fetches the next set of results using the `nextToken`
and
    stops once the desired maxResults (20 in this case) have been reached.
    */
    public static void listDocuments(SsmClient ssmClient, String awsAccount) {
        String nextToken = null;
        int totalDocumentsReturned = 0;
        int maxResults = 20;
        do {
            ListDocumentsRequest request = ListDocumentsRequest.builder()
                .documentFilterList(
                    DocumentFilter.builder()
                        .key("Owner")
                        .value(awsAccount)
                        .build()
                    )
                .maxResults(maxResults)
                .nextToken(nextToken)
                .build();

            ListDocumentsResponse response = ssmClient.listDocuments(request);
            response.documentIdentifiers().forEach(identifier ->
System.out.println("Document Name: " + identifier.name()));
            nextToken = response.nextToken();
            totalDocumentsReturned += response.documentIdentifiers().size();
```

```
        } while (nextToken != null && totalDocumentsReturned < maxResults);
    }
}
```

- Einzelheiten zur API finden Sie ListDocumentsin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import { paginateListDocuments, SSMClient } from "@aws-sdk/client-ssm";

// Call ListDocuments and display the result.
export const main = async () => {
  const client = new SSMClient();
  const listDocumentsPaginated = [];
  console.log(
    "Hello, AWS Systems Manager! Let's list some of your documents:\n",
  );
  try {
    // The paginate function is a wrapper around the base command.
    const paginator = paginateListDocuments({ client }, { MaxResults: 5 });
    for await (const page of paginator) {
      listDocumentsPaginated.push(...page.DocumentIdentifiers);
    }
  } catch (caught) {
    console.error(`There was a problem saying hello: ${caught.message}`);
    throw caught;
  }

  for (const { Name, DocumentFormat, CreatedDate } of listDocumentsPaginated) {
    console.log(`${Name} - ${DocumentFormat} - ${CreatedDate}`);
  }
};
```

```
// Call function if run directly.
import { fileURLToPath } from "node:url";
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main();
}
```

- Einzelheiten zur API finden Sie ListDocumentsin der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import boto3
from botocore.exceptions import ClientError


def hello_systems_manager(ssm_client):
    """
    Use the AWS SDK for Python (Boto3) to create an AWS Systems Manager
    client and list the first 5 documents in your account.
    This example uses the default settings specified in your shared credentials
    and config files.

    :param ssm_client: A Boto3 AWS Systems Manager Client object. This object
 wraps
                           the low-level AWS Systems Manager service API.
    """
    print("Hello, AWS Systems Manager! Let's list some of your documents:\n")

    paginator = ssm_client.get_paginator("list_documents")
    page_iterator = paginator.paginate(PaginationConfig={"MaxItems": 5})
    for page in page_iterator:
        for document in page["DocumentIdentifiers"]:
            print(f"  {document['Name']}")
```

```
if __name__ == "__main__":
    try:
        hello_systems_manager(boto3.client("ssm"))
    except ClientError as err:
        print("Hello systems manager had an error.")
        print(err.response["Error"]["Code"])
        print(err.response["Error"]["Message"])
```

- Einzelheiten zur API finden Sie ListDocumentsin AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erlernen Sie die Grundlagen von Systems Manager mit einem AWS SDK

Die folgenden Code-Beispiele veranschaulichen Folgendes:

- Erstellen Sie ein Wartungsfenster.
- Ändern Sie den Zeitplan des Wartungsfensters.
- Erstellen Sie ein Dokument.
- Sendet einen Befehl an eine angegebene EC2 Instanz.
- Erstellen Sie eine OpsItem.
- Aktualisieren und lösen Sie das OpsItem.
- Löschen Sie das Wartungsfenster OpsItem, und das Dokument.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import software.amazon.awssdk.services.ssm.model.DocumentAlreadyExistsException;
import software.amazon.awssdk.services.ssm.model.SsmException;

import java.util.Map;
import java.util.Scanner;
public class SSMScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");
    private static final String ROLES_STACK = "SsmStack3`1";

    public static void main(String[] args) {
        String usage = """
            Usage:
              <title> <source> <category> <severity>

            Where:
                title - The title of the parameter (default is Disk Space Alert).
                source - The source of the parameter (default is EC2).
                category - The category of the parameter. Valid values are
 'Availability', 'Cost', 'Performance', 'Recovery', 'Security' (default is
 Performance).
                severity - The severity of the parameter. Severity should be a
 number from 1 to 4 (default is 2).
        """;

        Scanner scanner = new Scanner(System.in);
        SSMActions actions = new SSMActions();
        String documentName;
        String windowName;

        System.out.println("Use AWS CloudFormation to create the EC2 instance
 that is required for this scenario.");
        CloudFormationHelper.deployCloudFormationStack(ROLES_STACK);
        Map<String, String> stackOutputs =
CloudFormationHelper.getStackOutputsAsync(ROLES_STACK).join();
        String instanceId = stackOutputs.get("InstanceId");
        System.out.println("The Instance ID: " + instanceId +" was created.");
        String title = "Disk Space Alert" ;
        String source = "EC2" ;
        String category = "Availability" ;
        String severity = "2" ;

        System.out.println(DASHES);
```

```
        System.out.println("""
                Welcome to the AWS Systems Manager SDK Basics scenario.
                This Java program demonstrates how to interact with AWS Systems
Manager using the AWS SDK for Java (v2).
                AWS Systems Manager is the operations hub for your AWS
applications and resources and a secure end-to-end management solution.
                The program's primary functionalities include creating a
maintenance window, creating a document, sending a command to a document,
                listing documents, listing commands, creating an OpsItem,
modifying an OpsItem, and deleting AWS SSM resources.
                Upon completion of the program, all AWS resources are cleaned up.
                Let's get started...

                """);
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println("1. Create an SSM maintenance window.");
        System.out.println("Please enter the maintenance window name (default is
ssm-maintenance-window):");
        String win = scanner.nextLine();
        windowName = win.isEmpty() ? "ssm-maintenance-window" : win;
        String winId = null;
        try {
            winId = actions.createMaintenanceWindow(windowName);
            waitForInputToContinue(scanner);
            System.out.println("The maintenance window ID is: " + winId);
        } catch (DocumentAlreadyExistsException e) {
            System.err.println("The SSM maintenance window already exists.
Retrieving existing window ID...");
            String existingWinId = actions.createMaintenanceWindow(windowName);
            System.out.println("Existing window ID: " + existingWinId);
        } catch (SsmException e) {
            System.err.println("SSM error: " + e.getMessage());
            return;
        } catch (RuntimeException e) {
            System.err.println("Unexpected error: " + e.getMessage());
            return;
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println("2. Modify the maintenance window by changing the
schedule");
```

```
        waitForInputToContinue(scanner);
        try {
            actions.updateSSMMaintenanceWindow(winId, windowName);
            waitForInputToContinue(scanner);
            System.out.println("The SSM maintenance window was successfully
updated");
        } catch (SsmException e) {
            System.err.println("SSM error: " + e.getMessage());
            return;
        } catch (RuntimeException e) {
            System.err.println("Unexpected error: " + e.getMessage());
            return;
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println("3. Create an SSM document that defines the actions
that Systems Manager performs on your managed nodes.");
        System.out.println("Please enter the document name (default is
ssmdocument):");
        String doc = scanner.nextLine();
        documentName = doc.isEmpty() ? "ssmdocument" : doc;
        try {
            actions.createSSMDoc(documentName);
            waitForInputToContinue(scanner);
            System.out.println("The SSM document was successfully created");
        } catch (DocumentAlreadyExistsException e) {
            System.err.println("The SSM document already exists. Moving on");
        } catch (SsmException e) {
            System.err.println("SSM error: " + e.getMessage());
            return;
        } catch (RuntimeException e) {
            System.err.println("Unexpected error: " + e.getMessage());
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println("4. Now we are going to run a command on an EC2
instance");
        waitForInputToContinue(scanner);
        String commandId="";
        try {
            commandId = actions.sendSSMCommand(documentName, instanceId);
            waitForInputToContinue(scanner);
```

```
            System.out.println("The command was successfully sent. Command ID: "
+ commandId);
        } catch (SsmException e) {
            System.err.println("SSM error: " + e.getMessage());
        } catch (InterruptedException e) {
            System.err.println("Thread was interrupted: " + e.getMessage());
        } catch (RuntimeException e) {
            System.err.println("Unexpected error: " + e.getMessage());
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println("5. Lets get the time when the specific command was
sent to the specific managed node");
        waitForInputToContinue(scanner);
        try {
            actions.displayCommands(commandId);
            System.out.println("The command invocations were successfully
displayed.");
        } catch (SsmException e) {
            System.err.println("SSM error: " + e.getMessage());
            return;
        } catch (RuntimeException e) {
            System.err.println("Unexpected error: " + e.getMessage());
            return;
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("""
             6. Now we will create an SSM OpsItem.
             A SSM OpsItem is a feature provided by Amazon's Systems Manager
(SSM) service.
             It is a type of operational data item that allows you to manage and
track various operational issues,
             events, or tasks within your AWS environment.

             You can create OpsItems to track and manage operational issues as
they arise.
             For example, you could create an OpsItem whenever your application
detects a critical error
             or an anomaly in your infrastructure.
            """);
```

```
        waitForInputToContinue(scanner);
        String opsItemId;
        try {
            opsItemId = actions.createSSMOpsItem(title, source, category,
severity);
            System.out.println(opsItemId + " was created");
        } catch (SsmException e) {
            System.err.println("SSM error: " + e.getMessage());
            return;
        } catch (RuntimeException e) {
            System.err.println("Unexpected error: " + e.getMessage());
            return;
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Now we will update the SSM OpsItem "+opsItemId);
        waitForInputToContinue(scanner);
        String description = "An update to "+opsItemId ;
        try {
            actions.updateOpsItem(opsItemId, title, description);
        } catch (SsmException e) {
            System.err.println("SSM error: " + e.getMessage());
            return;
        } catch (RuntimeException e) {
            System.err.println("Unexpected error: " + e.getMessage());
            return;
        }

        System.out.println(DASHES);
        System.out.println("8. Now we will get the status of the SSM OpsItem
"+opsItemId);
        waitForInputToContinue(scanner);
        try {
            actions.describeOpsItems(opsItemId);
        } catch (SsmException e) {
            System.err.println("SSM error: " + e.getMessage());
            return;
        } catch (RuntimeException e) {
            System.err.println("Unexpected error: " + e.getMessage());
            return;
        }
```

```
        System.out.println(DASHES);
        System.out.println("9. Now we will resolve the SSM OpsItem "+opsItemId);
        waitForInputToContinue(scanner);
        try {
            actions.resolveOpsItem(opsItemId);
        } catch (SsmException e) {
            System.err.println("SSM error: " + e.getMessage());
            return;
        } catch (RuntimeException e) {
            System.err.println("Unexpected error: " + e.getMessage());
            return;
        }

        System.out.println(DASHES);
        System.out.println("10. Would you like to delete the AWS Systems Manager
resources? (y/n)");
        String delAns = scanner.nextLine().trim();
        if (delAns.equalsIgnoreCase("y")) {
            System.out.println("You selected to delete the resources.");
            waitForInputToContinue(scanner);
            try {
                actions.deleteMaintenanceWindow(winId);
                actions.deleteDoc(documentName);
            } catch (SsmException e) {
                System.err.println("SSM error: " + e.getMessage());
                return;
            } catch (RuntimeException e) {
                System.err.println("Unexpected error: " + e.getMessage());
                return;
            }
        } else {
            System.out.println("The AWS Systems Manager resources will not be
deleted");
        }
        System.out.println(DASHES);
        CloudFormationHelper.destroyCloudFormationStack(ROLES_STACK);
        System.out.println("This concludes the AWS Systems Manager SDK Basics
scenario.");
        System.out.println(DASHES);
    }

    private static void waitForInputToContinue(Scanner scanner) {
        while (true) {
```

```
            System.out.println("");
            System.out.println("Enter 'c' followed by <ENTER> to continue:");
            String input = scanner.nextLine();

            if (input.trim().equalsIgnoreCase("c")) {
                System.out.println("Continuing with the program...");
                System.out.println("");
                break;
            } else {
                // Handle invalid input.
                System.out.println("Invalid input. Please try again.");
            }
        }
    }
}
```

Eine Wrapper-Klasse für Systems-Manager-SDK-Methoden.

```
public class SSMActions {

    private static SsmAsyncClient ssmAsyncClient;

    private static SsmAsyncClient getAsyncClient() {
        if (ssmAsyncClient == null) {
            SdkAsyncHttpClient httpClient = NettyNioAsyncHttpClient.builder()
                    .maxConcurrency(100)
                    .connectionTimeout(Duration.ofSeconds(60))
                    .readTimeout(Duration.ofSeconds(60))
                    .writeTimeout(Duration.ofSeconds(60))
                    .build();

            ClientOverrideConfiguration overrideConfig =
 ClientOverrideConfiguration.builder()
                    .apiCallTimeout(Duration.ofMinutes(2))
                    .apiCallAttemptTimeout(Duration.ofSeconds(90))
                    .retryPolicy(RetryPolicy.builder()
                            .numRetries(3)
                            .build())
                    .build();

            ssmAsyncClient = SsmAsyncClient.builder()
                    .region(Region.US_WEST_2)
```

```
                    .httpClient(httpClient)
                    .overrideConfiguration(overrideConfig)
                    .build();
        }
        return ssmAsyncClient;
    }

    /**
     * Deletes an AWS SSM document asynchronously.
     *
     * @param documentName The name of the document to delete.
     * <p>
     * This method initiates an asynchronous request to delete an SSM document.
     * If an exception occurs, it handles the error appropriately.
     */
    public void deleteDoc(String documentName) {
        DeleteDocumentRequest documentRequest = DeleteDocumentRequest.builder()
                .name(documentName)
                .build();

        CompletableFuture<Void> future = CompletableFuture.runAsync(() -> {
            getAsyncClient().deleteDocument(documentRequest)
                    .thenAccept(response -> {
                        System.out.println("The SSM document was successfully
deleted.");
                    })
                    .exceptionally(ex -> {
                        throw new CompletionException(ex);
                    }).join();
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
                throw new RuntimeException("SSM error: " + cause.getMessage(),
cause);
            } else {
                throw new RuntimeException("Unexpected error: " +
cause.getMessage(), cause);
            }
        });

        try {
            future.join();
        } catch (CompletionException ex) {
```

```
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
ex.getCause() : ex;
        }
    }

    /**
     * Deletes an AWS SSM Maintenance Window asynchronously.
     *
     * @param winId The ID of the Maintenance Window to delete.
     * <p>
     * This method initiates an asynchronous request to delete an SSM Maintenance
Window.
     * If an exception occurs, it handles the error appropriately.
     */
    public void deleteMaintenanceWindow(String winId) {
        DeleteMaintenanceWindowRequest windowRequest =
DeleteMaintenanceWindowRequest.builder()
                .windowId(winId)
                .build();

        CompletableFuture<Void> future = CompletableFuture.runAsync(() -> {
            getAsyncClient().deleteMaintenanceWindow(windowRequest)
                    .thenAccept(response -> {
                        System.out.println("The maintenance window was
successfully deleted.");
                    })
                    .exceptionally(ex -> {
                        throw new CompletionException(ex);
                    }).join();
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
                throw new RuntimeException("SSM error: " + cause.getMessage(),
cause);
            } else {
                throw new RuntimeException("Unexpected error: " +
cause.getMessage(), cause);
            }
        });

        try {
            future.join();
        } catch (CompletionException ex) {
```

```
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
ex.getCause() : ex;
        }
    }

    /**
     * Resolves an AWS SSM OpsItem asynchronously.
     *
     * @param opsID The ID of the OpsItem to resolve.
     * <p>
     * This method initiates an asynchronous request to resolve an SSM OpsItem.
     * If an exception occurs, it handles the error appropriately.
     */
    public void resolveOpsItem(String opsID) {
        UpdateOpsItemRequest opsItemRequest = UpdateOpsItemRequest.builder()
                .opsItemId(opsID)
                .status(OpsItemStatus.RESOLVED)
                .build();

        CompletableFuture<Void> future = CompletableFuture.runAsync(() -> {
            getAsyncClient().updateOpsItem(opsItemRequest)
                    .thenAccept(response -> {
                        System.out.println("OpsItem resolved successfully.");
                    })
                    .exceptionally(ex -> {
                        throw new CompletionException(ex);
                    }).join();
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
                throw new RuntimeException("SSM error: " + cause.getMessage(),
cause);
            } else {
                throw new RuntimeException("Unexpected error: " +
cause.getMessage(), cause);
            }
        });

        try {
            future.join();
        } catch (CompletionException ex) {
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
ex.getCause() : ex;
```

```
        }
    }

    /**
     * Describes AWS SSM OpsItems asynchronously.
     *
     * @param key The key to filter OpsItems by (e.g., OPS_ITEM_ID).
     *
     * This method initiates an asynchronous request to describe SSM OpsItems.
     * If the request is successful, it prints the title and status of each
OpsItem.
     * If an exception occurs, it handles the error appropriately.
     */
    public void describeOpsItems(String key) {
        OpsItemFilter filter = OpsItemFilter.builder()
                .key(OpsItemFilterKey.OPS_ITEM_ID)
                .values(key)
                .operator(OpsItemFilterOperator.EQUAL)
                .build();

        DescribeOpsItemsRequest itemsRequest = DescribeOpsItemsRequest.builder()
                .maxResults(10)
                .opsItemFilters(filter)
                .build();

        CompletableFuture<Void> future = CompletableFuture.runAsync(() -> {
            getAsyncClient().describeOpsItems(itemsRequest)
                    .thenAccept(itemsResponse -> {
                        List<OpsItemSummary> items =
itemsResponse.opsItemSummaries();
                        for (OpsItemSummary item : items) {
                            System.out.println("The item title is " +
item.title() + " and the status is " + item.status().toString());
                        }
                    })
                    .exceptionally(ex -> {
                        throw new CompletionException(ex);
                    }).join();
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
                throw new RuntimeException("SSM error: " + cause.getMessage(),
cause);
```

```
            } else {
                throw new RuntimeException("Unexpected error: " +
cause.getMessage(), cause);
            }
        });

        try {
            future.join();
        } catch (CompletionException ex) {
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
ex.getCause() : ex;
        }
    }

    /**
     * Updates the AWS SSM OpsItem asynchronously.
     *
     * @param opsItemId The ID of the OpsItem to update.
     * @param title The new title of the OpsItem.
     * @param description The new description of the OpsItem.
     * <p>
     * This method initiates an asynchronous request to update an SSM OpsItem.
     * If the request is successful, it completes without returning a value.
     * If an exception occurs, it handles the error appropriately.
     */
    public void updateOpsItem(String opsItemId, String title, String description)
{
        Map<String, OpsItemDataValue> operationalData = new HashMap<>();
        operationalData.put("key1",
OpsItemDataValue.builder().value("value1").build());
        operationalData.put("key2",
OpsItemDataValue.builder().value("value2").build());

        CompletableFuture<Void> future =
getOpsItem(opsItemId).thenCompose(opsItem -> {
            UpdateOpsItemRequest request = UpdateOpsItemRequest.builder()
                    .opsItemId(opsItemId)
                    .title(title)
                    .operationalData(operationalData)
                    .status(opsItem.statusAsString())
                    .description(description)
                    .build();
```

```
            return getAsyncClient().updateOpsItem(request).thenAccept(response ->
{
                System.out.println(opsItemId + " updated successfully.");
            }).exceptionally(ex -> {
                throw new CompletionException(ex);
            });
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
                throw new RuntimeException("SSM error: " + cause.getMessage(),
cause);
            } else {
                throw new RuntimeException("Unexpected error: " +
cause.getMessage(), cause);
            }
        });

        try {
            future.join();
        } catch (CompletionException ex) {
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
ex.getCause() : ex;
        }
    }


    private static CompletableFuture<OpsItem> getOpsItem(String opsItemId) {
        GetOpsItemRequest request =
GetOpsItemRequest.builder().opsItemId(opsItemId).build();
        return
getAsyncClient().getOpsItem(request).thenApply(GetOpsItemResponse::opsItem);
    }

    /**
     * Creates an SSM OpsItem asynchronously.
     *
     * @param title The title of the OpsItem.
     * @param source The source of the OpsItem.
     * @param category The category of the OpsItem.
     * @param severity The severity of the OpsItem.
     * @return The ID of the created OpsItem.
     * <p>
     * This method initiates an asynchronous request to create an SSM OpsItem.
```

```
    * If the request is successful, it returns the OpsItem ID.
    * If an exception occurs, it handles the error appropriately.
    */
   public String createSSMOpsItem(String title, String source, String category,
String severity) {
       CreateOpsItemRequest opsItemRequest = CreateOpsItemRequest.builder()
               .description("Created by the SSM Java API")
               .title(title)
               .source(source)
               .category(category)
               .severity(severity)
               .build();

       CompletableFuture<CreateOpsItemResponse> future =
getAsyncClient().createOpsItem(opsItemRequest);

       try {
           CreateOpsItemResponse response = future.join();
           return response.opsItemId();
       } catch (CompletionException e) {
           Throwable cause = e.getCause();
           if (cause instanceof SsmException) {
               throw (SsmException) cause;
           } else {
               throw new RuntimeException(cause);
           }
       }
   }

   /**
    * Displays the date and time when the specific command was invoked.
    *
    * @param commandId The ID of the command to describe.
    * <p>
    * This method initiates an asynchronous request to list command invocations
and prints the date and time of each command invocation.
    * If an exception occurs, it handles the error appropriately.
    */
   public void displayCommands(String commandId) {
       ListCommandInvocationsRequest commandInvocationsRequest =
ListCommandInvocationsRequest.builder()
               .commandId(commandId)
               .build();
```

```
        CompletableFuture<ListCommandInvocationsResponse> future =
getAsyncClient().listCommandInvocations(commandInvocationsRequest);
        future.thenAccept(response -> {
            List<CommandInvocation> commandList = response.commandInvocations();
            DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd
HH:mm:ss").withZone(ZoneId.systemDefault());
            for (CommandInvocation invocation : commandList) {
                System.out.println("The time of the command invocation is " +
formatter.format(invocation.requestedDateTime()));
            }
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
                throw (SsmException) cause;
            } else {
                throw new RuntimeException(cause);
            }
        }).join();
    }

    /**
     * Sends a SSM command to a managed node asynchronously.
     *
     * @param documentName The name of the document to use.
     * @param instanceId The ID of the instance to send the command to.
     * @return The command ID.
     * <p>
     * This method initiates asynchronous requests to send a SSM command to a
managed node.
     * It waits until the document is active, sends the command, and checks the
command execution status.
     */
    public String sendSSMCommand(String documentName, String instanceId) throws
InterruptedException, SsmException {
        // Before we use Document to send a command - make sure it is active.
        CompletableFuture<Void> documentActiveFuture =
CompletableFuture.runAsync(() -> {
            boolean isDocumentActive = false;
            DescribeDocumentRequest request = DescribeDocumentRequest.builder()
                    .name(documentName)
                    .build();

            while (!isDocumentActive) {
```

```
                CompletableFuture<DescribeDocumentResponse> response =
getAsyncClient().describeDocument(request);
                String documentStatus =
response.join().document().statusAsString();
                if (documentStatus.equals("Active")) {
                    System.out.println("The SSM document is active and ready to
use.");
                    isDocumentActive = true;
                } else {
                    System.out.println("The SSM document is not active. Status: "
+ documentStatus);
                    try {
                        Thread.sleep(5000);
                    } catch (InterruptedException e) {
                        throw new RuntimeException(e);
                    }
                }
            }
        });

        documentActiveFuture.join();

        // Create the SendCommandRequest.
        SendCommandRequest commandRequest = SendCommandRequest.builder()
                .documentName(documentName)
                .instanceIds(instanceId)
                .build();

        // Send the command.
        CompletableFuture<SendCommandResponse> commandFuture =
getAsyncClient().sendCommand(commandRequest);
        final String[] commandId = {null};

        commandFuture.whenComplete((commandResponse, ex) -> {
            if (commandResponse != null) {
                commandId[0] = commandResponse.command().commandId();
                System.out.println("Command ID: " + commandId[0]);

                // Wait for the command execution to complete.
                GetCommandInvocationRequest invocationRequest =
GetCommandInvocationRequest.builder()
                        .commandId(commandId[0])
                        .instanceId(instanceId)
                        .build();
```

```
                try {
                    System.out.println("Wait 5 secs");
                    TimeUnit.SECONDS.sleep(5);

                    // Retrieve the command execution details.
                    CompletableFuture<GetCommandInvocationResponse>
invocationFuture = getAsyncClient().getCommandInvocation(invocationRequest);
                    invocationFuture.whenComplete((commandInvocationResponse,
invocationEx) -> {
                        if (commandInvocationResponse != null) {
                            // Check the status of the command execution.
                            CommandInvocationStatus status =
commandInvocationResponse.status();
                            if (status == CommandInvocationStatus.SUCCESS) {
                                System.out.println("Command execution
successful");
                            } else {
                                System.out.println("Command execution failed.
Status: " + status);
                            }
                        } else {
                            Throwable invocationCause = (invocationEx instanceof
CompletionException) ? invocationEx.getCause() : invocationEx;
                            throw new CompletionException(invocationCause);
                        }
                    }).join();
                } catch (InterruptedException e) {
                    throw new RuntimeException(e);
                }
            } else {
                Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
                if (cause instanceof SsmException) {
                    throw (SsmException) cause;
                } else {
                    throw new RuntimeException(cause);
                }
            }
        }).join();

        return commandId[0];
    }
```

```
    /**
     * Creates an AWS SSM document asynchronously.
     *
     * @param docName The name of the document to create.
     * <p>
     * This method initiates an asynchronous request to create an SSM document.
     * If the request is successful, it prints the document status.
     * If an exception occurs, it handles the error appropriately.
     */
    public void createSSMDoc(String docName) throws SsmException {
        String jsonData = """
        {
        "schemaVersion": "2.2",
        "description": "Run a simple shell command",
        "mainSteps": [
            {
                "action": "aws:runShellScript",
                "name": "runEchoCommand",
                "inputs": {
                  "runCommand": [
                    "echo 'Hello, world!'"
                  ]
                }
              }
            ]
        }
        """;

        CreateDocumentRequest request = CreateDocumentRequest.builder()
                .content(jsonData)
                .name(docName)
                .documentType(DocumentType.COMMAND)
                .build();

        CompletableFuture<CreateDocumentResponse> future =
getAsyncClient().createDocument(request);
        future.thenAccept(response -> {
            System.out.println("The status of the SSM document is " +
response.documentDescription().status());
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof DocumentAlreadyExistsException) {
                throw new CompletionException(cause);
```

```
            } else if (cause instanceof SsmException) {
                throw new CompletionException(cause);
            } else {
                throw new RuntimeException(cause);
            }
        }).join();
    }

    /**
     * Updates an SSM maintenance window asynchronously.
     *
     * @param id The ID of the maintenance window to update.
     * @param name The new name for the maintenance window.
     * <p>
     * This method initiates an asynchronous request to update an SSM maintenance
window.
     * If the request is successful, it prints a success message.
     * If an exception occurs, it handles the error appropriately.
     */
    public void updateSSMMaintenanceWindow(String id, String name) throws
SsmException {
        UpdateMaintenanceWindowRequest updateRequest =
UpdateMaintenanceWindowRequest.builder()
                .windowId(id)
                .allowUnassociatedTargets(true)
                .duration(24)
                .enabled(true)
                .name(name)
                .schedule("cron(0 0 ? * MON *)")
                .build();

        CompletableFuture<UpdateMaintenanceWindowResponse> future =
getAsyncClient().updateMaintenanceWindow(updateRequest);
        future.whenComplete((response, ex) -> {
            if (response != null) {
                System.out.println("The SSM maintenance window was successfully
updated");
            } else {
                Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
                if (cause instanceof SsmException) {
                    throw new CompletionException(cause);
                } else {
                    throw new RuntimeException(cause);
```

```
                }
            }
        }).join();
    }

    /**
     * Creates an SSM maintenance window asynchronously.
     *
     * @param winName The name of the maintenance window.
     * @return The ID of the created or existing maintenance window.
     * <p>
     * This method initiates an asynchronous request to create an SSM maintenance
window.
     * If the request is successful, it prints the maintenance window ID.
     * If an exception occurs, it handles the error appropriately.
     */
    public String createMaintenanceWindow(String winName) throws SsmException,
DocumentAlreadyExistsException {
        CreateMaintenanceWindowRequest request =
CreateMaintenanceWindowRequest.builder()
                .name(winName)
                .description("This is my maintenance window")
                .allowUnassociatedTargets(true)
                .duration(2)
                .cutoff(1)
                .schedule("cron(0 10 ? * MON-FRI *)")
                .build();

        CompletableFuture<CreateMaintenanceWindowResponse> future =
getAsyncClient().createMaintenanceWindow(request);
        final String[] windowId = {null};
        future.whenComplete((response, ex) -> {
            if (response != null) {
                String maintenanceWindowId = response.windowId();
                System.out.println("The maintenance window id is " +
maintenanceWindowId);
                windowId[0] = maintenanceWindowId;
            } else {
                Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
                if (cause instanceof DocumentAlreadyExistsException) {
                    throw new CompletionException(cause);
                } else if (cause instanceof SsmException) {
                    throw new CompletionException(cause);
```

```
                } else {
                    throw new RuntimeException(cause);
                }
            }
        }).join();

        if (windowId[0] == null) {
            MaintenanceWindowFilter filter = MaintenanceWindowFilter.builder()
                    .key("name")
                    .values(winName)
                    .build();

            DescribeMaintenanceWindowsRequest winRequest =
 DescribeMaintenanceWindowsRequest.builder()
                    .filters(filter)
                    .build();

            CompletableFuture<DescribeMaintenanceWindowsResponse> describeFuture
 = getAsyncClient().describeMaintenanceWindows(winRequest);
            describeFuture.whenComplete((describeResponse, describeEx) -> {
                if (describeResponse != null) {
                    List<MaintenanceWindowIdentity> windows =
 describeResponse.windowIdentities();
                    if (!windows.isEmpty()) {
                        windowId[0] = windows.get(0).windowId();
                        System.out.println("Window ID: " + windowId[0]);
                    } else {
                        System.out.println("Window not found.");
                        windowId[0] = "";
                    }
                } else {
                    Throwable describeCause = (describeEx instanceof
 CompletionException) ? describeEx.getCause() : describeEx;
                    throw new RuntimeException("Error describing maintenance
 windows: " + describeCause.getMessage(), describeCause);
                }
            }).join();
        }

        return windowId[0];
    }
}
```

- Weitere API-Informationen finden Sie in den folgenden Themen der AWS SDK for Java 2.x - API-Referenz.
  - CreateDocument
  - CreateMaintenanceWindow
  - CreateOpsItem
  - DeleteMaintenanceWindow
  - ListCommandInvocations
  - SendCommand
  - UpdateOpsItem

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import {
  Scenario,
  ScenarioAction,
  ScenarioInput,
  ScenarioOutput,
} from "@aws-doc-sdk-examples/lib/scenario/index.js";
import { fileURLToPath } from "node:url";
import {
  CreateDocumentCommand,
  CreateMaintenanceWindowCommand,
  CreateOpsItemCommand,
  DeleteDocumentCommand,
  DeleteMaintenanceWindowCommand,
  DeleteOpsItemCommand,
  DescribeOpsItemsCommand,
  DocumentAlreadyExists,
  OpsItemStatus,
  waitUntilCommandExecuted,
```

```
  CancelCommandCommand,
  paginateListCommandInvocations,
  SendCommandCommand,
  UpdateMaintenanceWindowCommand,
  UpdateOpsItemCommand,
  SSMClient,
} from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * @typedef {{
 *   ssmClient: import('@aws-sdk/client-ssm').SSMClient,
 *   documentName?: string
 *   maintenanceWindow?: string
 *   winId?: int
 *   ec2InstanceId?: string
 *   requestedDateTime?: Date
 *   opsItemId?: string
 *   askToDeleteResources?: boolean
 * }} State
 */

const defaultMaintenanceWindow = "ssm-maintenance-window";
const defaultDocumentName = "ssmdocument";
// The timeout duration is highly dependent on the specific setup and environment
 necessary. This example handles only the most common error cases, and uses a
 much shorter duration than most productions systems would use.
const COMMAND_TIMEOUT_DURATION_SECONDS = 30; // 30 seconds

const pressEnter = new ScenarioInput("continue", "Press Enter to continue", {
  type: "confirm",
});

const greet = new ScenarioOutput(
  "greet",
  `Welcome to the AWS Systems Manager SDK Getting Started scenario.
    This program demonstrates how to interact with Systems Manager using the AWS
 SDK for JavaScript V3.
    Systems Manager is the operations hub for your AWS applications and resources
 and a secure end-to-end management solution.
    The program's primary functions include creating a maintenance window,
 creating a document, sending a command to a document,
    listing documents, listing commands, creating an OpsItem, modifying an
 OpsItem, and deleting Systems Manager resources.
```

```
    Upon completion of the program, all AWS resources are cleaned up.
    Let's get started...`,
  { header: true },
);

const createMaintenanceWindow = new ScenarioOutput(
  "createMaintenanceWindow",
  "Step 1: Create a Systems Manager maintenance window.",
);

const getMaintenanceWindow = new ScenarioInput(
  "maintenanceWindow",
  "Please enter the maintenance window name:",
  { type: "input", default: defaultMaintenanceWindow },
);

export const sdkCreateMaintenanceWindow = new ScenarioAction(
  "sdkCreateMaintenanceWindow",
  async (/** @type {State} */ state) => {
    try {
      const response = await state.ssmClient.send(
        new CreateMaintenanceWindowCommand({
          Name: state.maintenanceWindow,
          Schedule: "cron(0 10 ? * MON-FRI *)", //The schedule of the maintenance
 window in the form of a cron or rate expression.
          Duration: 2, //The duration of the maintenance window in hours.
          Cutoff: 1, //The number of hours before the end of the maintenance
 window that Amazon Web Services Systems Manager stops scheduling new tasks for
 execution.
          AllowUnassociatedTargets: true, //Allow the maintenance window to run
 on managed nodes, even if you haven't registered those nodes as targets.
        }),
      );
      state.winId = response.WindowId;
    } catch (caught) {
      console.error(caught.message);
      console.log(
        `An error occurred while creating the maintenance window. Please fix the
 error and try again. Error message: ${caught.message}`,
      );
      throw caught;
    }
  },
);
```

```
const modifyMaintenanceWindow = new ScenarioOutput(
  "modifyMaintenanceWindow",
  "Modify the maintenance window by changing the schedule.",
);

const sdkModifyMaintenanceWindow = new ScenarioAction(
  "sdkModifyMaintenanceWindow",
  async (/** @type {State} */ state) => {
    try {
      await state.ssmClient.send(
        new UpdateMaintenanceWindowCommand({
          WindowId: state.winId,
          Schedule: "cron(0 0 ? * MON *)",
        }),
      );
    } catch (caught) {
      console.error(caught.message);
      console.log(
        `An error occurred while modifying the maintenance window. Please fix the
 error and try again. Error message: ${caught.message}`,
      );
      throw caught;
    }
  },
);

const createSystemsManagerActions = new ScenarioOutput(
  "createSystemsManagerActions",
  "Create a document that defines the actions that Systems Manager performs on
 your EC2 instance.",
);

const getDocumentName = new ScenarioInput(
  "documentName",
  "Please enter the document: ",
  { type: "input", default: defaultDocumentName },
);

const sdkCreateSSMDoc = new ScenarioAction(
  "sdkCreateSSMDoc",
  async (/** @type {State} */ state) => {
    const contentData = `{
                "schemaVersion": "2.2",
```

```
                "description": "Run a simple shell command",
                "mainSteps": [
                    {
                        "action": "aws:runShellScript",
                        "name": "runEchoCommand",
                        "inputs": {
                          "runCommand": [
                            "echo 'Hello, world!'"
                          ]
                        }
                    }
                ]
            }`;
    try {
      await state.ssmClient.send(
        new CreateDocumentCommand({
          Content: contentData,
          Name: state.documentName,
          DocumentType: "Command",
        }),
      );
    } catch (caught) {
      console.log(`Exception type: (${typeof caught})`);
      if (caught instanceof DocumentAlreadyExists) {
        console.log("Document already exists. Continuing...\n");
      } else {
        console.error(caught.message);
        console.log(
          `An error occurred while creating the document. Please fix the error
 and try again. Error message: ${caught.message}`,
        );
        throw caught;
      }
    }
  },
);

const ec2HelloWorld = new ScenarioOutput(
  "ec2HelloWorld",
  `Now you have the option of running a command on an EC2 instance that echoes
 'Hello, world!'. In order to run this command, you must provide the instance ID
 of a Linux EC2 instance. If you do not already have a running Linux EC2 instance
 in your account, you can create one using the AWS console. For information
```

```
 about creating an EC2 instance, see https://docs.aws.amazon.com/AWSEC2/latest/
UserGuide/ec2-launch-instance-wizard.html.`,
);

const enterIdOrSkipEC2HelloWorld = new ScenarioInput(
  "enterIdOrSkipEC2HelloWorld",
  "Enter your EC2 InstanceId or press enter to skip this step: ",
  { type: "input", default: "" },
);

const sdkEC2HelloWorld = new ScenarioAction(
  "sdkEC2HelloWorld",
  async (/** @type {State} */ state) => {
    try {
      const response = await state.ssmClient.send(
        new SendCommandCommand({
          DocumentName: state.documentName,
          InstanceIds: [state.ec2InstanceId],
          TimeoutSeconds: COMMAND_TIMEOUT_DURATION_SECONDS,
        }),
      );
      state.CommandId = response.Command.CommandId;
    } catch (caught) {
      console.error(caught.message);
      console.log(
        `An error occurred while sending the command. Please fix the error and
 try again. Error message: ${caught.message}`,
      );
      throw caught;
    }
  },
  {
    skipWhen: (/** @type {State} */ state) =>
      state.enterIdOrSkipEC2HelloWorld === "",
  },
);

const sdkGetCommandTime = new ScenarioAction(
  "sdkGetCommandTime",
  async (/** @type {State} */ state) => {
    const listInvocationsPaginated = [];
    console.log(
      "Let's get the time when the specific command was sent to the specific
 managed node.",
```

```
    );

    console.log(
      `First, we'll wait for the command to finish executing. This may take up to
${COMMAND_TIMEOUT_DURATION_SECONDS} seconds.`,
    );
    const commandExecutedResult = waitUntilCommandExecuted(
      { client: state.ssmClient },
      {
        CommandId: state.CommandId,
        InstanceId: state.ec2InstanceId,
      },
    );
    // This is necessary because the TimeoutSeconds of SendCommandCommand is only
for the delivery, not execution.
    try {
      await new Promise((_, reject) =>
        setTimeout(
          reject,
          COMMAND_TIMEOUT_DURATION_SECONDS * 1000,
          new Error("Command Timed Out"),
        ),
      );
    } catch (caught) {
      if (caught.message === "Command Timed Out") {
        commandExecutedResult.state = "TIMED_OUT";
      } else {
        throw caught;
      }
    }

    if (commandExecutedResult.state !== "SUCCESS") {
      console.log(
        `The command with id: ${state.CommandId} did not execute in the allotted
time. Canceling command.`,
      );
      state.ssmClient.send(
        new CancelCommandCommand({
          CommandId: state.CommandId,
        }),
      );
      state.enterIdOrSkipEC2HelloWorld === "";
      return;
    }
```

```
    for await (const page of paginateListCommandInvocations(
      { client: state.ssmClient },
      { CommandId: state.CommandId },
    )) {
      listInvocationsPaginated.push(...page.CommandInvocations);
    }
    /**
     * @type {import('@aws-sdk/client-ssm').CommandInvocation}
     */
    const commandInvocation = listInvocationsPaginated.shift(); // Because the
 call was made with CommandId, there's only one result, so shift it off.
    state.requestedDateTime = commandInvocation.RequestedDateTime;

    console.log(
      `The command invocation happened at: ${state.requestedDateTime}.`,
    );
  },
  {
    skipWhen: (/** @type {State} */ state) =>
      state.enterIdOrSkipEC2HelloWorld === "",
  },
);

const createSSMOpsItem = new ScenarioOutput(
  "createSSMOpsItem",
  `Now we will create a Systems Manager OpsItem. An OpsItem is a feature provided
 by the Systems Manager service. It is a type of operational data item that
 allows you to manage and track various operational issues, events, or tasks
 within your AWS environment.
You can create OpsItems to track and manage operational issues as they arise.
 For example, you could create an OpsItem whenever your application detects a
 critical error or an anomaly in your infrastructure.`,
);

const sdkCreateSSMOpsItem = new ScenarioAction(
  "sdkCreateSSMOpsItem",
  async (/** @type {State} */ state) => {
    try {
      const response = await state.ssmClient.send(
        new CreateOpsItemCommand({
          Description: "Created by the System Manager Javascript API",
          Title: "Disk Space Alert",
          Source: "EC2",
```

```
          Category: "Performance",
          Severity: "2",
        }),
      );
      state.opsItemId = response.OpsItemId;
    } catch (caught) {
      console.error(caught.message);
      console.log(
        `An error occurred while creating the ops item. Please fix the error and
 try again. Error message: ${caught.message}`,
      );
      throw caught;
    }
  },
);

const updateOpsItem = new ScenarioOutput(
  "updateOpsItem",
  (/** @type {State} */ state) =>
    `Now we will update the OpsItem: ${state.opsItemId}`,
);

const sdkUpdateOpsItem = new ScenarioAction(
  "sdkUpdateOpsItem",
  async (/** @type {State} */ state) => {
    try {
      const _response = await state.ssmClient.send(
        new UpdateOpsItemCommand({
          OpsItemId: state.opsItemId,
          Description: `An update to ${state.opsItemId}`,
        }),
      );
    } catch (caught) {
      console.error(caught.message);
      console.log(
        `An error occurred while updating the ops item. Please fix the error and
 try again. Error message: ${caught.message}`,
      );
      throw caught;
    }
  },
);

const getOpsItemStatus = new ScenarioOutput(
```

```
  "getOpsItemStatus",
  (/** @type {State} */ state) =>
    `Now we will get the status of the OpsItem: ${state.opsItemId}`,
);

const sdkOpsItemStatus = new ScenarioAction(
  "sdkGetOpsItemStatus",
  async (/** @type {State} */ state) => {
    try {
      const response = await state.ssmClient.send(
        new DescribeOpsItemsCommand({
          OpsItemId: state.opsItemId,
        }),
      );
      state.opsItemStatus = response.OpsItemStatus;
    } catch (caught) {
      console.error(caught.message);
      console.log(
        `An error occurred while describing the ops item. Please fix the error
 and try again. Error message: ${caught.message}`,
      );
      throw caught;
    }
  },
);

const resolveOpsItem = new ScenarioOutput(
  "resolveOpsItem",
  (/** @type {State} */ state) =>
    `Now we will resolve the OpsItem: ${state.opsItemId}`,
);

const sdkResolveOpsItem = new ScenarioAction(
  "sdkResolveOpsItem",
  async (/** @type {State} */ state) => {
    try {
      const _response = await state.ssmClient.send(
        new UpdateOpsItemCommand({
          OpsItemId: state.opsItemId,
          Status: OpsItemStatus.RESOLVED,
        }),
      );
    } catch (caught) {
      console.error(caught.message);
```

```
      console.log(
        `An error occurred while updating the ops item. Please fix the error and
 try again. Error message: ${caught.message}`,
      );
      throw caught;
    }
  },
);

const askToDeleteResources = new ScenarioInput(
  "askToDeleteResources",
  "Would you like to delete the Systems Manager resources created during this
 example run?",
  { type: "confirm" },
);

const confirmDeleteChoice = new ScenarioOutput(
  "confirmDeleteChoice",
  (/** @type {State} */ state) => {
    if (state.askToDeleteResources) {
      return "You chose to delete the resources.";
    }
    return "The Systems Manager resources will not be deleted. Please delete them
 manually to avoid charges.";
  },
);

export const sdkDeleteResources = new ScenarioAction(
  "sdkDeleteResources",
  async (/** @type {State} */ state) => {
    try {
      await state.ssmClient.send(
        new DeleteOpsItemCommand({
          OpsItemId: state.opsItemId,
        }),
      );
      console.log(`The ops item: ${state.opsItemId} was successfully deleted.`);
    } catch (caught) {
      console.log(
        `There was a problem deleting the ops item: ${state.opsItemId}. Please
 delete it manually. Error: ${caught.message}`,
      );
    }
```

```
    try {
      await state.ssmClient.send(
        new DeleteMaintenanceWindowCommand({
          Name: state.maintenanceWindow,
          WindowId: state.winId,
        }),
      );
      console.log(
        `The maintenance window: ${state.maintenanceWindow} was successfully
 deleted.`,
      );
    } catch (caught) {
      console.log(
        `There was a problem deleting the maintenance window: ${state.opsItemId}.
 Please delete it manually. Error: ${caught.message}`,
      );
    }

    try {
      await state.ssmClient.send(
        new DeleteDocumentCommand({
          Name: state.documentName,
        }),
      );
      console.log(
        `The document: ${state.documentName} was successfully deleted.`,
      );
    } catch (caught) {
      console.log(
        `There was a problem deleting the document: ${state.documentName}. Please
 delete it manually. Error: ${caught.message}`,
      );
    }
  },
  { skipWhen: (/** @type {{}} */ state) => !state.askToDeleteResources },
);

const goodbye = new ScenarioOutput(
  "goodbye",
  "This concludes the Systems Manager Basics scenario for the AWS Javascript SDK
 v3. Thank you!",
);

const myScenario = new Scenario(
```

```
  "SSM Basics",
  [
    greet,
    pressEnter,
    createMaintenanceWindow,
    getMaintenanceWindow,
    sdkCreateMaintenanceWindow,
    modifyMaintenanceWindow,
    pressEnter,
    sdkModifyMaintenanceWindow,
    createSystemsManagerActions,
    getDocumentName,
    sdkCreateSSMDoc,
    ec2HelloWorld,
    enterIdOrSkipEC2HelloWorld,
    sdkEC2HelloWorld,
    sdkGetCommandTime,
    pressEnter,
    createSSMOpsItem,
    pressEnter,
    sdkCreateSSMOpsItem,
    updateOpsItem,
    pressEnter,
    sdkUpdateOpsItem,
    getOpsItemStatus,
    pressEnter,
    sdkOpsItemStatus,
    resolveOpsItem,
    pressEnter,
    sdkResolveOpsItem,
    askToDeleteResources,
    confirmDeleteChoice,
    sdkDeleteResources,
    goodbye,
  ],
  { ssmClient: new SSMClient({}) },
);

/** @type {{ stepHandlerOptions: StepHandlerOptions }} */
export const main = async (stepHandlerOptions) => {
  await myScenario.run(stepHandlerOptions);
};

// Invoke main function if this file was run directly.
```

```
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  const { values } = parseArgs({
    options: {
      yes: {
        type: "boolean",
        short: "y",
      },
    },
  });
  main({ confirmAll: values.yes });
}
```

- Weitere API-Informationen finden Sie in den folgenden Themen der AWS SDK für JavaScript -API-Referenz.
  - CreateDocument
  - CreateMaintenanceWindow
  - CreateOpsItem
  - DeleteMaintenanceWindow
  - ListCommandInvocations
  - SendCommand
  - UpdateOpsItem

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Führen Sie ein interaktives Szenario an einem Prompt aus.

```
class SystemsManagerScenario:
    """Runs an interactive scenario that shows how to get started using Amazon
 Systems Manager."""
```

```
    def __init__(self, document_wrapper, maintenance_window_wrapper,
 ops_item_wrapper):
        """
        :param document_wrapper: An object that wraps Systems Manager document
 functions.
        :param maintenance_window_wrapper: An object that wraps Systems Manager
 maintenance window functions.
        :param ops_item_wrapper: An object that wraps Systems Manager OpsItem
 functions.
        """
        self.document_wrapper = document_wrapper
        self.maintenance_window_wrapper = maintenance_window_wrapper
        self.ops_item_wrapper = ops_item_wrapper

    def run(self):
        """Demonstrates how to use the AWS SDK for Python (Boto3) to get started
 with Systems Manager."""
        try:
            print("-" * 88)
            print(
                """
Welcome to the AWS Systems Manager SDK Getting Started scenario.
This program demonstrates how to interact with Systems Manager using the AWS SDK
 for Python (Boto3).
Systems Manager is the operations hub for your AWS applications and resources and
 a secure end-to-end management
solution. The program's primary functions include creating a maintenance window,
 creating a document, sending a
command to a document, listing documents, listing commands, creating an OpsItem,
 modifying an OpsItem, and deleting
Systems Manager resources. Upon completion of the program, all AWS resources are
 cleaned up.
Let's get started..."""
            )
            q.ask("Please hit Enter")

            print("-" * 88)
            print("Create a Systems Manager maintenance window.")
            maintenance_window_name = q.ask(
                "Please enter the maintenance window name (default is ssm-
maintenance-window):",
            )
            if not maintenance_window_name:
                maintenance_window_name = "ssm-maintenance-window"
```

```
            self.maintenance_window_wrapper.create(
                name=maintenance_window_name,
                schedule="cron(0 10 ? * MON-FRI *)",
                duration=2,
                cutoff=1,
                allow_unassociated_targets=True,
            )

            print("-" * 88)
            print("Modify the maintenance window by changing the schedule")
            q.ask("Please hit Enter")

            self.maintenance_window_wrapper.update(
                name=maintenance_window_name,
                schedule="cron(0 0 ? * MON *)",
                duration=24,
                cutoff=1,
                allow_unassociated_targets=True,
                enabled=True,
            )

            print("-" * 88)
            print(
                "Create a document that defines the actions that Systems Manager
 performs on your EC2 instance."
            )
            document_name = q.ask(
                "Please enter the document name (default is ssmdocument):"
            )

            if not document_name:
                document_name = "ssmdocument"

            self.document_wrapper.create(
                name=document_name,
                content="""
{
    "schemaVersion": "2.2",
    "description": "Run a simple shell command",
    "mainSteps": [
        {
            "action": "aws:runShellScript",
            "name": "runEchoCommand",
```

```
            "inputs": {
              "runCommand": [
                "echo 'Hello, world!'"
              ]
            }
        }
    ]
}
            """,
            )

            self.document_wrapper.wait_until_active()

            print(
                """
Now you have the option of running a command on an EC2 instance that echoes
 'Hello, world!'.
In order to run this command, you must provide the instance ID of a Linux EC2
 instance. If you do
not already have a running Linux EC2 instance in your account, you can create one
 using the AWS console.
For information about creating an EC2 instance, see
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-launch-instance-
wizard.html.
            """
            )

            if q.ask(
                "Would you like to run a command on an EC2 instance? (y/n)",
                q.is_yesno,
            ):
                instance_id = q.ask(
                    "Please enter the instance ID of the EC2 instance:",
 q.non_empty
                )
                command_id = self.document_wrapper.send_command(
                    instance_ids=[instance_id]
                )

                self.document_wrapper.wait_command_executed(
                    command_id=command_id, instance_id=instance_id
                )

                print("-" * 88)
```

```
                print(
                    "Lets get the time when the specific command was sent to the
 specific managed node"
                )
                q.ask("Please hit Enter")


 self.document_wrapper.list_command_invocations(instance_id=instance_id)

            print("-" * 88)
            print("-" * 88)
            print(
                """
Now we will create a  Systems Manager OpsItem.
An OpsItem is a feature provided by the Systems Manager service.
It is a type of operational data item that allows you to manage and track various
 operational issues,
events, or tasks within your AWS environment.

You can create OpsItems to track and manage operational issues as they arise.
For example, you could create an OpsItem whenever your application detects a
 critical error
or an anomaly in your infrastructure.
            """
            )
            q.ask("Please hit Enter")

            self.ops_item_wrapper.create(
                title="Disk Space Alert",
                description="Created by the Systems Manager Python (Boto3) API",
                source="EC2",
                category="Performance",
                severity="2",
            )

            print("-" * 88)
            print("-" * 88)
            print(f"Now we will update  the OpsItem {self.ops_item_wrapper.id}")
            q.ask("Please hit Enter")

            self.ops_item_wrapper.update(
                title="Disk Space Alert",
                description=f"An update to {self.ops_item_wrapper.id}",
            )
```

```
            print(
                f"Now we will get the status of the OpsItem
{self.ops_item_wrapper.id}"
            )
            q.ask("Please hit Enter")

            # It may take a second for the ops item to be available
            counter = 0
            while not self.ops_item_wrapper.describe() and counter < 5:
                counter += 1
                time.sleep(1)

            print(f"Now we will resolve the OpsItem {self.ops_item_wrapper.id}")
            q.ask("Please hit Enter")

            self.ops_item_wrapper.update(status="Resolved")

            print("-" * 88)
            print("-" * 88)
            if q.ask(
                "Would you like to delete the Systems Manager resources? (y/n)",
                q.is_yesno,
            ):
                print("You selected to delete the resources.")
                self.cleanup()
            else:
                print("The Systems Manager resources will not be deleted")

            print("-" * 88)
            print("This concludes the Systems Manager SDK Getting Started
scenario.")
            print("-" * 88)

        except Exception:
            self.cleanup()
            raise

    def cleanup(self):
        self.maintenance_window_wrapper.delete()
        self.ops_item_wrapper.delete()
        self.document_wrapper.delete()
```

```
if __name__ == "__main__":
    try:
        scenario = SystemsManagerScenario(
            DocumentWrapper.from_client(),
            MaintenanceWindowWrapper.from_client(),
            OpsItemWrapper.from_client(),
        )
        scenario.run()
    except Exception:
        logging.exception("Something went wrong with the demo.")
```

Definieren Sie eine Klasse, die Dokument- und Befehlsaktionen umschließt.

```
class DocumentWrapper:
    """Encapsulates AWS Systems Manager Document actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.name = None

    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def create(self, content, name):
        """
        Creates a document.

        :param content: The content of the document.
        :param name: The name of the document.
        """
        try:
            self.ssm_client.create_document(
                Name=name, Content=content, DocumentType="Command"
            )
            self.name = name
        except self.ssm_client.exceptions.DocumentAlreadyExists:
```

```
            print(f"Document {name} already exists.")
            self.name = name
        except ClientError as err:
            logger.error(
                "Couldn't create %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def delete(self):
        """
        Deletes an AWS Systems Manager document.
        """
        if self.name is None:
            return

        try:
            self.ssm_client.delete_document(Name=self.name)
            print(f"Deleted document {self.name}.")
            self.name = None
        except ClientError as err:
            logger.error(
                "Couldn't delete %s. Here's why: %s: %s",
                self.name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def send_command(self, instance_ids):
        """
        Sends a command to one or more instances.

        :param instance_ids: The IDs of the instances to send the command to.
        :return: The ID of the command.
        """
        try:
            response = self.ssm_client.send_command(
                InstanceIds=instance_ids, DocumentName=self.name,
TimeoutSeconds=3600
```

```
            )
            return response["Command"]["CommandId"]
        except ClientError as err:
            logger.error(
                "Couldn't send command to %s. Here's why: %s: %s",
                self.name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def describe(self):
        """
        Describes the document.

        :return: Document status.
        """
        try:
            response = self.ssm_client.describe_document(Name=self.name)
            return response["Document"]["Status"]
        except ClientError as err:
            logger.error(
                "Couldn't get %s. Here's why: %s: %s",
                self.name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def wait_until_active(self, max_attempts=20, delay=5):
        """
        Waits until the document is active.

        :param max_attempts: The maximum number of attempts for checking the
status.
        :param delay: The delay in seconds between each check.
        """
        attempt = 0
        status = ""
        while attempt <= max_attempts:
            status = self.describe()
            if status == "Active":
```

```
                break
            attempt += 1
            time.sleep(delay)

        if status != "Active":
            logger.error("Document is not active.")
        else:
            logger.info("Document is active.")

    def wait_command_executed(self, command_id, instance_id):
        """
        Waits until the command is executed on the instance.

        :param command_id: The ID of the command.
        :param instance_id: The ID of the instance.
        """

        waiter = self.ssm_client.get_waiter("command_executed")
        waiter.wait(CommandId=command_id, InstanceId=instance_id)

    def list_command_invocations(self, instance_id):
        """
        Lists the commands for an instance.

        :param instance_id: The ID of the instance.
        :return: The list of commands.
        """
        try:
            paginator = self.ssm_client.get_paginator("list_command_invocations")
            command_invocations = []
            for page in paginator.paginate(InstanceId=instance_id):
                command_invocations.extend(page["CommandInvocations"])
            num_of_commands = len(command_invocations)
            print(
                f"{num_of_commands} command invocation(s) found for instance
{instance_id}."
            )

            if num_of_commands > 10:
                print("Displaying the first 10 commands:")
                num_of_commands = 10
            date_format = "%A, %d %B %Y %I:%M%p"
            for command in command_invocations[:num_of_commands]:
                print(
```

```
                    f"   The time of command invocation is
 {command['RequestedDateTime'].strftime(date_format)}"
                )
        except ClientError as err:
            logger.error(
                "Couldn't list commands for %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

Definieren Sie eine Klasse, die Ops-Objekt-Aktionen umschließt.

```
class OpsItemWrapper:
    """Encapsulates AWS Systems Manager OpsItem actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.id = None

    @classmethod
    def from_client(cls):
        """
        :return: A OpsItemWrapper instance.
        """
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def create(self, title, source, category, severity, description):
        """
        Create an OpsItem

        :param title: The OpsItem title.
        :param source: The OpsItem source.
```

```
        :param category: The OpsItem category.
        :param severity: The OpsItem severity.
        :param description: The OpsItem description.

        """
        try:
            response = self.ssm_client.create_ops_item(
                Title=title,
                Source=source,
                Category=category,
                Severity=severity,
                Description=description,
            )
            self.id = response["OpsItemId"]
        except self.ssm_client.exceptions.OpsItemLimitExceededException as err:
            logger.error(
                "Couldn't create ops item because you have exceeded your open
OpsItem limit. "
                "Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        except ClientError as err:
            logger.error(
                "Couldn't create ops item %s. Here's why: %s: %s",
                title,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise

    def delete(self):
        """
        Delete the OpsItem.
        """
        if self.id is None:
            return
        try:
            self.ssm_client.delete_ops_item(OpsItemId=self.id)
            print(f"Deleted ops item with id {self.id}")
            self.id = None
        except ClientError as err:
            logger.error(
```

```
                "Couldn't delete ops item %s. Here's why: %s: %s",
                self.id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def describe(self):
        """
        Describe an OpsItem.
        """
        try:
            paginator = self.ssm_client.get_paginator("describe_ops_items")
            ops_items = []
            for page in paginator.paginate(
                OpsItemFilters=[
                    {"Key": "OpsItemId", "Values": [self.id], "Operator":
"Equal"}
                ]
            ):
                ops_items.extend(page["OpsItemSummaries"])

            for item in ops_items:
                print(
                    f"The item title is {item['Title']} and the status is
{item['Status']}"
                )
            return len(ops_items) > 0
        except ClientError as err:
            logger.error(
                "Couldn't describe ops item %s. Here's why: %s: %s",
                self.id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def update(self, title=None, description=None, status=None):
        """
        Update an OpsItem.

        :param title: The new OpsItem title.
```

```
        :param description: The new OpsItem description.
        :param status: The new OpsItem status.
        :return:
        """
        args = dict(OpsItemId=self.id)
        if title is not None:
            args["Title"] = title
        if description is not None:
            args["Description"] = description
        if status is not None:
            args["Status"] = status
        try:
            self.ssm_client.update_ops_item(**args)
        except ClientError as err:
            logger.error(
                "Couldn't update ops item %s. Here's why: %s: %s",
                self.id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

Definieren Sie eine Klasse, die Wartungsfenster-Aktionen umschließt.

```
class MaintenanceWindowWrapper:
    """Encapsulates AWS Systems Manager maintenance window actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.window_id = None
        self.name = None

    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)
```

```
   def create(self, name, schedule, duration, cutoff,
allow_unassociated_targets):
       """
       Create an AWS Systems Manager maintenance window.

       :param name: The name of the maintenance window.
       :param schedule: The schedule of the maintenance window.
       :param duration: The duration of the maintenance window.
       :param cutoff: The cutoff time of the maintenance window.
       :param allow_unassociated_targets: Allow the maintenance window to run on
managed nodes, even
                                          if you haven't registered those nodes
as targets.
       """
       try:
           response = self.ssm_client.create_maintenance_window(
               Name=name,
               Schedule=schedule,
               Duration=duration,
               Cutoff=cutoff,
               AllowUnassociatedTargets=allow_unassociated_targets,
           )
           self.window_id = response["WindowId"]
           self.name = name
           logger.info("Created maintenance window %s.", self.window_id)
       except ParamValidationError as error:
           logger.error(
               "Parameter validation error when trying to create maintenance
window %s. Here's why: %s",
               self.window_id,
               error,
           )
           raise
       except ClientError as err:
           logger.error(
               "Couldn't create maintenance window %s. Here's why: %s: %s",
               name,
               err.response["Error"]["Code"],
               err.response["Error"]["Message"],
           )
           raise
```

```
   def delete(self):
       """
       Delete the associated AWS Systems Manager maintenance window.
       """
       if self.window_id is None:
           return

       try:
           self.ssm_client.delete_maintenance_window(WindowId=self.window_id)
           logger.info("Deleted maintenance window %s.", self.window_id)
           print(f"Deleted maintenance window {self.name}")
           self.window_id = None
       except ClientError as err:
           logger.error(
               "Couldn't delete maintenance window %s. Here's why: %s: %s",
               self.window_id,
               err.response["Error"]["Code"],
               err.response["Error"]["Message"],
           )
           raise


   def update(
       self, name, enabled, schedule, duration, cutoff,
allow_unassociated_targets
   ):
       """
       Update an AWS Systems Manager maintenance window.

       :param name: The name of the maintenance window.
       :param enabled: Whether the maintenance window is enabled to run on
managed nodes.
       :param schedule: The schedule of the maintenance window.
       :param duration: The duration of the maintenance window.
       :param cutoff: The cutoff time of the maintenance window.
       :param allow_unassociated_targets: Allow the maintenance window to run on
managed nodes, even
                                          if you haven't registered those nodes
as targets.
       """
       try:
           self.ssm_client.update_maintenance_window(
               WindowId=self.window_id,
```

```
                Name=name,
                Enabled=enabled,
                Schedule=schedule,
                Duration=duration,
                Cutoff=cutoff,
                AllowUnassociatedTargets=allow_unassociated_targets,
            )
            self.name = name
            logger.info("Updated maintenance window %s.", self.window_id)
        except ParamValidationError as error:
            logger.error(
                "Parameter validation error when trying to update maintenance
window %s. Here's why: %s",
                self.window_id,
                error,
            )
            raise
        except ClientError as err:
            logger.error(
                "Couldn't update maintenance window %s. Here's why: %s: %s",
                self.name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Weitere API-Informationen finden Sie in den folgenden Themen der API-Referenz zum AWS SDK für Python (Boto3).
  - CreateDocument
  - CreateMaintenanceWindow
  - CreateOpsItem
  - DeleteMaintenanceWindow
  - ListCommandInvocations
  - SendCommand
  - UpdateOpsItem

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden dieses Dienstes mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Aktionen für Systems Manager mit AWS SDKs

Die folgenden Codebeispiele zeigen, wie einzelne Systems Manager Manager-Aktionen mit ausgeführt AWS SDKs werden. Jedes Beispiel enthält einen Link zu GitHub, wo Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Die folgenden Beispiele enthalten nur die am häufigsten verwendeten Aktionen. Eine vollständige Liste finden Sie in der AWS Systems Manager -API-Referenz.

Beispiele

- Verwendung von AddTagsToResource mit einer CLI
- Verwendung von CancelCommand mit einer CLI
- Verwendung von CreateActivation mit einer CLI
- Verwendung von CreateAssociation mit einer CLI
- Verwendung von CreateAssociationBatch mit einer CLI
- Verwendung CreateDocument mit einem AWS SDK oder CLI
- Verwendung CreateMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung CreateOpsItem mit einem AWS SDK oder CLI
- Verwendung von CreatePatchBaseline mit einer CLI
- Verwendung von DeleteActivation mit einer CLI
- Verwendung von DeleteAssociation mit einer CLI
- Verwendung DeleteDocument mit einem AWS SDK oder CLI
- Verwendung DeleteMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung DeleteOpsItem mit einem AWS SDK
- Verwendung von DeleteParameter mit einer CLI
- Verwendung von DeletePatchBaseline mit einer CLI
- Verwendung von DeregisterManagedInstance mit einer CLI
- Verwendung von DeregisterPatchBaselineForPatchGroup mit einer CLI
- Verwendung von DeregisterTargetFromMaintenanceWindow mit einer CLI
- Verwendung von DeregisterTaskFromMaintenanceWindow mit einer CLI

- Verwendung von DescribeActivations mit einer CLI
- Verwendung von DescribeAssociation mit einer CLI
- Verwendung von DescribeAssociationExecutionTargets mit einer CLI
- Verwendung von DescribeAssociationExecutions mit einer CLI
- Verwendung von DescribeAutomationExecutions mit einer CLI
- Verwendung von DescribeAutomationStepExecutions mit einer CLI
- Verwendung von DescribeAvailablePatches mit einer CLI
- Verwendung von DescribeDocument mit einer CLI
- Verwendung von DescribeDocumentPermission mit einer CLI
- Verwendung von DescribeEffectiveInstanceAssociations mit einer CLI
- Verwendung von DescribeEffectivePatchesForPatchBaseline mit einer CLI
- Verwendung von DescribeInstanceAssociationsStatus mit einer CLI
- Verwendung von DescribeInstanceInformation mit einer CLI
- Verwendung von DescribeInstancePatchStates mit einer CLI
- Verwendung von DescribeInstancePatchStatesForPatchGroup mit einer CLI
- Verwendung von DescribeInstancePatches mit einer CLI
- Verwendung von DescribeMaintenanceWindowExecutionTaskInvocations mit einer CLI
- Verwendung von DescribeMaintenanceWindowExecutionTasks mit einer CLI
- Verwendung von DescribeMaintenanceWindowExecutions mit einer CLI
- Verwendung von DescribeMaintenanceWindowTargets mit einer CLI
- Verwendung von DescribeMaintenanceWindowTasks mit einer CLI
- Verwendung von DescribeMaintenanceWindows mit einer CLI
- Verwendung DescribeOpsItems mit einem AWS SDK oder CLI
- Verwendung DescribeParameters mit einem AWS SDK oder CLI
- Verwendung von DescribePatchBaselines mit einer CLI
- Verwendung von DescribePatchGroupState mit einer CLI
- Verwendung von DescribePatchGroups mit einer CLI
- Verwendung von GetAutomationExecution mit einer CLI
- Verwendung von GetCommandInvocation mit einer CLI
- Verwendung von GetConnectionStatus mit einer CLI

- Verwendung von GetDefaultPatchBaseline mit einer CLI
- Verwendung von GetDeployablePatchSnapshotForInstance mit einer CLI
- Verwendung von GetDocument mit einer CLI
- Verwendung von GetInventory mit einer CLI
- Verwendung von GetInventorySchema mit einer CLI
- Verwendung von GetMaintenanceWindow mit einer CLI
- Verwendung von GetMaintenanceWindowExecution mit einer CLI
- Verwendung von GetMaintenanceWindowExecutionTask mit einer CLI
- Verwendung GetParameter mit einem AWS SDK oder CLI
- Verwendung von GetParameterHistory mit einer CLI
- Verwendung von GetParameters mit einer CLI
- Verwendung von GetPatchBaseline mit einer CLI
- Verwendung von GetPatchBaselineForPatchGroup mit einer CLI
- Verwendung von ListAssociationVersions mit einer CLI
- Verwendung von ListAssociations mit einer CLI
- Verwendung ListCommandInvocations mit einem AWS SDK oder CLI
- Verwendung von ListCommands mit einer CLI
- Verwendung von ListComplianceItems mit einer CLI
- Verwendung von ListComplianceSummaries mit einer CLI
- Verwendung von ListDocumentVersions mit einer CLI
- Verwendung von ListDocuments mit einer CLI
- Verwendung von ListInventoryEntries mit einer CLI
- Verwendung von ListResourceComplianceSummaries mit einer CLI
- Verwendung von ListTagsForResource mit einer CLI
- Verwendung von ModifyDocumentPermission mit einer CLI
- Verwendung von PutComplianceItems mit einer CLI
- Verwendung von PutInventory mit einer CLI
- Verwendung PutParameter mit einem AWS SDK oder CLI
- Verwendung von RegisterDefaultPatchBaseline mit einer CLI
- Verwendung von RegisterPatchBaselineForPatchGroup mit einer CLI

- Verwendung von RegisterTargetWithMaintenanceWindow mit einer CLI
- Verwendung von RegisterTaskWithMaintenanceWindow mit einer CLI
- Verwendung von RemoveTagsFromResource mit einer CLI
- Verwendung SendCommand mit einem AWS SDK oder CLI
- Verwendung von StartAutomationExecution mit einer CLI
- Verwendung von StartSession mit einer CLI
- Verwendung von StopAutomationExecution mit einer CLI
- Verwendung von UpdateAssociation mit einer CLI
- Verwendung von UpdateAssociationStatus mit einer CLI
- Verwendung von UpdateDocument mit einer CLI
- Verwendung von UpdateDocumentDefaultVersion mit einer CLI
- Verwendung UpdateMaintenanceWindow mit einem AWS SDK oder CLI
- Verwendung von UpdateManagedInstanceRole mit einer CLI
- Verwendung UpdateOpsItem mit einem AWS SDK oder CLI
- Verwendung von UpdatePatchBaseline mit einer CLI

Verwendung von **`AddTagsToResource`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `AddTagsToResource` verwendet wird.

CLI

AWS CLI

Beispiel 1: So fügen Sie einem Wartungsfenster Tags hinzu

Im folgenden Beispiel `add-tags-to-resource` wird dem angegebenen Wartungsfenster ein Tag hinzugefügt.

```
aws ssm add-tags-to-resource \
    --resource-type "MaintenanceWindow" \
    --resource-id "mw-03eb9db428EXAMPLE" \
    --tags "Key=Stack,Value=Production"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: So fügen Sie einem Parameter Tags hinzu

Im folgenden Beispiel `add-tags-to-resource` werden dem angegebenen Parameter zwei Tags hinzugefügt.

```
aws ssm add-tags-to-resource \
    --resource-type "Parameter" \
    --resource-id "My-Parameter" \
    --tags '[{"Key":"Region","Value":"East"},{"Key":"Environment",
 "Value":"Production"}]'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 3: So fügen Sie Tags zu SSM-Dokumenten hinzu

Im folgenden `add-tags-to-resource`-Beispiel wird dem angegebenen Dokument ein Tag hinzugefügt.

```
aws ssm add-tags-to-resource \
    --resource-type "Document" \
    --resource-id "My-Document" \
    --tags "Key=Quarter,Value=Q322"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Markieren von Systems-Manager-Ressourcen im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie AddTagsToResourcein der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird ein Wartungsfenster mit neuen Tags aktualisiert. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe. Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$option1 = @{Key="Stack";Value=@("Production")}
Add-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow" -Tag $option1
```

Beispiel 2: Bei PowerShell Version 2 müssen Sie New-Object verwenden, um jedes Tag zu erstellen. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
$tag1 = New-Object Amazon.SimpleSystemsManagement.Model.Tag
$tag1.Key = "Stack"
$tag1.Value = "Production"

Add-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow" -Tag $tag1
```

- Einzelheiten zur API finden Sie unter AddTagsToResource AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird ein Wartungsfenster mit neuen Tags aktualisiert. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe. Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$option1 = @{Key="Stack";Value=@("Production")}
Add-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow" -Tag $option1
```

Beispiel 2: Bei PowerShell Version 2 müssen Sie New-Object verwenden, um jedes Tag zu erstellen. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
$tag1 = New-Object Amazon.SimpleSystemsManagement.Model.Tag
$tag1.Key = "Stack"
$tag1.Value = "Production"

Add-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow" -Tag $tag1
```

- Einzelheiten zur API finden Sie unter AddTagsToResource AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`CancelCommand`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `CancelCommand` verwendet wird.

CLI

AWS CLI

Beispiel 1: Um einen Befehl für alle Instances abzubrechen

Im folgenden `cancel-command`-Beispiel wird versucht, den angegebenen Befehl abzubrechen, der bereits für alle Instances ausgeführt wird.

```
aws ssm cancel-command \
    --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Beispiel 2: So brechen Sie einen Befehl für bestimmte Instances ab

Im folgenden `cancel-command`-Beispiel wird versucht, einen Befehl nur für die angegebene Instance abzubrechen.

```
aws ssm cancel-command \
    --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE" \
    --instance-ids "i-02573cafcfEXAMPLE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Markieren von Systems-Manager-Parametern im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie CancelCommandin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird versucht, einen Befehl abzubrechen. Wenn der Vorgang erfolgreich ist, erfolgt keine Ausgabe.

```
Stop-SSMCommand -CommandId "9ded293e-e792-4440-8e3e-7b8ec5feaa38"
```

- Einzelheiten zur API finden Sie unter CancelCommand AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird versucht, einen Befehl abzubrechen. Wenn der Vorgang erfolgreich ist, erfolgt keine Ausgabe.

```
Stop-SSMCommand -CommandId "9ded293e-e792-4440-8e3e-7b8ec5feaa38"
```

- Einzelheiten zur API finden Sie unter CancelCommand AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`CreateActivation`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `CreateActivation` verwendet wird.

CLI

AWS CLI

So erstellen Sie eine Aktivierung für eine verwaltete Instance

Das folgende `create-activation`-Beispiel erstellt eine verwaltete Instance-Aktivierung.

```
aws ssm create-activation \
    --default-instance-name "HybridWebServers" \
    --iam-role "HybridWebServersRole" \
    --registration-limit 5
```

Ausgabe:

```
{
    "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",
```

```
    "ActivationCode": "dRmgnYaFv567vEXAMPLE"
}
```

Weitere Informationen finden Sie unter Schritt 4: Aktivierung einer verwalteten Instance für eine Hybridumgebung erstellen im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie CreateActivationin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird eine verwaltete Instance erstellt.

```
New-SSMActivation -DefaultInstanceName "MyWebServers" -IamRole
 "SSMAutomationRole" -RegistrationLimit 10
```

Ausgabe:

```
ActivationCode       ActivationId
--------------       ------------
KWChhOxBTiwDcKE9BlKC 08e51e79-1e36-446c-8e63-9458569c1363
```

- Einzelheiten zur API finden Sie unter CreateActivation AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird eine verwaltete Instance erstellt.

```
New-SSMActivation -DefaultInstanceName "MyWebServers" -IamRole
 "SSMAutomationRole" -RegistrationLimit 10
```

Ausgabe:

```
ActivationCode       ActivationId
--------------       ------------
KWChhOxBTiwDcKE9BlKC 08e51e79-1e36-446c-8e63-9458569c1363
```

- Einzelheiten zur API finden Sie unter CreateActivation AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`CreateAssociation`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `CreateAssociation` verwendet wird.

CLI

AWS CLI

Beispiel 1: Um ein Dokument mithilfe einer Instanz zuzuordnen IDs

In diesem Beispiel wird mithilfe von instance ein Konfigurationsdokument einer Instanz zugeordnet IDs.

```
aws ssm create-association \
    --instance-id "i-0cb2b964d3e14fd9f" \
    --name "AWS-UpdateSSMAgent"
```

Ausgabe:

```
{
    "AssociationDescription": {
        "Status": {
            "Date": 1487875500.33,
            "Message": "Associated with AWS-UpdateSSMAgent",
            "Name": "Associated"
        },
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-0cb2b964d3e14fd9f",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1487875500.33,
        "Date": 1487875500.33,
        "Targets": [
            {
                "Values": [
```

```
                    "i-0cb2b964d3e14fd9f"
                ],
                "Key": "InstanceIds"
            }
        ]
    }
}
```

Weitere Informationen finden Sie CreateAssociationin der AWS Systems Manager API-Referenz.

Beispiel 2: So verknüpfen Sie ein Dokument mithilfe von Zielen

In diesem Beispiel wird mithilfe von Zielen ein Konfigurationsdokument einer Instance zugeordnet.

```
aws ssm create-association \
    --name "AWS-UpdateSSMAgent" \
    --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f"
```

Ausgabe:

```
{
    "AssociationDescription": {
        "Status": {
            "Date": 1487875500.33,
            "Message": "Associated with AWS-UpdateSSMAgent",
            "Name": "Associated"
        },
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-0cb2b964d3e14fd9f",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1487875500.33,
        "Date": 1487875500.33,
        "Targets": [
            {
                "Values": [
                    "i-0cb2b964d3e14fd9f"
```

```
                ],
                "Key": "InstanceIds"
            }
        ]
    }
}
```

Weitere Informationen finden Sie CreateAssociationin der AWS Systems Manager API-Referenz.

Beispiel 3: Um eine Zuordnung zu erstellen, die nur einmal ausgeführt wird

In diesem Beispiel wird eine neue Zuordnung erstellt, die nur einmal am angegebenen Datum und zu der angegebenen Uhrzeit ausgeführt wird. Verknüpfungen, die mit einem Datum in der Vergangenheit oder Gegenwart erstellt wurden (zum Zeitpunkt der Verarbeitung liegt das Datum in der Vergangenheit), werden sofort ausgeführt.

```
aws ssm create-association \
    --name "AWS-UpdateSSMAgent" \
    --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f" \
    --schedule-expression "at(2020-05-14T15:55:00)"  \
    --apply-only-at-cron-interval
```

Ausgabe:

```
{
    "AssociationDescription": {
        "Status": {
            "Date": 1487875500.33,
            "Message": "Associated with AWS-UpdateSSMAgent",
            "Name": "Associated"
        },
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-0cb2b964d3e14fd9f",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "AssociationId": "b7c3266e-a544-44db-877e-b20d3a108189",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1487875500.33,
        "Date": 1487875500.33,
```

```
        "Targets": [
            {
                "Values": [
                    "i-0cb2b964d3e14fd9f"
                ],
                "Key": "InstanceIds"
            }
        ]
    }
}
```

Weitere Informationen finden Sie CreateAssociationin der AWS Systems Manager API-Referenz oder Referenz: Cron- und Rate-Ausdrücke für Systems Manager im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie CreateAssociationin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird mithilfe von instance ein Konfigurationsdokument mit einer Instanz verknüpft IDs.

```
New-SSMAssociation -InstanceId "i-0cb2b964d3e14fd9f" -Name "AWS-UpdateSSMAgent"
```

Ausgabe:

```
Name                  : AWS-UpdateSSMAgent
InstanceId            : i-0000293ffd8c57862
Date                  : 2/23/2017 6:55:22 PM
Status.Name           : Associated
Status.Date           : 2/20/2015 8:31:11 AM
Status.Message        : Associated with AWS-UpdateSSMAgent
Status.AdditionalInfo :
```

Beispiel 2: In diesem Beispiel wird mithilfe von Zielen ein Konfigurationsdokument einer Instance zugeordnet.

```
$target = @{Key="instanceids";Values=@("i-0cb2b964d3e14fd9f")}
New-SSMAssociation -Name "AWS-UpdateSSMAgent" -Target $target
```

Ausgabe:

```
Name                  : AWS-UpdateSSMAgent
InstanceId            :
Date                  : 3/1/2017 6:22:21 PM
Status.Name           :
Status.Date           :
Status.Message        :
Status.AdditionalInfo :
```

Beispiel 3: In diesem Beispiel wird mithilfe von Zielen und Parametern ein Konfigurationsdokument einer Instance zugeordnet.

```
$target = @{Key="instanceids";Values=@("i-0cb2b964d3e14fd9f")}
$params = @{
  "action"="configure"
  "mode"="ec2"
  "optionalConfigurationSource"="ssm"
  "optionalConfigurationLocation"=""
  "optionalRestart"="yes"
}
New-SSMAssociation -Name "Configure-CloudWatch" -AssociationName
 "CWConfiguration" -Target $target -Parameter $params
```

Ausgabe:

```
Name                  : Configure-CloudWatch
InstanceId            :
Date                  : 5/17/2018 3:17:44 PM
Status.Name           :
Status.Date           :
Status.Message        :
Status.AdditionalInfo :
```

Beispiel 4: In diesem Beispiel wird eine Zuordnung mit allen Instances in der Region erstellt, mit**`AWS-GatherSoftwareInventory`**. Außerdem werden benutzerdefinierte Dateien und Registrierungsverzeichnisse in den zu erfassenden Parametern bereitgestellt

```
$params =
 [Collections.Generic.Dictionary[String,Collections.Generic.List[String]]]::new()
$params["windowsRegistry"] ='[{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Amazon
\MachineImage","Recursive":false,"ValueNames":["AMIName"]}]'
```

```
$params["files"] = '[{"Path":"C:\Program Files","Pattern":
["*.exe"],"Recursive":true}, {"Path":"C:\ProgramData","Pattern":
["*.log"],"Recursive":true}]'
New-SSMAssociation -AssociationName new-in-mum -Name AWS-GatherSoftwareInventory
 -Target @{Key="instanceids";Values="*"} -Parameter $params -region ap-south-1 -
ScheduleExpression "rate(720 minutes)"
```

Ausgabe:

```
Name                  : AWS-GatherSoftwareInventory
InstanceId            :
Date                  : 6/9/2019 8:57:56 AM
Status.Name           :
Status.Date           :
Status.Message        :
Status.AdditionalInfo :
```

- Einzelheiten zur API finden Sie unter CreateAssociation AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird mithilfe von instance ein Konfigurationsdokument mit einer Instanz verknüpft. IDs

```
New-SSMAssociation -InstanceId "i-0cb2b964d3e14fd9f" -Name "AWS-UpdateSSMAgent"
```

Ausgabe:

```
Name                  : AWS-UpdateSSMAgent
InstanceId            : i-0000293ffd8c57862
Date                  : 2/23/2017 6:55:22 PM
Status.Name           : Associated
Status.Date           : 2/20/2015 8:31:11 AM
Status.Message        : Associated with AWS-UpdateSSMAgent
Status.AdditionalInfo :
```

Beispiel 2: In diesem Beispiel wird mithilfe von Zielen ein Konfigurationsdokument einer Instance zugeordnet.

```
$target = @{Key="instanceids";Values=@("i-0cb2b964d3e14fd9f")}
New-SSMAssociation -Name "AWS-UpdateSSMAgent" -Target $target
```

Ausgabe:

```
Name                  : AWS-UpdateSSMAgent
InstanceId            :
Date                  : 3/1/2017 6:22:21 PM
Status.Name           :
Status.Date           :
Status.Message        :
Status.AdditionalInfo :
```

Beispiel 3: In diesem Beispiel wird mithilfe von Zielen und Parametern ein Konfigurationsdokument einer Instance zugeordnet.

```
$target = @{Key="instanceids";Values=@("i-0cb2b964d3e14fd9f")}
$params = @{
  "action"="configure"
  "mode"="ec2"
  "optionalConfigurationSource"="ssm"
  "optionalConfigurationLocation"=""
  "optionalRestart"="yes"
}
New-SSMAssociation -Name "Configure-CloudWatch" -AssociationName
 "CWConfiguration" -Target $target -Parameter $params
```

Ausgabe:

```
Name                  : Configure-CloudWatch
InstanceId            :
Date                  : 5/17/2018 3:17:44 PM
Status.Name           :
Status.Date           :
Status.Message        :
Status.AdditionalInfo :
```

Beispiel 4: In diesem Beispiel wird eine Zuordnung mit allen Instances in der Region erstellt, mit**`AWS-GatherSoftwareInventory`**. Außerdem werden benutzerdefinierte Dateien und Registrierungsverzeichnisse in den zu erfassenden Parametern bereitgestellt

```
$params =
 [Collections.Generic.Dictionary[String,Collections.Generic.List[String]]]::new()
$params["windowsRegistry"] ='[{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Amazon
\MachineImage","Recursive":false,"ValueNames":["AMIName"]}]'
```

```
$params["files"] = '[{"Path":"C:\Program Files","Pattern":
["*.exe"],"Recursive":true}, {"Path":"C:\ProgramData","Pattern":
["*.log"],"Recursive":true}]'
New-SSMAssociation -AssociationName new-in-mum -Name AWS-GatherSoftwareInventory
 -Target @{Key="instanceids";Values="*"} -Parameter $params -region ap-south-1 -
ScheduleExpression "rate(720 minutes)"
```

Ausgabe:

```
Name                 : AWS-GatherSoftwareInventory
InstanceId           :
Date                 : 6/9/2019 8:57:56 AM
Status.Name          :
Status.Date          :
Status.Message       :
Status.AdditionalInfo :
```

- Einzelheiten zur API finden Sie unter CreateAssociation AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`CreateAssociationBatch`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `CreateAssociationBatch` verwendet wird.

CLI

AWS CLI

So erstellen Sie mehrere Zuordnungen

In diesem Beispiel wird ein Konfigurationsdokument mehreren Instances zugeordnet. Die Ausgabe gibt gegebenenfalls eine Liste mit erfolgreichen und fehlgeschlagenen Vorgängen zurück.

Befehl:

```
aws ssm create-association-batch --entries "Name=AWS-
UpdateSSMAgent,InstanceId=i-1234567890abcdef0" "Name=AWS-
UpdateSSMAgent,InstanceId=i-9876543210abcdef0"
```

Ausgabe:

```
{
  "Successful": [
      {
          "Name": "AWS-UpdateSSMAgent",
          "InstanceId": "i-1234567890abcdef0",
          "AssociationVersion": "1",
          "Date": 1550504725.007,
          "LastUpdateAssociationDate": 1550504725.007,
          "Status": {
              "Date": 1550504725.007,
              "Name": "Associated",
              "Message": "Associated with AWS-UpdateSSMAgent"
          },
          "Overview": {
              "Status": "Pending",
              "DetailedStatus": "Creating"
          },
          "DocumentVersion": "$DEFAULT",
          "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
          "Targets": [
              {
                  "Key": "InstanceIds",
                  "Values": [
                      "i-1234567890abcdef0"
                  ]
              }
          ]
      },
      {
          "Name": "AWS-UpdateSSMAgent",
          "InstanceId": "i-9876543210abcdef0",
          "AssociationVersion": "1",
          "Date": 1550504725.057,
          "LastUpdateAssociationDate": 1550504725.057,
          "Status": {
              "Date": 1550504725.057,
              "Name": "Associated",
```

```
            "Message": "Associated with AWS-UpdateSSMAgent"
        },
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "DocumentVersion": "$DEFAULT",
        "AssociationId": "9c9f7f20-5154-4fed-a83e-0123456789ab",
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "i-9876543210abcdef0"
                ]
            }
        ]
    }
  ],
  "Failed": []
}
```

- Einzelheiten zur API finden Sie CreateAssociationBatchin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird ein Konfigurationsdokument mehreren Instances zugeordnet. Die Ausgabe gibt gegebenenfalls eine Liste mit erfolgreichen und fehlgeschlagenen Vorgängen zurück.

```
$option1 = @{InstanceId="i-0cb2b964d3e14fd9f";Name=@("AWS-UpdateSSMAgent")}
$option2 = @{InstanceId="i-0000293ffd8c57862";Name=@("AWS-UpdateSSMAgent")}
New-SSMAssociationFromBatch -Entry $option1,$option2
```

Ausgabe:

```
Failed  Successful
------  ----------
{}      {Amazon.SimpleSystemsManagement.Model.FailedCreateAssociation,
 Amazon.SimpleSystemsManagement.Model.FailedCreateAsso...
```

Beispiel 2: In diesem Beispiel werden alle Details eines erfolgreichen Vorgangs angezeigt.

```
$option1 = @{InstanceId="i-0cb2b964d3e14fd9f";Name=@("AWS-UpdateSSMAgent")}
$option2 = @{InstanceId="i-0000293ffd8c57862";Name=@("AWS-UpdateSSMAgent")}
(New-SSMAssociationFromBatch -Entry $option1,$option2).Successful
```

- Einzelheiten zur API finden Sie unter CreateAssociationBatch AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird ein Konfigurationsdokument mehreren Instances zugeordnet. Die Ausgabe gibt gegebenenfalls eine Liste mit erfolgreichen und fehlgeschlagenen Vorgängen zurück.

```
$option1 = @{InstanceId="i-0cb2b964d3e14fd9f";Name=@("AWS-UpdateSSMAgent")}
$option2 = @{InstanceId="i-0000293ffd8c57862";Name=@("AWS-UpdateSSMAgent")}
New-SSMAssociationFromBatch -Entry $option1,$option2
```

Ausgabe:

```
Failed  Successful
------  ----------
{}      {Amazon.SimpleSystemsManagement.Model.FailedCreateAssociation,
 Amazon.SimpleSystemsManagement.Model.FailedCreateAsso...
```

Beispiel 2: In diesem Beispiel werden alle Details eines erfolgreichen Vorgangs angezeigt.

```
$option1 = @{InstanceId="i-0cb2b964d3e14fd9f";Name=@("AWS-UpdateSSMAgent")}
$option2 = @{InstanceId="i-0000293ffd8c57862";Name=@("AWS-UpdateSSMAgent")}
(New-SSMAssociationFromBatch -Entry $option1,$option2).Successful
```

- Einzelheiten zur API finden Sie unter CreateAssociationBatch AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CreateDocument`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `CreateDocument` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Kennenlernen der Grundlagen

CLI

AWS CLI

So erstellen Sie ein Dokument

Das folgende `create-document`-Beispiel erstellt ein Systems-Manager-Dokument.

```
aws ssm create-document \
    --content file://exampleDocument.yml \
    --name "Example" \
    --document-type "Automation" \
    --document-format YAML
```

Ausgabe:

```
{
    "DocumentDescription": {
        "Hash":
 "fc2410281f40779e694a8b95975d0f9f316da8a153daa94e3d9921102EXAMPLE",
        "HashType": "Sha256",
        "Name": "Example",
        "Owner": "29884EXAMPLE",
        "CreatedDate": 1583256349.452,
        "Status": "Creating",
        "DocumentVersion": "1",
        "Description": "Document Example",
        "Parameters": [
            {
                "Name": "AutomationAssumeRole",
                "Type": "String",
                "Description": "(Required) The ARN of the role that allows
 Automation to perform the actions on your behalf. If no role is specified,
 Systems Manager Automation uses your IAM permissions to execute this document.",
```

```
                "DefaultValue": ""
            },
            {
                "Name": "InstanceId",
                "Type": "String",
                "Description": "(Required) The ID of the Amazon EC2 instance.",
                "DefaultValue": ""
            }
        ],
        "PlatformTypes": [
            "Windows",
            "Linux"
        ],
        "DocumentType": "Automation",
        "SchemaVersion": "0.3",
        "LatestVersion": "1",
        "DefaultVersion": "1",
        "DocumentFormat": "YAML",
        "Tags": []
    }
}
```

Weitere Informationen finden Sie unter Erstellen von Systems-Manager-Dokumenten im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie CreateDocumentin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    /**
     * Creates an AWS SSM document asynchronously.
     *
     * @param docName The name of the document to create.
     * <p>
```

```
     * This method initiates an asynchronous request to create an SSM document.
     * If the request is successful, it prints the document status.
     * If an exception occurs, it handles the error appropriately.
     */
    public void createSSMDoc(String docName) throws SsmException {
        String jsonData = """
        {
        "schemaVersion": "2.2",
        "description": "Run a simple shell command",
        "mainSteps": [
            {
                "action": "aws:runShellScript",
                "name": "runEchoCommand",
                "inputs": {
                  "runCommand": [
                    "echo 'Hello, world!'"
                  ]
                }
              }
            ]
        }
        """;

        CreateDocumentRequest request = CreateDocumentRequest.builder()
                .content(jsonData)
                .name(docName)
                .documentType(DocumentType.COMMAND)
                .build();

        CompletableFuture<CreateDocumentResponse> future =
getAsyncClient().createDocument(request);
        future.thenAccept(response -> {
            System.out.println("The status of the SSM document is " +
response.documentDescription().status());
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof DocumentAlreadyExistsException) {
                throw new CompletionException(cause);
            } else if (cause instanceof SsmException) {
                throw new CompletionException(cause);
            } else {
                throw new RuntimeException(cause);
            }
```

```
        }).join();
    }
```

- Einzelheiten zur API finden Sie CreateDocumentin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import { CreateDocumentCommand, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * Create an SSM document.
 * @param {{ content: string, name: string, documentType?: DocumentType }}
 */
export const main = async ({ content, name, documentType }) => {
  const client = new SSMClient({});
  try {
    const { documentDescription } = await client.send(
      new CreateDocumentCommand({
        Content: content, // The content for the new SSM document. The content
 must not exceed 64KB.
        Name: name,
        DocumentType: documentType, // Document format type can be JSON, YAML, or
 TEXT. The default format is JSON.
      }),
    );
    console.log("Document created successfully.");
    return { DocumentDescription: documentDescription };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "DocumentAlreadyExists") {
      console.warn(`${caught.message}. Did you provide a new document name?`);
    } else {
      throw caught;
```

```
    }
  }
};
```

- Einzelheiten zur API finden Sie CreateDocumentin der AWS SDK für JavaScript API-Referenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird ein Dokument in Ihrem Konto erstellt. Das Dokument muss im JSON-Format vorliegen. Weitere Informationen zum Schreiben eines Konfigurationsdokuments finden Sie unter Konfigurationsdokument in der SSM-API-Referenz.

```
New-SSMDocument -Content (Get-Content -Raw "c:\temp\RunShellScript.json") -Name
 "RunShellScript" -DocumentType "Command"
```

Ausgabe:

```
CreatedDate     : 3/1/2017 1:21:33 AM
DefaultVersion  : 1
Description     : Run an updated script
DocumentType    : Command
DocumentVersion : 1
Hash            :
 1d5ce820e999ff051eb4841ed887593daf77120fd76cae0d18a53cc42e4e22c1
HashType        : Sha256
LatestVersion   : 1
Name            : RunShellScript
Owner           : 809632081692
Parameters      : {commands}
PlatformTypes   : {Linux}
SchemaVersion   : 2.0
Sha1            :
Status          : Creating
```

- Einzelheiten zur API finden Sie unter CreateDocument AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird ein Dokument in Ihrem Konto erstellt. Das Dokument muss im JSON-Format vorliegen. Weitere Informationen zum Schreiben eines Konfigurationsdokuments finden Sie unter Konfigurationsdokument in der SSM-API-Referenz.

```
New-SSMDocument -Content (Get-Content -Raw "c:\temp\RunShellScript.json") -Name
 "RunShellScript" -DocumentType "Command"
```

Ausgabe:

```
CreatedDate     : 3/1/2017 1:21:33 AM
DefaultVersion  : 1
Description     : Run an updated script
DocumentType    : Command
DocumentVersion : 1
Hash            :
 1d5ce820e999ff051eb4841ed887593daf77120fd76cae0d18a53cc42e4e22c1
HashType        : Sha256
LatestVersion   : 1
Name            : RunShellScript
Owner           : 809632081692
Parameters      : {commands}
PlatformTypes   : {Linux}
SchemaVersion   : 2.0
Sha1            :
Status          : Creating
```

- Einzelheiten zur API finden Sie unter CreateDocument AWS -Tools für PowerShellCmdlet-Referenz (V5).

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class DocumentWrapper:
    """Encapsulates AWS Systems Manager Document actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.name = None

    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def create(self, content, name):
        """
        Creates a document.

        :param content: The content of the document.
        :param name: The name of the document.
        """
        try:
            self.ssm_client.create_document(
                Name=name, Content=content, DocumentType="Command"
            )
            self.name = name
        except self.ssm_client.exceptions.DocumentAlreadyExists:
            print(f"Document {name} already exists.")
            self.name = name
        except ClientError as err:
            logger.error(
                "Couldn't create %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Einzelheiten zur API finden Sie CreateDocumentin AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CreateMaintenanceWindow`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `CreateMaintenanceWindow` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Kennenlernen der Grundlagen

CLI

AWS CLI

Beispiel 1: So erstellen Sie ein Wartungsfenster

Im folgenden `create-maintenance-window`-Beispiel wird ein neues Wartungsfenster erstellt, das alle fünf Minuten für bis zu zwei Stunden (je nach Bedarf) alle fünf Minuten erstellt, verhindert, dass neue Aufgaben innerhalb einer Stunde nach Ende der Ausführung des Wartungsfensters gestartet werden, nicht zugeordnete Ziele (Instances, die Sie nicht für das Wartungsfenster registriert haben) zulässt und durch die Verwendung benutzerdefinierter Tags darauf hinweist, dass der Ersteller beabsichtigt, es in einem Tutorial zu verwenden.

```
aws ssm create-maintenance-window \
    --name "My-Tutorial-Maintenance-Window" \
    --schedule "rate(5 minutes)" \
    --duration 2 --cutoff 1 \
    --allow-unassociated-targets \
    --tags "Key=Purpose,Value=Tutorial"
```

Ausgabe:

```
{
    "WindowId": "mw-0c50858d01EXAMPLE"
```

```
}
```

Beispiel 2: Erstellen eines Wartungsfensters, das nur einmal ausgeführt wird

Im folgenden `create-maintenance-window`-Beispiel wird ein neues Wartungsfenster erstellt, das nur einmal am angegebenen Datum und zur angegebenen Uhrzeit ausgeführt wird.

```
aws ssm create-maintenance-window \
    --name My-One-Time-Maintenance-Window \
    --schedule "at(2020-05-14T15:55:00)" \
    --duration 5 \
    --cutoff 2 \
    --allow-unassociated-targets \
    --tags "Key=Environment,Value=Production"
```

Ausgabe:

```
{
    "WindowId": "mw-01234567890abcdef"
}
```

Weitere Informationen finden Sie unter Wartungsfenster im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie CreateMaintenanceWindowin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    /**
     * Creates an SSM maintenance window asynchronously.
     *
     * @param winName The name of the maintenance window.
```

```
    * @return The ID of the created or existing maintenance window.
    * <p>
    * This method initiates an asynchronous request to create an SSM maintenance
window.
    * If the request is successful, it prints the maintenance window ID.
    * If an exception occurs, it handles the error appropriately.
    */
   public String createMaintenanceWindow(String winName) throws SsmException,
DocumentAlreadyExistsException {
       CreateMaintenanceWindowRequest request =
CreateMaintenanceWindowRequest.builder()
               .name(winName)
               .description("This is my maintenance window")
               .allowUnassociatedTargets(true)
               .duration(2)
               .cutoff(1)
               .schedule("cron(0 10 ? * MON-FRI *)")
               .build();

       CompletableFuture<CreateMaintenanceWindowResponse> future =
getAsyncClient().createMaintenanceWindow(request);
       final String[] windowId = {null};
       future.whenComplete((response, ex) -> {
           if (response != null) {
               String maintenanceWindowId = response.windowId();
               System.out.println("The maintenance window id is " +
maintenanceWindowId);
               windowId[0] = maintenanceWindowId;
           } else {
               Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
               if (cause instanceof DocumentAlreadyExistsException) {
                   throw new CompletionException(cause);
               } else if (cause instanceof SsmException) {
                   throw new CompletionException(cause);
               } else {
                   throw new RuntimeException(cause);
               }
           }
       }).join();

       if (windowId[0] == null) {
           MaintenanceWindowFilter filter = MaintenanceWindowFilter.builder()
                   .key("name")
```

```
                    .values(winName)
                    .build();

            DescribeMaintenanceWindowsRequest winRequest =
DescribeMaintenanceWindowsRequest.builder()
                    .filters(filter)
                    .build();

            CompletableFuture<DescribeMaintenanceWindowsResponse> describeFuture
= getAsyncClient().describeMaintenanceWindows(winRequest);
            describeFuture.whenComplete((describeResponse, describeEx) -> {
                if (describeResponse != null) {
                    List<MaintenanceWindowIdentity> windows =
describeResponse.windowIdentities();
                    if (!windows.isEmpty()) {
                        windowId[0] = windows.get(0).windowId();
                        System.out.println("Window ID: " + windowId[0]);
                    } else {
                        System.out.println("Window not found.");
                        windowId[0] = "";
                    }
                } else {
                    Throwable describeCause = (describeEx instanceof
CompletionException) ? describeEx.getCause() : describeEx;
                    throw new RuntimeException("Error describing maintenance
windows: " + describeCause.getMessage(), describeCause);
                }
            }).join();
        }

        return windowId[0];
    }
```

- Einzelheiten zur API finden Sie CreateMaintenanceWindowin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import { CreateMaintenanceWindowCommand, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * Create an SSM maintenance window.
 * @param {{ name: string, allowUnassociatedTargets: boolean, duration: number,
 cutoff: number, schedule: string, description?: string }}
 */
export const main = async ({
  name,
  allowUnassociatedTargets, // Allow the maintenance window to run on managed
 nodes, even if you haven't registered those nodes as targets.
  duration, // The duration of the maintenance window in hours.
  cutoff, // The number of hours before the end of the maintenance window that
 Amazon Web Services Systems Manager stops scheduling new tasks for execution.
  schedule, // The schedule of the maintenance window in the form of a cron or
 rate expression.
  description = undefined,
}) => {
  const client = new SSMClient({});

  try {
    const { windowId } = await client.send(
      new CreateMaintenanceWindowCommand({
        Name: name,
        Description: description,
        AllowUnassociatedTargets: allowUnassociatedTargets, // Allow the
 maintenance window to run on managed nodes, even if you haven't registered those
 nodes as targets.
        Duration: duration, // The duration of the maintenance window in hours.
```

```
        Cutoff: cutoff, // The number of hours before the end of the maintenance
 window that Amazon Web Services Systems Manager stops scheduling new tasks for
 execution.
        Schedule: schedule, // The schedule of the maintenance window in the form
 of a cron or rate expression.
      }),
    );
    console.log(`Maintenance window created with Id: ${windowId}`);
    return { WindowId: windowId };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "MissingParameter") {
      console.warn(`${caught.message}. Did you provide these values?`);
    } else {
      throw caught;
    }
  }
};
```

- Einzelheiten zur API finden Sie CreateMaintenanceWindowin der AWS SDK für JavaScript API-Referenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird ein neues Wartungsfenster mit dem angegebenen Namen erstellt, das an jedem Dienstag um 16 Uhr für 4 Stunden läuft, mit einer Frist von 1 Stunde, und das Ziele ohne Zuordnung zulässt.

```
New-SSMMaintenanceWindow -Name "MyMaintenanceWindow" -Duration 4 -Cutoff 1 -
AllowUnassociatedTarget $true -Schedule "cron(0 16 ? * TUE *)"
```

Ausgabe:

```
mw-03eb53e1ea7383998
```

- Einzelheiten zur API finden Sie unter CreateMaintenanceWindow AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird ein neues Wartungsfenster mit dem angegebenen Namen erstellt, das an jedem Dienstag um 16 Uhr für 4 Stunden läuft, mit einer Frist von 1 Stunde, und das Ziele ohne Zuordnung zulässt.

```
New-SSMMaintenanceWindow -Name "MyMaintenanceWindow" -Duration 4 -Cutoff 1 -
AllowUnassociatedTarget $true -Schedule "cron(0 16 ? * TUE *)"
```

Ausgabe:

```
mw-03eb53e1ea7383998
```

- Einzelheiten zur API finden Sie unter CreateMaintenanceWindow AWS -Tools für PowerShellCmdlet-Referenz (V5).

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class MaintenanceWindowWrapper:
    """Encapsulates AWS Systems Manager maintenance window actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.window_id = None
        self.name = None

    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
```

```
        return cls(ssm_client)


    def create(self, name, schedule, duration, cutoff,
allow_unassociated_targets):
        """
        Create an AWS Systems Manager maintenance window.

        :param name: The name of the maintenance window.
        :param schedule: The schedule of the maintenance window.
        :param duration: The duration of the maintenance window.
        :param cutoff: The cutoff time of the maintenance window.
        :param allow_unassociated_targets: Allow the maintenance window to run on
managed nodes, even
                                           if you haven't registered those nodes
as targets.
        """
        try:
            response = self.ssm_client.create_maintenance_window(
                Name=name,
                Schedule=schedule,
                Duration=duration,
                Cutoff=cutoff,
                AllowUnassociatedTargets=allow_unassociated_targets,
            )
            self.window_id = response["WindowId"]
            self.name = name
            logger.info("Created maintenance window %s.", self.window_id)
        except ParamValidationError as error:
            logger.error(
                "Parameter validation error when trying to create maintenance
window %s. Here's why: %s",
                self.window_id,
                error,
            )
            raise
        except ClientError as err:
            logger.error(
                "Couldn't create maintenance window %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Einzelheiten zur API finden Sie CreateMaintenanceWindowin AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CreateOpsItem`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `CreateOpsItem` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Kennenlernen der Grundlagen

CLI

AWS CLI

Um ein zu erstellen OpsItems

Im folgenden `create-ops-item` Beispiel wird der Schlüssel /aws/resources verwendet, OperationalData um eine OpsItem mit einer Amazon DynamoDB DynamoDB-bezogene Ressource zu erstellen.

```
aws ssm create-ops-item \
    --title "EC2 instance disk full" \
    --description "Log clean up may have failed which caused the disk to be full"
 \
    --priority 2 \
    --source ec2 \
    --operational-data '{"/aws/resources":{"Value":"[{\"arn
\": \"arn:aws:dynamodb:us-west-2:12345678:table/OpsItems
\"}]","Type":"SearchableString"}}' \
    --notifications Arn="arn:aws:sns:us-west-2:12345678:TestUser"
```

Ausgabe:

```
{
    "OpsItemId": "oi-1a2b3c4d5e6f"
}
```

Weitere Informationen finden Sie unter Erstellen OpsItems im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie CreateOpsItemunter AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    /**
     * Creates an SSM OpsItem asynchronously.
     *
     * @param title The title of the OpsItem.
     * @param source The source of the OpsItem.
     * @param category The category of the OpsItem.
     * @param severity The severity of the OpsItem.
     * @return The ID of the created OpsItem.
     * <p>
     * This method initiates an asynchronous request to create an SSM OpsItem.
     * If the request is successful, it returns the OpsItem ID.
     * If an exception occurs, it handles the error appropriately.
     */
    public String createSSMOpsItem(String title, String source, String category,
String severity) {
        CreateOpsItemRequest opsItemRequest = CreateOpsItemRequest.builder()
                .description("Created by the SSM Java API")
                .title(title)
                .source(source)
                .category(category)
                .severity(severity)
                .build();
```

```
        CompletableFuture<CreateOpsItemResponse> future =
 getAsyncClient().createOpsItem(opsItemRequest);

        try {
            CreateOpsItemResponse response = future.join();
            return response.opsItemId();
        } catch (CompletionException e) {
            Throwable cause = e.getCause();
            if (cause instanceof SsmException) {
                throw (SsmException) cause;
            } else {
                throw new RuntimeException(cause);
            }
        }
    }
```

- Einzelheiten zur API finden Sie CreateOpsItemin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import { CreateOpsItemCommand, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * Create an SSM OpsItem.
 * @param {{ title: string, source: string, category?: string, severity?:
 string }}
 */
export const main = async ({
  title,
  source,
  category = undefined,
```

```
  severity = undefined,
}) => {
  const client = new SSMClient({});
  try {
    const { opsItemArn, opsItemId } = await client.send(
      new CreateOpsItemCommand({
        Title: title,
        Source: source, // The origin of the OpsItem, such as Amazon EC2 or
 Systems Manager.
        Category: category,
        Severity: severity,
      }),
    );
    console.log(`Ops item created with id: ${opsItemId}`);
    return { OpsItemArn: opsItemArn, OpsItemId: opsItemId };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "MissingParameter") {
      console.warn(`${caught.message}. Did you provide these values?`);
    } else {
      throw caught;
    }
  }
};
```

- Einzelheiten zur API finden Sie CreateOpsItemin der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class OpsItemWrapper:
    """Encapsulates AWS Systems Manager OpsItem actions."""
```

```
    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.id = None

    @classmethod
    def from_client(cls):
        """
        :return: A OpsItemWrapper instance.
        """
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def create(self, title, source, category, severity, description):
        """
        Create an OpsItem

        :param title: The OpsItem title.
        :param source: The OpsItem source.
        :param category: The OpsItem category.
        :param severity: The OpsItem severity.
        :param description: The OpsItem description.

        """
        try:
            response = self.ssm_client.create_ops_item(
                Title=title,
                Source=source,
                Category=category,
                Severity=severity,
                Description=description,
            )
            self.id = response["OpsItemId"]
        except self.ssm_client.exceptions.OpsItemLimitExceededException as err:
            logger.error(
                "Couldn't create ops item because you have exceeded your open
OpsItem limit. "
                "Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
```

```
            raise
        except ClientError as err:
            logger.error(
                "Couldn't create ops item %s. Here's why: %s: %s",
                title,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Einzelheiten zur API finden Sie CreateOpsItemin AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`CreatePatchBaseline`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `CreatePatchBaseline` verwendet wird.

CLI

AWS CLI

Beispiel 1: So erstellen Sie eine Patch-Baseline mit automatischer Genehmigung

Das folgende `create-patch-baseline`-Beispiel erstellt eine Patch-Baseline für Windows Server, die Patches sieben Tage nach ihrer Veröffentlichung durch Microsoft für eine Produktionsumgebung genehmigt.

```
aws ssm create-patch-baseline \
    --name "Windows-Production-Baseline-AutoApproval" \
    --operating-system "WINDOWS" \
    --approval-
rules "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,1
{Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,UpdateRollups,CriticalUpdates]}]},App
 \
    --description "Baseline containing all updates approved for Windows Server
 production systems"
```

Ausgabe:

```
{
    "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

Beispiel 2: So erstellen Sie eine Patch-Baseline mit einem Stichtag für die Genehmigung

Im folgenden `create-patch-baseline`-Beispiel wird eine Patch-Baseline für Windows Server erstellt, die alle Patches für eine Produktionsumgebung genehmigt, die am oder vor dem 7. Juli 2020 veröffentlicht wurden.

```
aws ssm create-patch-baseline \
    --name "Windows-Production-Baseline-AutoApproval" \
    --operating-system "WINDOWS" \
    --approval-
rules "PatchRules=[{PatchFilterGroup={PatchFilters=[{Key=MSRC_SEVERITY,Values=[Critical,1
{Key=CLASSIFICATION,Values=[SecurityUpdates,Updates,UpdateRollups,CriticalUpdates]}]},App
 \
    --description "Baseline containing all updates approved for Windows Server
 production systems"
```

Ausgabe:

```
{
    "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

Beispiel 3: So erstellen Sie eine Patch-Baseline mit Genehmigungsregeln, die in einer JSON-Datei gespeichert sind

Im folgenden `create-patch-baseline`-Beispiel wird eine Patch-Baseline für Amazon Linux 2017.09 erstellt, die Patches für eine Produktionsumgebung sieben Tage nach ihrer Veröffentlichung genehmigt, Genehmigungsregeln für die Patch-Baseline festlegt und ein benutzerdefiniertes Repository für Patches festlegt.

```
aws ssm create-patch-baseline \
    --cli-input-json file://my-amazon-linux-approval-rules-and-repo.json
```

Inhalt von `my-amazon-linux-approval-rules-and-repo.json`:

```
{
    "Name": "Amazon-Linux-2017.09-Production-Baseline",
    "Description": "My approval rules patch baseline for Amazon Linux 2017.09
 instances",
    "OperatingSystem": "AMAZON_LINUX",
    "Tags": [
        {
            "Key": "Environment",
            "Value": "Production"
        }
    ],
    "ApprovalRules": {
        "PatchRules": [
            {
                "ApproveAfterDays": 7,
                "EnableNonSecurity": true,
                "PatchFilterGroup": {
                    "PatchFilters": [
                        {
                            "Key": "SEVERITY",
                            "Values": [
                                "Important",
                                "Critical"
                            ]
                        },
                        {
                            "Key": "CLASSIFICATION",
                            "Values": [
                                "Security",
                                "Bugfix"
                            ]
                        },
                        {
                            "Key": "PRODUCT",
                            "Values": [
                                "AmazonLinux2017.09"
                            ]
                        }
                    ]
                }
            }
        ]
    },
```

```
    "Sources": [
        {
            "Name": "My-AL2017.09",
            "Products": [
                "AmazonLinux2017.09"
            ],
            "Configuration": "[amzn-main] \nname=amzn-main-Base
\nmirrorlist=http://repo./$awsregion./$awsdomain//$releasever/main/
mirror.list //nmirrorlist_expire=300//nmetadata_expire=300 \npriority=10
 \nfailovermethod=priority \nfastestmirror_enabled=0 \ngpgcheck=1
 \ngpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-amazon-ga \nenabled=1 \nretries=3
 \ntimeout=5\nreport_instanceid=yes"
        }
    ]
}
```

Beispiel 4: So erstellen Sie eine Patch-Baseline, die genehmigte und abgelehnte Patches angibt

Im folgenden `create-patch-baseline`-Beispiel werden Patches, die genehmigt und abgelehnt werden sollen, ausdrücklich als Ausnahme von den Standard-Genehmigungsregeln angegeben.

```
aws ssm create-patch-baseline \
    --name "Amazon-Linux-2017.09-Alpha-Baseline" \
    --description "My custom approve/reject patch baseline for Amazon Linux
 2017.09 instances" \
    --operating-system "AMAZON_LINUX" \
    --approved-patches "CVE-2018-1234567,example-pkg-EE-2018*.amzn1.noarch" \
    --approved-patches-compliance-level "HIGH" \
    --approved-patches-enable-non-security \
    --tags "Key=Environment,Value=Alpha"
```

Weitere Informationen finden Sie unter Benutzerdefineirte Patch-Baseline erstellen im AWS - Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie CreatePatchBaselinein der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird eine Patch-Baseline erstellt, die Patches sieben Tage nach der Veröffentlichung durch Microsoft für verwaltete Instances genehmigt, auf denen Windows Server 2019 in einer Produktionsumgebung ausgeführt wird.

```
$rule = New-Object Amazon.SimpleSystemsManagement.Model.PatchRule
$rule.ApproveAfterDays = 7

$ruleFilters = New-Object Amazon.SimpleSystemsManagement.Model.PatchFilterGroup

$patchFilter = New-Object Amazon.SimpleSystemsManagement.Model.PatchFilter
$patchFilter.Key="PRODUCT"
$patchFilter.Values="WindowsServer2019"

$severityFilter = New-Object Amazon.SimpleSystemsManagement.Model.PatchFilter
$severityFilter.Key="MSRC_SEVERITY"
$severityFilter.Values.Add("Critical")
$severityFilter.Values.Add("Important")
$severityFilter.Values.Add("Moderate")

$classificationFilter = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchFilter
$classificationFilter.Key = "CLASSIFICATION"
$classificationFilter.Values.Add( "SecurityUpdates" )
$classificationFilter.Values.Add( "Updates" )
$classificationFilter.Values.Add( "UpdateRollups" )
$classificationFilter.Values.Add( "CriticalUpdates" )

$ruleFilters.PatchFilters.Add($severityFilter)
$ruleFilters.PatchFilters.Add($classificationFilter)
$ruleFilters.PatchFilters.Add($patchFilter)
$rule.PatchFilterGroup = $ruleFilters

New-SSMPatchBaseline -Name "Production-Baseline-Windows2019" -Description
 "Baseline containing all updates approved for production systems" -
ApprovalRules_PatchRule $rule
```

Ausgabe:

```
pb-0z4z6221c4296b23z
```

- Einzelheiten zur API finden Sie unter CreatePatchBaseline AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird eine Patch-Baseline erstellt, die Patches sieben Tage nach der Veröffentlichung durch Microsoft für verwaltete Instances genehmigt, auf denen Windows Server 2019 in einer Produktionsumgebung ausgeführt wird.

```
$rule = New-Object Amazon.SimpleSystemsManagement.Model.PatchRule
$rule.ApproveAfterDays = 7

$ruleFilters = New-Object Amazon.SimpleSystemsManagement.Model.PatchFilterGroup

$patchFilter = New-Object Amazon.SimpleSystemsManagement.Model.PatchFilter
$patchFilter.Key="PRODUCT"
$patchFilter.Values="WindowsServer2019"

$severityFilter = New-Object Amazon.SimpleSystemsManagement.Model.PatchFilter
$severityFilter.Key="MSRC_SEVERITY"
$severityFilter.Values.Add("Critical")
$severityFilter.Values.Add("Important")
$severityFilter.Values.Add("Moderate")

$classificationFilter = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchFilter
$classificationFilter.Key = "CLASSIFICATION"
$classificationFilter.Values.Add( "SecurityUpdates" )
$classificationFilter.Values.Add( "Updates" )
$classificationFilter.Values.Add( "UpdateRollups" )
$classificationFilter.Values.Add( "CriticalUpdates" )

$ruleFilters.PatchFilters.Add($severityFilter)
$ruleFilters.PatchFilters.Add($classificationFilter)
$ruleFilters.PatchFilters.Add($patchFilter)
$rule.PatchFilterGroup = $ruleFilters

New-SSMPatchBaseline -Name "Production-Baseline-Windows2019" -Description
 "Baseline containing all updates approved for production systems" -
ApprovalRules_PatchRule $rule
```

Ausgabe:

```
pb-0z4z6221c4296b23z
```

- Einzelheiten zur API finden Sie unter CreatePatchBaseline AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DeleteActivation`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DeleteActivation` verwendet wird.

CLI

AWS CLI

So löschen Sie eine Aktivierung für eine verwaltete Instance

Im folgenden Beispiel für `delete-activation` wird die Aktivierung einer verwalteten Instance gelöscht.

```
aws ssm delete-activation \
    --activation-id "aa673477-d926-42c1-8757-1358cEXAMPLE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Setting Up AWS Systems Manager for Hybrid Environments im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DeleteActivationunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird eine Aktivierung gelöscht. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Remove-SSMActivation -ActivationId "08e51e79-1e36-446c-8e63-9458569c1363"
```

- Einzelheiten zur API finden Sie unter DeleteActivation AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird eine Aktivierung gelöscht. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Remove-SSMActivation -ActivationId "08e51e79-1e36-446c-8e63-9458569c1363"
```

- Einzelheiten zur API finden Sie unter DeleteActivation AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DeleteAssociation`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DeleteAssociation` verwendet wird.

CLI

AWS CLI

Beispiel 1: So löschen Sie eine Zuordnung mithilfe der Zuordnungs-ID

Im folgenden Beispiel `delete-association` wird die Zuordnung für die angegebene Zuordnungs-ID gelöscht. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
aws ssm delete-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Bearbeiten und Erstellen einer neuen Version einer Zuordnung im AWS -Systems-Manager-Benutzerhandbuch.

Beispiel 2: So löschen Sie eine Zuordnung

Im folgenden Beispiel für `delete-association` wird die Verknüpfung zwischen einer Instance und einem Dokument gelöscht. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
aws ssm delete-association \
    --instance-id "i-1234567890abcdef0" \
    --name "AWS-UpdateSSMAgent"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DeleteAssociationin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird die Verknüpfung zwischen einer Instance und einem Dokument gelöscht. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Remove-SSMAssociation -InstanceId "i-0cb2b964d3e14fd9f" -Name "AWS-
UpdateSSMAgent"
```

- Einzelheiten zur API finden Sie unter DeleteAssociation AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird die Verknüpfung zwischen einer Instance und einem Dokument gelöscht. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Remove-SSMAssociation -InstanceId "i-0cb2b964d3e14fd9f" -Name "AWS-
UpdateSSMAgent"
```

- Einzelheiten zur API finden Sie unter DeleteAssociation AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeleteDocument`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DeleteDocument` verwendet wird.

CLI

AWS CLI

Löschen eines Dokuments

Das folgende `delete-document`-Beispiel löscht ein Systems-Manager-Dokument.

```
aws ssm delete-document \
    --name "Example"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Erstellen von Systems-Manager-Dokumenten im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DeleteDocumentin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    /**
     * Deletes an AWS SSM document asynchronously.
     *
     * @param documentName The name of the document to delete.
```

```
     * <p>
     * This method initiates an asynchronous request to delete an SSM document.
     * If an exception occurs, it handles the error appropriately.
     */
    public void deleteDoc(String documentName) {
        DeleteDocumentRequest documentRequest = DeleteDocumentRequest.builder()
                .name(documentName)
                .build();

        CompletableFuture<Void> future = CompletableFuture.runAsync(() -> {
            getAsyncClient().deleteDocument(documentRequest)
                    .thenAccept(response -> {
                        System.out.println("The SSM document was successfully
deleted.");
                    })
                    .exceptionally(ex -> {
                        throw new CompletionException(ex);
                    }).join();
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
                throw new RuntimeException("SSM error: " + cause.getMessage(),
cause);
            } else {
                throw new RuntimeException("Unexpected error: " +
cause.getMessage(), cause);
            }
        });

        try {
            future.join();
        } catch (CompletionException ex) {
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
ex.getCause() : ex;
        }
    }
```

- Einzelheiten zur API finden Sie DeleteDocumentin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import { DeleteDocumentCommand, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * Delete an SSM document.
 * @param {{ documentName: string }}
 */
export const main = async ({ documentName }) => {
  const client = new SSMClient({});
  try {
    await client.send(new DeleteDocumentCommand({ Name: documentName }));
    console.log(`Document '${documentName}' deleted.`);
    return { Deleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "MissingParameter") {
      console.warn(`${caught.message}. Did you provide this value?`);
    } else {
      throw caught;
    }
  }
};
```

- Einzelheiten zur API finden Sie DeleteDocumentin der AWS SDK für JavaScript API-Referenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird ein Dokument gelöscht. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Remove-SSMDocument -Name "RunShellScript"
```

- Einzelheiten zur API finden Sie unter DeleteDocument AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird ein Dokument gelöscht. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Remove-SSMDocument -Name "RunShellScript"
```

- Einzelheiten zur API finden Sie unter DeleteDocument AWS -Tools für PowerShellCmdlet-Referenz (V5).

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class DocumentWrapper:
    """Encapsulates AWS Systems Manager Document actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.name = None
```

```
    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def delete(self):
        """
        Deletes an AWS Systems Manager document.
        """
        if self.name is None:
            return

        try:
            self.ssm_client.delete_document(Name=self.name)
            print(f"Deleted document {self.name}.")
            self.name = None
        except ClientError as err:
            logger.error(
                "Couldn't delete %s. Here's why: %s: %s",
                self.name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Einzelheiten zur API finden Sie DeleteDocumentin AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeleteMaintenanceWindow`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DeleteMaintenanceWindow` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Kennenlernen der Grundlagen

CLI

AWS CLI

Löschen eines Wartungsfensters

In diesem `delete-maintenance-window`-Beispiel wird das angegebene Wartungsfenster entfernt.

```
aws ssm delete-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9"
```

Ausgabe:

```
{
    "WindowId":"mw-1a2b3c4d5e6f7g8h9"
}
```

Weitere Informationen finden Sie unter Löschen eines Wartungsfensters (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DeleteMaintenanceWindowunter AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    /**
     * Deletes an AWS SSM Maintenance Window asynchronously.
     *
     * @param winId The ID of the Maintenance Window to delete.
```

```
    * <p>
    * This method initiates an asynchronous request to delete an SSM Maintenance
Window.
    * If an exception occurs, it handles the error appropriately.
    */
   public void deleteMaintenanceWindow(String winId) {
       DeleteMaintenanceWindowRequest windowRequest =
DeleteMaintenanceWindowRequest.builder()
               .windowId(winId)
               .build();

       CompletableFuture<Void> future = CompletableFuture.runAsync(() -> {
           getAsyncClient().deleteMaintenanceWindow(windowRequest)
                   .thenAccept(response -> {
                       System.out.println("The maintenance window was
successfully deleted.");
                   })
                   .exceptionally(ex -> {
                       throw new CompletionException(ex);
                   }).join();
       }).exceptionally(ex -> {
           Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
           if (cause instanceof SsmException) {
               throw new RuntimeException("SSM error: " + cause.getMessage(),
cause);
           } else {
               throw new RuntimeException("Unexpected error: " +
cause.getMessage(), cause);
           }
       });

       try {
           future.join();
       } catch (CompletionException ex) {
           throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
ex.getCause() : ex;
       }
   }
```

- Einzelheiten zur API finden Sie DeleteMaintenanceWindowin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import { DeleteMaintenanceWindowCommand, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * Delete an SSM maintenance window.
 * @param {{ windowId: string }}
 */
export const main = async ({ windowId }) => {
  const client = new SSMClient({});
  try {
    await client.send(
      new DeleteMaintenanceWindowCommand({ WindowId: windowId }),
    );
    console.log(`Maintenance window '${windowId}' deleted.`);
    return { Deleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "MissingParameter") {
      console.warn(`${caught.message}. Did you provide this value?`);
    } else {
      throw caught;
    }
  }
};
```

- Einzelheiten zur API finden Sie DeleteMaintenanceWindowin der AWS SDK für JavaScript API-Referenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird ein Wartungsfenster entfernt.

```
Remove-SSMMaintenanceWindow -WindowId "mw-06d59c1a07c022145"
```

Ausgabe:

```
mw-06d59c1a07c022145
```

- Einzelheiten zur API finden Sie unter DeleteMaintenanceWindow AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird ein Wartungsfenster entfernt.

```
Remove-SSMMaintenanceWindow -WindowId "mw-06d59c1a07c022145"
```

Ausgabe:

```
mw-06d59c1a07c022145
```

- Einzelheiten zur API finden Sie unter DeleteMaintenanceWindow AWS -Tools für PowerShellCmdlet-Referenz (V5).

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class MaintenanceWindowWrapper:
```

```
    """Encapsulates AWS Systems Manager maintenance window actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.window_id = None
        self.name = None

    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def delete(self):
        """
        Delete the associated AWS Systems Manager maintenance window.
        """
        if self.window_id is None:
            return

        try:
            self.ssm_client.delete_maintenance_window(WindowId=self.window_id)
            logger.info("Deleted maintenance window %s.", self.window_id)
            print(f"Deleted maintenance window {self.name}")
            self.window_id = None
        except ClientError as err:
            logger.error(
                "Couldn't delete maintenance window %s. Here's why: %s: %s",
                self.window_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Einzelheiten zur API finden Sie DeleteMaintenanceWindowin AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeleteOpsItem`** mit einem AWS SDK

Das folgende Codebeispiel zeigt, wie es verwendet wird`DeleteOpsItem`.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class OpsItemWrapper:
    """Encapsulates AWS Systems Manager OpsItem actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.id = None

    @classmethod
    def from_client(cls):
        """
        :return: A OpsItemWrapper instance.
        """
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def delete(self):
        """
        Delete the OpsItem.
        """
        if self.id is None:
```

```
            return
        try:
            self.ssm_client.delete_ops_item(OpsItemId=self.id)
            print(f"Deleted ops item with id {self.id}")
            self.id = None
        except ClientError as err:
            logger.error(
                "Couldn't delete ops item %s. Here's why: %s: %s",
                self.id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Einzelheiten zur API finden Sie DeleteOpsItemin AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DeleteParameter`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DeleteParameter` verwendet wird.

CLI

AWS CLI

So löschen Sie einen Parameter

Das folgende `delete-parameter`Beispiel löscht die angegebene einzelne Parameter.

```
aws ssm delete-parameter \
    --name "MyParameter"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Arbeiten mit Parameter Store im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DeleteParameterin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird ein Parameter gelöscht. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Remove-SSMParameter -Name "helloWorld"
```

- Einzelheiten zur API finden Sie unter DeleteParameter AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird ein Parameter gelöscht. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Remove-SSMParameter -Name "helloWorld"
```

- Einzelheiten zur API finden Sie unter DeleteParameter AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DeletePatchBaseline`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DeletePatchBaseline` verwendet wird.

CLI

AWS CLI

So löschen Sie eine Patch-Baseline

Das folgende `delete-patch-baseline`-Beispiel löscht die angegebene Patch-Baseline.

```
aws ssm delete-patch-baseline \
```

```
    --baseline-id "pb-045f10b4f382baeda"
```

Ausgabe:

```
{
    "BaselineId": "pb-045f10b4f382baeda"
}
```

Weitere Informationen finden Sie unter Aktualisieren und Löschen einer Patch-Baseline (Konsole) im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DeletePatchBaselinein der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird eine Patch-Baseline gelöscht.

```
Remove-SSMPatchBaseline -BaselineId "pb-045f10b4f382baeda"
```

Ausgabe:

```
pb-045f10b4f382baeda
```

- Einzelheiten zur API finden Sie unter DeletePatchBaseline AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird eine Patch-Baseline gelöscht.

```
Remove-SSMPatchBaseline -BaselineId "pb-045f10b4f382baeda"
```

Ausgabe:

```
pb-045f10b4f382baeda
```

- Einzelheiten zur API finden Sie unter DeletePatchBaseline AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DeregisterManagedInstance`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DeregisterManagedInstance` verwendet wird.

CLI

AWS CLI

So heben Sie die Registrierung einer verwalteten Instance auf

Im folgenden `deregister-managed-instance`-Beispiel wird die Registrierung der angegebenen verwalteten Instance aufgehoben.

```
aws ssm deregister-managed-instance \
    --instance-id 'mi-08ab247cdfEXAMPLE'
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Abmelden verwalteter Knoten in einer Hybrid- und Multi-Cloud-Umgebung im Benutzerhandbuch zuAWS Systems Manager.

- Einzelheiten zur API finden Sie DeregisterManagedInstancein der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird die Registrierung einer verwalteten Instance aufgehoben. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Unregister-SSMManagedInstance -InstanceId "mi-08ab247cdf1046573"
```

- Einzelheiten zur API finden Sie unter DeregisterManagedInstance AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird die Registrierung einer verwalteten Instance aufgehoben. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Unregister-SSMManagedInstance -InstanceId "mi-08ab247cdf1046573"
```

- Einzelheiten zur API finden Sie unter DeregisterManagedInstance AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DeregisterPatchBaselineForPatchGroup`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DeregisterPatchBaselineForPatchGroup` verwendet wird.

CLI

AWS CLI

So heben Sie die Registrierung einer Patch-Gruppe für eine Patch-Baseline auf

Im folgenden `deregister-patch-baseline-for-patch-group`-Beispiel wird die Registrierung der angegebenen Patchgruppe von der angegebenen Patch-Baseline aufgehoben.

```
aws ssm deregister-patch-baseline-for-patch-group \
    --patch-group "Production" \
    --baseline-id "pb-0ca44a362fEXAMPLE"
```

Ausgabe:

```
{
  "PatchGroup":"Production",
  "BaselineId":"pb-0ca44a362fEXAMPLE"
}
```

Weitere Informationen finden Sie unter Hinzufügen einer Patch-Gruppe zu einer Patch-Baseline im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DeregisterPatchBaselineForPatchGroupin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird die Registrierung einer Patchgruppe von einer Patch-Baseline aufgehoben.

```
Unregister-SSMPatchBaselineForPatchGroup -BaselineId "pb-045f10b4f382baeda" -
PatchGroup "Production"
```

Ausgabe:

```
BaselineId           PatchGroup
----------           ----------
pb-045f10b4f382baeda Production
```

- Einzelheiten zur API finden Sie unter DeregisterPatchBaselineForPatchGroup AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird die Registrierung einer Patchgruppe von einer Patch-Baseline aufgehoben.

```
Unregister-SSMPatchBaselineForPatchGroup -BaselineId "pb-045f10b4f382baeda" -
PatchGroup "Production"
```

Ausgabe:

```
BaselineId           PatchGroup
----------           ----------
pb-045f10b4f382baeda Production
```

- Einzelheiten zur API finden Sie unter DeregisterPatchBaselineForPatchGroup AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **DeregisterTargetFromMaintenanceWindow** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DeregisterTargetFromMaintenanceWindow` verwendet wird.

CLI

AWS CLI

So entfernen Sie ein Ziel aus einem Wartungsfenster

Im folgenden Beispiel `deregister-target-from-maintenance-window` wird das angegebene Ziel aus dem angegebenen Wartungsfenster entfernt.

```
aws ssm deregister-target-from-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --window-target-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
```

Ausgabe:

```
{
    "WindowId":"mw-ab12cd34ef56gh78",
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Weitere Informationen finden Sie unter Aktualisieren eines Wartungsfensters (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DeregisterTargetFromMaintenanceWindowunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird ein Ziel aus einem Wartungsfenster entfernt.

```
Unregister-SSMTargetFromMaintenanceWindow -WindowTargetId
 "6ab5c208-9fc4-4697-84b7-b02a6cc25f7d" -WindowId "mw-06cf17cbefcb4bf4f"
```

Ausgabe:

```
WindowId             WindowTargetId
--------             --------------
mw-06cf17cbefcb4bf4f 6ab5c208-9fc4-4697-84b7-b02a6cc25f7d
```

- Einzelheiten zur API finden Sie unter DeregisterTargetFromMaintenanceWindow AWS - Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird ein Ziel aus einem Wartungsfenster entfernt.

```
Unregister-SSMTargetFromMaintenanceWindow -WindowTargetId
 "6ab5c208-9fc4-4697-84b7-b02a6cc25f7d" -WindowId "mw-06cf17cbefcb4bf4f"
```

Ausgabe:

```
WindowId             WindowTargetId
--------             --------------
mw-06cf17cbefcb4bf4f 6ab5c208-9fc4-4697-84b7-b02a6cc25f7d
```

- Einzelheiten zur API finden Sie unter DeregisterTargetFromMaintenanceWindow AWS - Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DeregisterTaskFromMaintenanceWindow`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DeregisterTaskFromMaintenanceWindow` verwendet wird.

CLI

AWS CLI

So entfernen Sie eine Aufgabe aus einem Wartungsfenster

Im folgenden Beispiel `deregister-task-from-maintenance-window` wird die angegebene Aufgabe aus dem angegebenen Wartungsfenster entfernt.

```
aws ssm deregister-task-from-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --window-task-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c"
```

Ausgabe:

```
{
    "WindowTaskId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c",
    "WindowId":"mw-ab12cd34ef56gh78"
}
```

Weitere Informationen finden Sie unter Systems Manager Maintenance Windows Tutorials (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DeregisterTaskFromMaintenanceWindowunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird eine Aufgabe aus einem Wartungsfenster entfernt.

```
Unregister-SSMTaskFromMaintenanceWindow -WindowTaskId "f34a2c47-ddfd-4c85-
a88d-72366b69af1b" -WindowId "mw-03a342e62c96d31b0"
```

Ausgabe:

```
WindowId             WindowTaskId
--------             ------------
mw-03a342e62c96d31b0 f34a2c47-ddfd-4c85-a88d-72366b69af1b
```

- Einzelheiten zur API finden Sie unter DeregisterTaskFromMaintenanceWindow AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird eine Aufgabe aus einem Wartungsfenster entfernt.

```
Unregister-SSMTaskFromMaintenanceWindow -WindowTaskId "f34a2c47-ddfd-4c85-
a88d-72366b69af1b" -WindowId "mw-03a342e62c96d31b0"
```

Ausgabe:

```
WindowId             WindowTaskId
--------             ------------
mw-03a342e62c96d31b0 f34a2c47-ddfd-4c85-a88d-72366b69af1b
```

- Einzelheiten zur API finden Sie unter DeregisterTaskFromMaintenanceWindow AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeActivations`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeActivations` verwendet wird.

CLI

AWS CLI

Um Aktivierungen zu beschreiben

Das folgende `describe-activations` Beispiel listet Details zu den Aktivierungen in Ihrem AWS Konto auf.

```
aws ssm describe-activations
```

Ausgabe:

```
{
    "ActivationList": [
        {
            "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",
            "Description": "Example1",
            "IamRole": "HybridWebServersRole,
            "RegistrationLimit": 5,
            "RegistrationsCount": 5,
            "ExpirationDate": 1584316800.0,
            "Expired": false,
```

```
            "CreatedDate": 1581954699.792
        },
        {
            "ActivationId": "3ee0322b-f62d-40eb-b672-13ebfEXAMPLE",
            "Description": "Example2",
            "IamRole": "HybridDatabaseServersRole",
            "RegistrationLimit": 5,
            "RegistrationsCount": 5,
            "ExpirationDate": 1580515200.0,
            "Expired": true,
            "CreatedDate": 1578064132.002
        },
    ]
}
```

Weitere Informationen finden Sie unter Schritt 4: Aktivierung einer verwalteten Instance für eine Hybridumgebung erstellen im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeActivationsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dieses Beispiel enthält Details zu den Aktivierungen in Ihrem Konto.

```
Get-SSMActivation
```

Ausgabe:

```
ActivationId        : 08e51e79-1e36-446c-8e63-9458569c1363
CreatedDate         : 3/1/2017 12:01:51 AM
DefaultInstanceName : MyWebServers
Description         :
ExpirationDate      : 3/2/2017 12:01:51 AM
Expired             : False
IamRole             : AutomationRole
RegistrationLimit   : 10
RegistrationsCount  : 0
```

- Einzelheiten zur API finden Sie unter DescribeActivations AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dieses Beispiel enthält Details zu den Aktivierungen in Ihrem Konto.

```
Get-SSMActivation
```

Ausgabe:

```
ActivationId        : 08e51e79-1e36-446c-8e63-9458569c1363
CreatedDate         : 3/1/2017 12:01:51 AM
DefaultInstanceName : MyWebServers
Description         :
ExpirationDate      : 3/2/2017 12:01:51 AM
Expired             : False
IamRole             : AutomationRole
RegistrationLimit   : 10
RegistrationsCount  : 0
```

- Einzelheiten zur API finden Sie unter DescribeActivations AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeAssociation`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeAssociation` verwendet wird.

CLI

AWS CLI

Beispiel 1: So rufen Sie Details zu einer Zuordnung ab

Das folgende `describe-association`-Beispiel beschreibt die Zuordnung für die angegebene Zuordnungs-ID.

```
aws ssm describe-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Ausgabe:


```
{
    "AssociationDescription": {
        "Name": "AWS-GatherSoftwareInventory",
        "AssociationVersion": "1",
        "Date": 1534864780.995,
        "LastUpdateAssociationDate": 1543235759.81,
        "Overview": {
            "Status": "Success",
            "AssociationStatusAggregatedCount": {
                "Success": 2
            }
        },
        "DocumentVersion": "$DEFAULT",
        "Parameters": {
            "applications": [
                "Enabled"
            ],
            "awsComponents": [
                "Enabled"
            ],
            "customInventory": [
                "Enabled"
            ],
            "files": [
                ""
            ],
            "instanceDetailedInformation": [
                "Enabled"
            ],
            "networkConfig": [
                "Enabled"
            ],
            "services": [
                "Enabled"
            ],
            "windowsRegistry": [
                ""
            ],
            "windowsRoles": [
                "Enabled"
            ],
            "windowsUpdates": [
```

```
                "Enabled"
            ]
        },
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "*"
                ]
            }
        ],
        "ScheduleExpression": "rate(24 hours)",
        "LastExecutionDate": 1550501886.0,
        "LastSuccessfulExecutionDate": 1550501886.0,
        "AssociationName": "Inventory-Association"
    }
}
```

Weitere Informationen finden Sie unter Bearbeiten und Erstellen einer neuen Version einer Zuordnung im AWS -Systems-Manager-Benutzerhandbuch.

Beispiel 2: So rufen Sie Details zu einer Zuordnung für eine bestimmte Instance und ein bestimmtes Dokument ab

Das folgende `describe-association`-Beispiel beschreibt die Zuordnung zwischen einer Instance und einem Dokument.

```
aws ssm describe-association \
    --instance-id "i-1234567890abcdef0" \
    --name "AWS-UpdateSSMAgent"
```

Ausgabe:

```
{
    "AssociationDescription": {
        "Status": {
            "Date": 1487876122.564,
            "Message": "Associated with AWS-UpdateSSMAgent",
            "Name": "Associated"
        },
        "Name": "AWS-UpdateSSMAgent",
```

```
        "InstanceId": "i-1234567890abcdef0",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Associated",
            "AssociationStatusAggregatedCount": {
                "Pending": 1
            }
        },
        "AssociationId": "d8617c07-2079-4c18-9847-1234567890ab",
        "DocumentVersion": "$DEFAULT",
        "LastUpdateAssociationDate": 1487876122.564,
        "Date": 1487876122.564,
        "Targets": [
            {
                "Values": [
                    "i-1234567890abcdef0"
                ],
                "Key": "InstanceIds"
            }
        ]
    }
}
```

Weitere Informationen finden Sie unter Bearbeiten und Erstellen einer neuen Version einer Zuordnung im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeAssociationin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dieses Beispiel beschreibt die Zuordnung zwischen einer Instance und einem Dokument.

```
Get-SSMAssociation -InstanceId "i-0000293ffd8c57862" -Name "AWS-UpdateSSMAgent"
```

Ausgabe:

```
Name                 : AWS-UpdateSSMAgent
InstanceId           : i-0000293ffd8c57862
Date                 : 2/23/2017 6:55:22 PM
```

```
Status.Name           : Pending
Status.Date           : 2/20/2015 8:31:11 AM
Status.Message        : temp_status_change
Status.AdditionalInfo : Additional-Config-Needed
```

- Einzelheiten zur API finden Sie unter DescribeAssociation AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dieses Beispiel beschreibt die Zuordnung zwischen einer Instance und einem Dokument.

```
Get-SSMAssociation -InstanceId "i-0000293ffd8c57862" -Name "AWS-UpdateSSMAgent"
```

Ausgabe:

```
Name                  : AWS-UpdateSSMAgent
InstanceId            : i-0000293ffd8c57862
Date                  : 2/23/2017 6:55:22 PM
Status.Name           : Pending
Status.Date           : 2/20/2015 8:31:11 AM
Status.Message        : temp_status_change
Status.AdditionalInfo : Additional-Config-Needed
```

- Einzelheiten zur API finden Sie unter DescribeAssociation AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeAssociationExecutionTargets`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeAssociationExecutionTargets` verwendet wird.

CLI

AWS CLI

So rufen Sie Details zu einer Zuordnung ab

Das folgende `describe-association-execution-targets`-Beispiel beschreibt die angegebene Zuordnungsausführung.

```
aws ssm describe-association-execution-targets \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --execution-id "7abb6378-a4a5-4f10-8312-0123456789ab"
```

Ausgabe:

```
{
    "AssociationExecutionTargets": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
            "ResourceId": "i-1234567890abcdef0",
            "ResourceType": "ManagedInstance",
            "Status": "Success",
            "DetailedStatus": "Success",
            "LastExecutionDate": 1550505538.497,
            "OutputSource": {
                "OutputSourceId": "97fff367-fc5a-4299-aed8-0123456789ab",
                "OutputSourceType": "RunCommand"
            }
        }
    ]
}
```

Weitere Informationen finden Sie unter Anzeigen von Zuordnungsverläufen im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeAssociationExecutionTargetsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden die Ressourcen-ID und ihr Ausführungsstatus angezeigt, die Teil der Ausführungsziele der Zuordnung sind

```
Get-SSMAssociationExecutionTarget -AssociationId 123a45a0-
c678-9012-3456-78901234db5e -ExecutionId 123a45a0-c678-9012-3456-78901234db5e |
 Select-Object ResourceId, Status
```

Ausgabe:

```
ResourceId           Status
----------           ------
i-0b1b2a3456f7a890b  Success
i-01c12a45d6fc7a89f  Success
i-0a1caf234f56d7dc8  Success
i-012a3fd45af6dbcfe  Failed
i-0ddc1df23c4a5fb67  Success
```

Beispiel 2: Dieser Befehl überprüft die jeweilige Ausführung einer bestimmten Automatisierung seit gestern, der ein Befehlsdokument zugeordnet ist. Außerdem wird geprüft, ob die Ausführung der Zuordnung fehlgeschlagen ist, und wenn ja, werden die Details zum Befehlsaufruf für die Ausführung zusammen mit der Instance-ID angezeigt

```
$AssociationExecution= Get-SSMAssociationExecutionTarget -
AssociationId 1c234567-890f-1aca-a234-5a678d901cb0 -ExecutionId
 12345ca12-3456-2345-2b45-23456789012 |
    Where-Object {$_.LastExecutionDate -gt (Get-Date -Hour 00 -Minute
 00).AddDays(-1)}

foreach ($execution in $AssociationExecution) {
    if($execution.Status -ne 'Success'){
        Write-Output "There was an issue executing the association
 $($execution.AssociationId) on $($execution.ResourceId)"
        Get-SSMCommandInvocation -CommandId
 $execution.OutputSource.OutputSourceId -Detail:$true | Select-Object -
ExpandProperty CommandPlugins
    }
}
```

Ausgabe:

```
There was an issue executing the association 1c234567-890f-1aca-a234-5a678d901cb0
 on i-0a1caf234f56d7dc8
```

```
Name                   : aws:runPowerShellScript
Output                 :
                         ----------ERROR-------
                         failed to run commands: exit status 1
OutputS3BucketName     :
OutputS3KeyPrefix      :
OutputS3Region         : eu-west-1
ResponseCode           : 1
ResponseFinishDateTime : 5/29/2019 11:04:49 AM
ResponseStartDateTime  : 5/29/2019 11:04:49 AM
StandardErrorUrl       :
StandardOutputUrl      :
Status                 : Failed
StatusDetails          : Failed
```

- Einzelheiten zur API finden Sie unter DescribeAssociationExecutionTargets AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden die Ressourcen-ID und ihr Ausführungsstatus angezeigt, die Teil der Ausführungsziele der Zuordnung sind

```
Get-SSMAssociationExecutionTarget -AssociationId 123a45a0-
c678-9012-3456-78901234db5e -ExecutionId 123a45a0-c678-9012-3456-78901234db5e |
 Select-Object ResourceId, Status
```

Ausgabe:

```
ResourceId           Status
----------           ------
i-0b1b2a3456f7a890b  Success
i-01c12a45d6fc7a89f  Success
i-0a1caf234f56d7dc8  Success
i-012a3fd45af6dbcfe  Failed
i-0ddc1df23c4a5fb67  Success
```

Beispiel 2: Dieser Befehl überprüft die jeweilige Ausführung einer bestimmten Automatisierung seit gestern, der ein Befehlsdokument zugeordnet ist. Außerdem wird geprüft, ob die Ausführung der Zuordnung fehlgeschlagen ist, und wenn ja, werden die Details zum Befehlsaufruf für die Ausführung zusammen mit der Instance-ID angezeigt

```
$AssociationExecution= Get-SSMAssociationExecutionTarget -
AssociationId 1c234567-890f-1aca-a234-5a678d901cb0 -ExecutionId
 12345ca12-3456-2345-2b45-23456789012 |
    Where-Object {$_.LastExecutionDate -gt (Get-Date -Hour 00 -Minute
 00).AddDays(-1)}

foreach ($execution in $AssociationExecution) {
    if($execution.Status -ne 'Success'){
        Write-Output "There was an issue executing the association
 $($execution.AssociationId) on $($execution.ResourceId)"
        Get-SSMCommandInvocation -CommandId
 $execution.OutputSource.OutputSourceId -Detail:$true | Select-Object -
ExpandProperty CommandPlugins
    }
}
```

Ausgabe:

```
There was an issue executing the association 1c234567-890f-1aca-a234-5a678d901cb0
 on i-0a1caf234f56d7dc8


Name                   : aws:runPowerShellScript
Output                 :
                           ----------ERROR-------
                           failed to run commands: exit status 1
OutputS3BucketName     :
OutputS3KeyPrefix      :
OutputS3Region         : eu-west-1
ResponseCode           : 1
ResponseFinishDateTime : 5/29/2019 11:04:49 AM
ResponseStartDateTime  : 5/29/2019 11:04:49 AM
StandardErrorUrl       :
StandardOutputUrl      :
Status                 : Failed
StatusDetails          : Failed
```

- Einzelheiten zur API finden Sie unter DescribeAssociationExecutionTargets AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeAssociationExecutions`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeAssociationExecutions` verwendet wird.

CLI

AWS CLI

Beispiel 1: So erhalten Sie Details zu allen Ausführungen für eine Zuordnung

Das folgende `describe-association-executions`-Beispiel beschreibt alle Ausführungen der angegebenen Zuordnung.

```
aws ssm describe-association-executions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Ausgabe:

```
{
    "AssociationExecutions": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "474925ef-1249-45a2-b93d-0123456789ab",
            "Status": "Success",
            "DetailedStatus": "Success",
            "CreatedTime": 1550505827.119,
            "ResourceCountByStatus": "{Success=1}"
        },
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
            "Status": "Success",
            "DetailedStatus": "Success",
            "CreatedTime": 1550505536.843,
            "ResourceCountByStatus": "{Success=1}"
        },
```

```
        ...
    ]
}
```

Weitere Informationen finden Sie unter Anzeigen von Zuordnungsverläufen im AWS -Systems-Manager-Benutzerhandbuch.

Beispiel 2: So erhalten Sie Details zu allen Ausführungen für eine Zuordnung nach einem bestimmten Datum und einer bestimmten Uhrzeit

Im folgenden `describe-association-executions`-Beispiel werden alle Ausführungen einer Zuordnung nach dem angegebenen Datum und der angegebenen Uhrzeit beschrieben.

```
aws ssm describe-association-executions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --filters "Key=CreatedTime,Value=2019-02-18T16:00:00Z,Type=GREATER_THAN"
```

Ausgabe:

```
{
    "AssociationExecutions": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "474925ef-1249-45a2-b93d-0123456789ab",
            "Status": "Success",
            "DetailedStatus": "Success",
            "CreatedTime": 1550505827.119,
            "ResourceCountByStatus": "{Success=1}"
        },
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
            "Status": "Success",
            "DetailedStatus": "Success",
            "CreatedTime": 1550505536.843,
            "ResourceCountByStatus": "{Success=1}"
        },
        ...
    ]
}
```

Weitere Informationen finden Sie unter Anzeigen von Zuordnungsverläufen im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeAssociationExecutionsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden die Ausführungen für die angegebene Zuordnungs-ID zurückgegeben

```
Get-SSMAssociationExecution -AssociationId 123a45a0-c678-9012-3456-78901234db5e
```

Ausgabe:

```
AssociationId         : 123a45a0-c678-9012-3456-78901234db5e
AssociationVersion    : 2
CreatedTime           : 3/2/2019 8:53:29 AM
DetailedStatus        :
ExecutionId           : 123a45a0-c678-9012-3456-78901234db5e
LastExecutionDate     : 1/1/0001 12:00:00 AM
ResourceCountByStatus : {Success=4}
Status                : Success
```

- Einzelheiten zur API finden Sie unter DescribeAssociationExecutions AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden die Ausführungen für die angegebene Zuordnungs-ID zurückgegeben

```
Get-SSMAssociationExecution -AssociationId 123a45a0-c678-9012-3456-78901234db5e
```

Ausgabe:

```
AssociationId         : 123a45a0-c678-9012-3456-78901234db5e
AssociationVersion    : 2
```

```
CreatedTime           : 3/2/2019 8:53:29 AM
DetailedStatus        :
ExecutionId           : 123a45a0-c678-9012-3456-78901234db5e
LastExecutionDate     : 1/1/0001 12:00:00 AM
ResourceCountByStatus : {Success=4}
Status                : Success
```

- Einzelheiten zur API finden Sie unter DescribeAssociationExecutions AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeAutomationExecutions`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeAutomationExecutions` verwendet wird.

CLI

AWS CLI

So beschreiben Sie die Ausführung einer Automatisierung

Im folgenden `describe-automation-executions`-Beispiel werden Details einer Automation-Ausführung gezeigt.

```
aws ssm describe-automation-executions \
    --filters Key=ExecutionId,Values=73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Ausgabe:

```
{
    "AutomationExecutionMetadataList": [
        {
            "AutomationExecutionId": "73c8eef8-f4ee-4a05-820c-e354fEXAMPLE",
            "DocumentName": "AWS-StartEC2Instance",
            "DocumentVersion": "1",
            "AutomationExecutionStatus": "Success",
            "ExecutionStartTime": 1583737233.748,
            "ExecutionEndTime": 1583737234.719,
```

```
            "ExecutedBy": "arn:aws:sts::29884EXAMPLE:assumed-role/
mw_service_role/OrchestrationService",
            "LogFile": "",
            "Outputs": {},
            "Mode": "Auto",
            "Targets": [],
            "ResolvedTargets": {
                "ParameterValues": [],
                "Truncated": false
            },
            "AutomationType": "Local"
        }
    ]
}
```

Weitere Informationen finden Sie unter Ausführen eines einfachen Automation-Workflows im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeAutomationExecutionsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden alle aktiven und beendeten Automation-Ausführungen beschrieben, die mit Ihrem Konto verknüpft sind.

```
Get-SSMAutomationExecutionList
```

Ausgabe:

```
AutomationExecutionId     : 4105a4fc-f944-11e6-9d32-8fb2db27a909
AutomationExecutionStatus : Failed
DocumentName              : AWS-UpdateLinuxAmi
DocumentVersion           : 1
ExecutedBy                : admin
ExecutionEndTime          : 2/22/2017 9:17:08 PM
ExecutionStartTime        : 2/22/2017 9:17:02 PM
LogFile                   :
Outputs                   : {[createImage.ImageId,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
```

Beispiel 2: In diesem Beispiel werden die Ausführungs-ID, das Dokument und der start/end Ausführungszeitstempel für Ausführungen angezeigt, bei denen es sich nicht um „Success“ AutomationExecutionStatus handelt

```
Get-SSMAutomationExecutionList | Where-Object AutomationExecutionStatus
 -ne "Success" | Select-Object AutomationExecutionId, DocumentName,
 AutomationExecutionStatus, ExecutionStartTime, ExecutionEndTime | Format-Table -
AutoSize
```

Ausgabe:

```
AutomationExecutionId                DocumentName
 AutomationExecutionStatus ExecutionStartTime   ExecutionEndTime
--------------------                 ------------
 ------------------------ ------------------   ----------------
e1d2bad3-4567-8901-ae23-456c7c8901be AWS-UpdateWindowsAmi
 Cancelled                 4/16/2019 5:37:04 AM 4/16/2019 5:47:29 AM
61234567-a7f8-90e1-2b34-567b8bf9012c Fixed-UpdateAmi
 Cancelled                 4/16/2019 5:33:04 AM 4/16/2019 5:40:15 AM
91234d56-7e89-0ac1-2aee-34ea5d6a7c89 AWS-UpdateWindowsAmi
 Failed                    4/16/2019 5:22:46 AM 4/16/2019 5:27:29 AM
```

- Einzelheiten zur API finden Sie unter AWS -Tools für PowerShell Cmdlet-Referenz (DescribeAutomationExecutionsV4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden alle aktiven und beendeten Automation-Ausführungen beschrieben, die mit Ihrem Konto verknüpft sind.

```
Get-SSMAutomationExecutionList
```

Ausgabe:

```
AutomationExecutionId     : 4105a4fc-f944-11e6-9d32-8fb2db27a909
AutomationExecutionStatus : Failed
DocumentName              : AWS-UpdateLinuxAmi
DocumentVersion           : 1
ExecutedBy                : admin
ExecutionEndTime          : 2/22/2017 9:17:08 PM
ExecutionStartTime        : 2/22/2017 9:17:02 PM
```

```
LogFile                    :
Outputs                    : {[createImage.ImageId,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
```

Beispiel 2: In diesem Beispiel werden die Ausführungs-ID, das Dokument und der start/end Ausführungszeitstempel für Ausführungen angezeigt, bei denen es sich nicht um „Success“ handelt AutomationExecutionStatus

```
Get-SSMAutomationExecutionList | Where-Object AutomationExecutionStatus
 -ne "Success" | Select-Object AutomationExecutionId, DocumentName,
 AutomationExecutionStatus, ExecutionStartTime, ExecutionEndTime | Format-Table -
AutoSize
```

Ausgabe:

```
AutomationExecutionId                DocumentName
 AutomationExecutionStatus ExecutionStartTime   ExecutionEndTime
---------------------                ------------
 ------------------------ ------------------   ----------------
e1d2bad3-4567-8901-ae23-456c7c8901be AWS-UpdateWindowsAmi
 Cancelled                 4/16/2019 5:37:04 AM 4/16/2019 5:47:29 AM
61234567-a7f8-90e1-2b34-567b8bf9012c Fixed-UpdateAmi
 Cancelled                 4/16/2019 5:33:04 AM 4/16/2019 5:40:15 AM
91234d56-7e89-0ac1-2aee-34ea5d6a7c89 AWS-UpdateWindowsAmi
 Failed                    4/16/2019 5:22:46 AM 4/16/2019 5:27:29 AM
```

- Einzelheiten zur API finden Sie unter AWS -Tools für PowerShell Cmdlet-Referenz (DescribeAutomationExecutionsV5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeAutomationStepExecutions`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeAutomationStepExecutions` verwendet wird.

CLI

AWS CLI

Beispiel 1: So können Sie alle Schritte für eine Automatisierungsausführung beschreiben

Im folgenden `describe-automation-step-executions`-Beispiel werden Details zu den Schritten einer Automation-Ausführung gezeigt.

```
aws ssm describe-automation-step-executions \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Ausgabe:

```
{
    "StepExecutions": [
        {
            "StepName": "startInstances",
            "Action": "aws:changeInstanceState",
            "ExecutionStartTime": 1583737234.134,
            "ExecutionEndTime": 1583737234.672,
            "StepStatus": "Success",
            "Inputs": {
                "DesiredState": "\"running\"",
                "InstanceIds": "[\"i-0cb99161f6EXAMPLE\"]"
            },
            "Outputs": {
                "InstanceStates": [
                    "running"
                ]
            },
            "StepExecutionId": "95e70479-cf20-4d80-8018-7e4e2EXAMPLE",
            "OverriddenParameters": {}
        }
    ]
}
```

Beispiel 2: So können Sie einen bestimmten Schritt für eine Automatisierungsausführung beschreiben

Im folgenden `describe-automation-step-executions`-Beispiel werden Details zu einem bestimmten Schritt einer Automation-Ausführung gezeigt.

```
aws ssm describe-automation-step-executions \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \
    --filters Key=StepExecutionId,Values=95e70479-cf20-4d80-8018-7e4e2EXAMPLE
```

Weitere Informationen finden Sie unter Schrittweises Ausführen eines Automation-Workflows (Befehlszeile) im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeAutomationStepExecutionsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden Informationen über alle aktiven und beendeten Schrittausführungen in einem Automation-Workflow angezeigt.

```
Get-SSMAutomationStepExecution -AutomationExecutionId e1d2bad3-4567-8901-
ae23-456c7c8901be | Select-Object StepName, Action, StepStatus
```

Ausgabe:

```
StepName                 Action                   StepStatus
--------                 ------                   ----------
LaunchInstance           aws:runInstances         Success
OSCompatibilityCheck     aws:runCommand           Success
RunPreUpdateScript       aws:runCommand           Success
UpdateEC2Config          aws:runCommand           Cancelled
UpdateSSMAgent           aws:runCommand           Pending
UpdateAWSPVDriver        aws:runCommand           Pending
UpdateAWSEnaNetworkDriver aws:runCommand          Pending
UpdateAWSNVMe            aws:runCommand           Pending
InstallWindowsUpdates    aws:runCommand           Pending
RunPostUpdateScript      aws:runCommand           Pending
RunSysprepGeneralize     aws:runCommand           Pending
StopInstance             aws:changeInstanceState Pending
CreateImage              aws:createImage          Pending
TerminateInstance        aws:changeInstanceState Pending
```

- Einzelheiten zur API finden Sie unter DescribeAutomationStepExecutions AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden Informationen über alle aktiven und beendeten Schrittausführungen in einem Automation-Workflow angezeigt.

```
Get-SSMAutomationStepExecution -AutomationExecutionId e1d2bad3-4567-8901-
ae23-456c7c8901be | Select-Object StepName, Action, StepStatus
```

Ausgabe:

```
StepName                  Action                   StepStatus
--------                  ------                   ----------
LaunchInstance            aws:runInstances         Success
OSCompatibilityCheck      aws:runCommand           Success
RunPreUpdateScript        aws:runCommand           Success
UpdateEC2Config           aws:runCommand           Cancelled
UpdateSSMAgent            aws:runCommand           Pending
UpdateAWSPVDriver         aws:runCommand           Pending
UpdateAWSEnaNetworkDriver aws:runCommand           Pending
UpdateAWSNVMe             aws:runCommand           Pending
InstallWindowsUpdates     aws:runCommand           Pending
RunPostUpdateScript       aws:runCommand           Pending
RunSysprepGeneralize      aws:runCommand           Pending
StopInstance              aws:changeInstanceState Pending
CreateImage               aws:createImage          Pending
TerminateInstance         aws:changeInstanceState Pending
```

- Einzelheiten zur API finden Sie unter DescribeAutomationStepExecutions AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeAvailablePatches`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeAvailablePatches` verwendet wird.

CLI

AWS CLI

So erhalten Sie verfügbare Patches

Im folgenden Beispiel `describe-available-patches` werden Details zu allen verfügbaren Patches für Windows Server 2019 abgerufen, die den MSRC-Schweregrad Kritisch haben.

```
aws ssm describe-available-patches \
    --
filters "Key=PRODUCT,Values=WindowsServer2019" "Key=MSRC_SEVERITY,Values=Critical"
```

Ausgabe:

```
{
    "Patches": [
        {
            "Id": "fe6bd8c2-3752-4c8b-ab3e-1a7ed08767ba",
            "ReleaseDate": 1544047205.0,
            "Title": "2018-11 Update for Windows Server 2019 for x64-based
 Systems (KB4470788)",
            "Description": "Install this update to resolve issues in Windows.
 For a complete listing of the issues that are included in this update, see the
 associated Microsoft Knowledge Base article for more information. After you
 install this item, you may have to restart your computer.",
            "ContentUrl": "https://support.microsoft.com/en-us/kb/4470788",
            "Vendor": "Microsoft",
            "ProductFamily": "Windows",
            "Product": "WindowsServer2019",
            "Classification": "SecurityUpdates",
            "MsrcSeverity": "Critical",
            "KbNumber": "KB4470788",
            "MsrcNumber": "",
            "Language": "All"
        },
        {
            "Id": "c96115e1-5587-4115-b851-22baa46a3f11",
            "ReleaseDate": 1549994410.0,
            "Title": "2019-02 Security Update for Adobe Flash Player for Windows
 Server 2019 for x64-based Systems (KB4487038)",
            "Description": "A security issue has been identified in a Microsoft
 software product that could affect your system. You can help protect your system
```

```
 by installing this update from Microsoft. For a complete listing of the issues
 that are included in this update, see the associated Microsoft Knowledge Base
 article. After you install this update, you may have to restart your system.",
            "ContentUrl": "https://support.microsoft.com/en-us/kb/4487038",
            "Vendor": "Microsoft",
            "ProductFamily": "Windows",
            "Product": "WindowsServer2019",
            "Classification": "SecurityUpdates",
            "MsrcSeverity": "Critical",
            "KbNumber": "KB4487038",
            "MsrcNumber": "",
            "Language": "All"
        },
        ...
    ]
}
```

So erhalten Sie Details zu einem bestimmten Patch

Im folgenden `describe-available-patches`-Beispiel werden Details zum angegebenen Patch abgerufen.

```
aws ssm describe-available-patches \
    --filters "Key=PATCH_ID,Values=KB4480979"
```

Ausgabe:

```
{
    "Patches": [
        {
            "Id": "680861e3-fb75-432e-818e-d72e5f2be719",
            "ReleaseDate": 1546970408.0,
            "Title": "2019-01 Security Update for Adobe Flash Player for Windows
 Server 2016 for x64-based Systems (KB4480979)",
            "Description": "A security issue has been identified in a Microsoft
 software product that could affect your system. You can help protect your system
 by installing this update from Microsoft. For a complete listing of the issues
 that are included in this update, see the associated Microsoft Knowledge Base
 article. After you install this update, you may have to restart your system.",
            "ContentUrl": "https://support.microsoft.com/en-us/kb/4480979",
            "Vendor": "Microsoft",
            "ProductFamily": "Windows",
```

```
            "Product": "WindowsServer2016",
            "Classification": "SecurityUpdates",
            "MsrcSeverity": "Critical",
            "KbNumber": "KB4480979",
            "MsrcNumber": "",
            "Language": "All"
        }
    ]
}
```

Weitere Informationen finden Sie unter So funktionieren Patch-Manager-Operationen im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeAvailablePatchesin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dieses Beispiel ruft alle verfügbaren Patches für Windows Server 2012 ab, die den MSRC-Schweregrad „Kritisch“ aufweisen. Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$filter1 = @{Key="PRODUCT";Values=@("WindowsServer2012")}
$filter2 = @{Key="MSRC_SEVERITY";Values=@("Critical")}

Get-SSMAvailablePatch -Filter $filter1,$filter2
```

Ausgabe:

```
Classification : SecurityUpdates
ContentUrl     : https://support.microsoft.com/en-us/kb/2727528
Description    : A security issue has been identified that could allow an
 unauthenticated remote attacker to compromise your system and gain control
                  over it. You can help protect your system by installing this
 update from Microsoft. After you install this update, you may have to
                  restart your system.
Id             : 1eb507be-2040-4eeb-803d-abc55700b715
KbNumber       : KB2727528
Language       : All
MsrcNumber     : MS12-072
MsrcSeverity   : Critical
```

```
Product        : WindowsServer2012
ProductFamily  : Windows
ReleaseDate    : 11/13/2012 6:00:00 PM
Title          : Security Update for Windows Server 2012 (KB2727528)
Vendor         : Microsoft
...
```

Beispiel 2: Bei PowerShell Version 2 müssen Sie New-Object verwenden, um jeden Filter zu erstellen.

```
$filter1 = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchOrchestratorFilter
$filter1.Key = "PRODUCT"
$filter1.Values = "WindowsServer2012"
$filter2 = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchOrchestratorFilter
$filter2.Key = "MSRC_SEVERITY"
$filter2.Values = "Critical"

Get-SSMAvailablePatch -Filter $filter1,$filter2
```

Beispiel 3: In diesem Beispiel werden alle Updates abgerufen, die in den letzten 20 Tagen veröffentlicht wurden und für Produkte gelten, die 2019 entsprechen WindowsServer

```
Get-SSMAvailablePatch | Where-Object ReleaseDate -ge (Get-Date).AddDays(-20)
 | Where-Object Product -eq "WindowsServer2019" | Select-Object ReleaseDate,
 Product, Title
```

Ausgabe:

```
ReleaseDate         Product           Title
-----------         -------           -----
4/9/2019 5:00:12 PM WindowsServer2019 2019-04 Security Update for Adobe Flash
 Player for Windows Server 2019 for x64-based Systems (KB4493478)
4/9/2019 5:00:06 PM WindowsServer2019 2019-04 Cumulative Update for Windows
 Server 2019 for x64-based Systems (KB4493509)
4/2/2019 5:00:06 PM WindowsServer2019 2019-03 Servicing Stack Update for Windows
 Server 2019 for x64-based Systems (KB4493510)
```

- Einzelheiten zur API finden Sie unter DescribeAvailablePatches AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dieses Beispiel ruft alle verfügbaren Patches für Windows Server 2012 ab, die den MSRC-Schweregrad „Kritisch“ aufweisen. Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$filter1 = @{Key="PRODUCT";Values=@("WindowsServer2012")}
$filter2 = @{Key="MSRC_SEVERITY";Values=@("Critical")}

Get-SSMAvailablePatch -Filter $filter1,$filter2
```

Ausgabe:

```
Classification : SecurityUpdates
ContentUrl     : https://support.microsoft.com/en-us/kb/2727528
Description    : A security issue has been identified that could allow an
 unauthenticated remote attacker to compromise your system and gain control
                 over it. You can help protect your system by installing this
 update from Microsoft. After you install this update, you may have to
                 restart your system.
Id             : 1eb507be-2040-4eeb-803d-abc55700b715
KbNumber       : KB2727528
Language       : All
MsrcNumber     : MS12-072
MsrcSeverity   : Critical
Product        : WindowsServer2012
ProductFamily  : Windows
ReleaseDate    : 11/13/2012 6:00:00 PM
Title          : Security Update for Windows Server 2012 (KB2727528)
Vendor         : Microsoft
...
```

Beispiel 2: Bei PowerShell Version 2 müssen Sie New-Object verwenden, um jeden Filter zu erstellen.

```
$filter1 = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchOrchestratorFilter
$filter1.Key = "PRODUCT"
$filter1.Values = "WindowsServer2012"
$filter2 = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchOrchestratorFilter
$filter2.Key = "MSRC_SEVERITY"
```

```
$filter2.Values = "Critical"

Get-SSMAvailablePatch -Filter $filter1,$filter2
```

Beispiel 3: In diesem Beispiel werden alle Updates abgerufen, die in den letzten 20 Tagen veröffentlicht wurden und für Produkte gelten, die 2019 entsprechen WindowsServer

```
Get-SSMAvailablePatch | Where-Object ReleaseDate -ge (Get-Date).AddDays(-20)
 | Where-Object Product -eq "WindowsServer2019" | Select-Object ReleaseDate,
 Product, Title
```

Ausgabe:

```
ReleaseDate         Product           Title
-----------         -------           -----
4/9/2019 5:00:12 PM WindowsServer2019 2019-04 Security Update for Adobe Flash
 Player for Windows Server 2019 for x64-based Systems (KB4493478)
4/9/2019 5:00:06 PM WindowsServer2019 2019-04 Cumulative Update for Windows
 Server 2019 for x64-based Systems (KB4493509)
4/2/2019 5:00:06 PM WindowsServer2019 2019-03 Servicing Stack Update for Windows
 Server 2019 for x64-based Systems (KB4493510)
```

- Einzelheiten zur API finden Sie unter DescribeAvailablePatches AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeDocument`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeDocument` verwendet wird.

CLI

AWS CLI

So können Sie Details eines Dokuments anzeigen

Im folgenden `describe-document` Beispiel werden Details zu einem Systems Manager Manager-Dokument in Ihrem AWS Konto angezeigt.

```
aws ssm describe-document \
    --name "Example"
```

Ausgabe:

```
{
    "Document": {
        "Hash":
 "fc2410281f40779e694a8b95975d0f9f316da8a153daa94e3d9921102EXAMPLE",
        "HashType": "Sha256",
        "Name": "Example",
        "Owner": "29884EXAMPLE",
        "CreatedDate": 1583257938.266,
        "Status": "Active",
        "DocumentVersion": "1",
        "Description": "Document Example",
        "Parameters": [
            {
                "Name": "AutomationAssumeRole",
                "Type": "String",
                "Description": "(Required) The ARN of the role that allows
 Automation to perform the actions on your behalf. If no role is specified,
 Systems Manager Automation uses your IAM permissions to execute this document.",
                "DefaultValue": ""
            },
            {
                "Name": "InstanceId",
                "Type": "String",
                "Description": "(Required) The ID of the Amazon EC2 instance.",
                "DefaultValue": ""
            }
        ],
        "PlatformTypes": [
            "Windows",
            "Linux"
        ],
        "DocumentType": "Automation",
        "SchemaVersion": "0.3",
        "LatestVersion": "1",
        "DefaultVersion": "1",
        "DocumentFormat": "YAML",
        "Tags": []
    }
```

```
}
```

Weitere Informationen finden Sie unter Erstellen von Systems-Manager-Dokumenten im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeDocumentunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden Informationen zu einem Dokument zurückgegeben.

```
Get-SSMDocumentDescription -Name "RunShellScript"
```

Ausgabe:

```
CreatedDate     : 2/24/2017 5:25:13 AM
DefaultVersion  : 1
Description     : Run an updated script
DocumentType    : Command
DocumentVersion : 1
Hash            :
 f775e5df4904c6fa46686c4722fae9de1950dace25cd9608ff8d622046b68d9b
HashType        : Sha256
LatestVersion   : 1
Name            : RunShellScript
Owner           : 123456789012
Parameters      : {commands}
PlatformTypes   : {Linux}
SchemaVersion   : 2.0
Sha1            :
Status          : Active
```

- Einzelheiten zur API finden Sie unter DescribeDocument AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden Informationen zu einem Dokument zurückgegeben.

```
Get-SSMDocumentDescription -Name "RunShellScript"
```

Ausgabe:

```
CreatedDate     : 2/24/2017 5:25:13 AM
DefaultVersion  : 1
Description     : Run an updated script
DocumentType    : Command
DocumentVersion : 1
Hash            :
 f775e5df4904c6fa46686c4722fae9de1950dace25cd9608ff8d622046b68d9b
HashType        : Sha256
LatestVersion   : 1
Name            : RunShellScript
Owner           : 123456789012
Parameters      : {commands}
PlatformTypes   : {Linux}
SchemaVersion   : 2.0
Sha1            :
Status          : Active
```

- Einzelheiten zur API finden Sie unter DescribeDocument AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeDocumentPermission`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeDocumentPermission` verwendet wird.

CLI

AWS CLI

So können Sie die Berechtigungen für Dokumente beschreiben

Im folgenden `describe-document-permission`-Beispiel werden Berechtigungsdetails zu einem Systems-Manager-Dokument angezeigt, das öffentlich geteilt wird.

```
aws ssm describe-document-permission \
    --name "Example" \
```

```
    --permission-type "Share"
```

Ausgabe:

```
{
    "AccountIds": [
        "all"
    ],
    "AccountSharingInfoList": [
        {
            "AccountId": "all",
            "SharedDocumentVersion": "$DEFAULT"
        }
    ]
}
```

Weitere Informationen finden Sie unter Systems-Manager-Dokument teilen im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeDocumentPermissionin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden alle Versionen eines Dokuments aufgeführt.

```
Get-SSMDocumentVersionList -Name "RunShellScript"
```

Ausgabe:

```
CreatedDate          DocumentVersion IsDefaultVersion Name
-----------          --------------- ---------------- ----
2/24/2017 5:25:13 AM 1               True             RunShellScript
```

- Einzelheiten zur API finden Sie unter DescribeDocumentPermission AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden alle Versionen eines Dokuments aufgeführt.

```
Get-SSMDocumentVersionList -Name "RunShellScript"
```

Ausgabe:

```
CreatedDate          DocumentVersion IsDefaultVersion Name
-----------          --------------- ---------------- ----
2/24/2017 5:25:13 AM 1               True             RunShellScript
```

- Einzelheiten zur API finden Sie unter DescribeDocumentPermission AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeEffectiveInstanceAssociations`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeEffectiveInstanceAssociations` verwendet wird.

CLI

AWS CLI

Um Details zu den effektiven Verknüpfungen für eine Instance abzurufen

Im folgenden `describe-effective-instance-associations`-Beispiel werden Details zu den effektiven Verknüpfungen für eine Instance abgerufen.

Befehl:

```
aws ssm describe-effective-instance-associations --instance-
id "i-1234567890abcdef0"
```

Ausgabe:

```
{
    "Associations": [
```

```
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "InstanceId": "i-1234567890abcdef0",
            "Content": "{\n    \"schemaVersion\": \"1.2\",\n    \"description\":
 \"Update the Amazon SSM Agent to the latest version or specified version.\",\n
  \"parameters\": {\n        \"version\": {\n            \"default\": \"\",\n
       \"description\": \"(Optional) A specific version of the Amazon SSM Agent
 to install. If not specified, the agent will be updated to the latest version.
\",\n            \"type\": \"String\"\n        },\n        \"allowDowngrade
\": {\n            \"default\": \"false\",\n            \"description\":
 \"(Optional) Allow the Amazon SSM Agent service to be downgraded to an earlier
 version. If set to false, the service can be upgraded to newer versions only
 (default). If set to true, specify the earlier version.\",\n            \"type
\": \"String\",\n            \"allowedValues\": [\n                \"true\",\n
              \"false\"\n            ]\n        }\n    },\n    \"runtimeConfig
\": {\n        \"aws:updateSsmAgent\": {\n            \"properties\": [\n
         {\n                \"agentName\": \"amazon-ssm-agent\",\n
  \"source\": \"https://s3.{Region}.amazonaws.com/amazon-ssm-{Region}/ssm-agent-
manifest.json\",\n                \"allowDowngrade\": \"{{ allowDowngrade }}\",\n
               \"targetVersion\": \"{{ version }}\"\n                }\n
    ]\n        }\n    }\n}\n",
            "AssociationVersion": "1"
        }
    ]
}
```

- Einzelheiten zur API finden Sie DescribeEffectiveInstanceAssociationsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dieses Beispiel beschreibt die effektiven Zuordnungen für eine Instance.

```
Get-SSMEffectiveInstanceAssociationList -InstanceId "i-0000293ffd8c57862" -
MaxResult 5
```

Ausgabe:

```
AssociationId                       Content
-------------                       -------
```

```
d8617c07-2079-4c18-9847-1655fc2698b0 {...
```

Beispiel 2: In diesem Beispiel wird der Inhalt der effektiven Verknüpfungen für eine Instance angezeigt.

```
(Get-SSMEffectiveInstanceAssociationList -InstanceId "i-0000293ffd8c57862" -
MaxResult 5).Content
```

Ausgabe:

```
{
    "schemaVersion": "1.2",
    "description": "Update the Amazon SSM Agent to the latest version or
 specified version.",
    "parameters": {
        "version": {
            "default": "",
            "description": "(Optional) A specific version of the Amazon SSM Agent
 to install. If not specified, the agen
t will be updated to the latest version.",
            "type": "String"
        },
        "allowDowngrade": {
            "default": "false",
            "description": "(Optional) Allow the Amazon SSM Agent service to be
 downgraded to an earlier version. If set
 to false, the service can be upgraded to newer versions only (default). If set
 to true, specify the earlier version.",
            "type": "String",
            "allowedValues": [
                "true",
                "false"
            ]
        }
    },
    "runtimeConfig": {
        "aws:updateSsmAgent": {
            "properties": [
                {
                "agentName": "amazon-ssm-agent",
                "source": "https://s3.{Region}.amazonaws.com/amazon-ssm-{Region}/
ssm-agent-manifest.json",
                "allowDowngrade": "{{ allowDowngrade }}",
```

```
                "targetVersion": "{{ version }}"
                }
            ]
        }
    }
}
```

- Einzelheiten zur API finden Sie unter DescribeEffectiveInstanceAssociations AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dieses Beispiel beschreibt die effektiven Zuordnungen für eine Instance.

```
Get-SSMEffectiveInstanceAssociationList -InstanceId "i-0000293ffd8c57862" -
MaxResult 5
```

Ausgabe:

```
AssociationId                        Content
-------------                        -------
d8617c07-2079-4c18-9847-1655fc2698b0 {...
```

Beispiel 2: In diesem Beispiel wird der Inhalt der effektiven Verknüpfungen für eine Instance angezeigt.

```
(Get-SSMEffectiveInstanceAssociationList -InstanceId "i-0000293ffd8c57862" -
MaxResult 5).Content
```

Ausgabe:

```
{
    "schemaVersion": "1.2",
    "description": "Update the Amazon SSM Agent to the latest version or
 specified version.",
    "parameters": {
        "version": {
            "default": "",
            "description": "(Optional) A specific version of the Amazon SSM Agent
 to install. If not specified, the agen
t will be updated to the latest version.",
```

```
            "type": "String"
        },
        "allowDowngrade": {
            "default": "false",
            "description": "(Optional) Allow the Amazon SSM Agent service to be
 downgraded to an earlier version. If set
 to false, the service can be upgraded to newer versions only (default). If set
 to true, specify the earlier version.",
            "type": "String",
            "allowedValues": [
                "true",
                "false"
            ]
        }
    },
    "runtimeConfig": {
        "aws:updateSsmAgent": {
            "properties": [
                {
                "agentName": "amazon-ssm-agent",
                "source": "https://s3.{Region}.amazonaws.com/amazon-ssm-{Region}/
ssm-agent-manifest.json",
                "allowDowngrade": "{{ allowDowngrade }}",
                "targetVersion": "{{ version }}"
                }
            ]
        }
    }
}
```

- Einzelheiten zur API finden Sie unter DescribeEffectiveInstanceAssociations AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeEffectivePatchesForPatchBaseline`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeEffectivePatchesForPatchBaseline` verwendet wird.

CLI

AWS CLI

Beispiel 1: So erhalten Sie alle Patches, die durch eine benutzerdefinierte Patch-Baseline definiert sind

Im folgenden `describe-effective-patches-for-patch-baseline` Beispiel werden die Patches zurückgegeben, die durch eine benutzerdefinierte Patch-Baseline im aktuellen AWS Konto definiert sind. Beachten Sie, dass für eine benutzerdefinierte Baseline nur die ID für `--baseline-id` erforderlich ist.

```
aws ssm describe-effective-patches-for-patch-baseline \
    --baseline-id "pb-08b654cf9b9681f04"
```

Ausgabe:

```
{
    "EffectivePatches": [
        {
            "Patch": {
                "Id": "fe6bd8c2-3752-4c8b-ab3e-1a7ed08767ba",
                "ReleaseDate": 1544047205.0,
                "Title": "2018-11 Update for Windows Server 2019 for x64-based
 Systems (KB4470788)",
                "Description": "Install this update to resolve issues in Windows.
 For a complete listing of the issues that are included in this update, see the
 associated Microsoft Knowledge Base article for more information. After you
 install this item, you may have to restart your computer.",
                "ContentUrl": "https://support.microsoft.com/en-us/kb/4470788",
                "Vendor": "Microsoft",
                "ProductFamily": "Windows",
                "Product": "WindowsServer2019",
                "Classification": "SecurityUpdates",
                "MsrcSeverity": "Critical",
                "KbNumber": "KB4470788",
                "MsrcNumber": "",
                "Language": "All"
            },
            "PatchStatus": {
                "DeploymentStatus": "APPROVED",
                "ComplianceLevel": "CRITICAL",
```

```
                "ApprovalDate": 1544047205.0
            }
        },
        {
            "Patch": {
                "Id": "915a6b1a-f556-4d83-8f50-b2e75a9a7e58",
                "ReleaseDate": 1549994400.0,
                "Title": "2019-02 Cumulative Update for .NET Framework 3.5 and
 4.7.2 for Windows Server 2019 for x64 (KB4483452)",
                "Description": "A security issue has been identified in a
 Microsoft software product that could affect your system. You can help protect
 your system by installing this update from Microsoft. For a complete listing
 of the issues that are included in this update, see the associated Microsoft
 Knowledge Base article. After you install this update, you may have to restart
 your system.",
                "ContentUrl": "https://support.microsoft.com/en-us/kb/4483452",
                "Vendor": "Microsoft",
                "ProductFamily": "Windows",
                "Product": "WindowsServer2019",
                "Classification": "SecurityUpdates",
                "MsrcSeverity": "Important",
                "KbNumber": "KB4483452",
                "MsrcNumber": "",
                "Language": "All"
            },
            "PatchStatus": {
                "DeploymentStatus": "APPROVED",
                "ComplianceLevel": "CRITICAL",
                "ApprovalDate": 1549994400.0
            }
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

Beispiel 2: Um alle Patches abzurufen, die durch eine AWS verwaltete Patch-Baseline definiert sind

Im folgenden `describe-effective-patches-for-patch-baseline` Beispiel werden die durch eine AWS verwaltete Patch-Baseline definierten Patches zurückgegeben. Beachten Sie, dass für eine AWS verwaltete Baseline der vollständige Baseline-ARN erforderlich ist für `--baseline-id`

```
aws ssm describe-effective-patches-for-patch-baseline \
    --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-020d361a05defe4ed"
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter So werden Sicherheitspatches ausgewählt im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeEffectivePatchesForPatchBaselinein der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden alle Patch-Baselines mit einer maximalen Ergebnisliste von 1 aufgeführt.

```
Get-SSMEffectivePatchesForPatchBaseline -BaselineId "pb-0a2f1059b670ebd31" -
MaxResult 1
```

Ausgabe:

```
Patch                                       PatchStatus
-----                                       -----------
Amazon.SimpleSystemsManagement.Model.Patch
 Amazon.SimpleSystemsManagement.Model.PatchStatus
```

Beispiel 2: In diesem Beispiel wird der Patchstatus für alle Patch-Baselines mit einer maximalen Ergebnisliste von 1 angezeigt.

```
(Get-SSMEffectivePatchesForPatchBaseline -BaselineId "pb-0a2f1059b670ebd31" -
MaxResult 1).PatchStatus
```

Ausgabe:

```
ApprovalDate         DeploymentStatus
------------         ----------------
```

```
12/21/2010 6:00:00 PM APPROVED
```

- Einzelheiten zur API finden Sie unter DescribeEffectivePatchesForPatchBaseline AWS - Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden alle Patch-Baselines mit einer maximalen Ergebnisliste von 1 aufgeführt.

```
Get-SSMEffectivePatchesForPatchBaseline -BaselineId "pb-0a2f1059b670ebd31" -
MaxResult 1
```

Ausgabe:

```
Patch                                       PatchStatus
-----                                       -----------
Amazon.SimpleSystemsManagement.Model.Patch
 Amazon.SimpleSystemsManagement.Model.PatchStatus
```

Beispiel 2: In diesem Beispiel wird der Patchstatus für alle Patch-Baselines mit einer maximalen Ergebnisliste von 1 angezeigt.

```
(Get-SSMEffectivePatchesForPatchBaseline -BaselineId "pb-0a2f1059b670ebd31" -
MaxResult 1).PatchStatus
```

Ausgabe:

```
ApprovalDate          DeploymentStatus
------------          ----------------
12/21/2010 6:00:00 PM APPROVED
```

- Einzelheiten zur API finden Sie unter DescribeEffectivePatchesForPatchBaseline AWS - Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeInstanceAssociationsStatus`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeInstanceAssociationsStatus` verwendet wird.

CLI

AWS CLI

So beschreiben Sie den Status der Zuordnungen einer Instance

Dieses Beispiel zeigt Details zu den Zuordnungen für eine Instance.

Befehl:

```
aws ssm describe-instance-associations-status --instance-id "i-1234567890abcdef0"
```

Ausgabe:

```
{
  "InstanceAssociationStatusInfos": [
      {
          "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
          "Name": "AWS-GatherSoftwareInventory",
          "DocumentVersion": "1",
          "AssociationVersion": "1",
          "InstanceId": "i-1234567890abcdef0",
          "ExecutionDate": 1550501886.0,
          "Status": "Success",
          "ExecutionSummary": "1 out of 1 plugin processed, 1 success, 0 failed,
 0 timedout, 0 skipped. ",
          "AssociationName": "Inventory-Association"
      },
      {
          "AssociationId": "5c5a31f6-6dae-46f9-944c-0123456789ab",
          "Name": "AWS-UpdateSSMAgent",
          "DocumentVersion": "1",
          "AssociationVersion": "1",
          "InstanceId": "i-1234567890abcdef0",
          "ExecutionDate": 1550505828.548,
          "Status": "Success",
          "DetailedStatus": "Success",
          "AssociationName": "UpdateSSMAgent"
```

```
      }
  ]
}
```

- Einzelheiten zur API finden Sie DescribeInstanceAssociationsStatusin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dieses Beispiel zeigt Details zu den Zuordnungen für eine Instance.

```
Get-SSMInstanceAssociationsStatus -InstanceId "i-0000293ffd8c57862"
```

Ausgabe:

```
AssociationId    : d8617c07-2079-4c18-9847-1655fc2698b0
DetailedStatus   : Pending
DocumentVersion  : 1
ErrorCode        :
ExecutionDate    : 2/20/2015 8:31:11 AM
ExecutionSummary : temp_status_change
InstanceId       : i-0000293ffd8c57862
Name             : AWS-UpdateSSMAgent
OutputUrl        :
Status           : Pending
```

Beispiel 2: In diesem Beispiel wird der Status der Instance-Zuordnung für die angegebene Instance-ID überprüft und außerdem der Ausführungsstatus dieser Zuordnungen angezeigt

```
Get-SSMInstanceAssociationsStatus -InstanceId i-012e3cb4df567e8aa | ForEach-
Object {Get-SSMAssociationExecution -AssociationId .AssociationId}
```

Ausgabe:

```
AssociationId         : 512a34a5-c678-1234-1234-12345678db9e
AssociationVersion    : 2
CreatedTime           : 3/2/2019 8:53:29 AM
DetailedStatus        :
ExecutionId           : 512a34a5-c678-1234-1234-12345678db9e
```

```
LastExecutionDate     : 1/1/0001 12:00:00 AM
ResourceCountByStatus : {Success=9}
Status                : Success
```

- Einzelheiten zur API finden Sie unter DescribeInstanceAssociationsStatus AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dieses Beispiel zeigt Details zu den Zuordnungen für eine Instance.

```
Get-SSMInstanceAssociationsStatus -InstanceId "i-0000293ffd8c57862"
```

Ausgabe:

```
AssociationId    : d8617c07-2079-4c18-9847-1655fc2698b0
DetailedStatus   : Pending
DocumentVersion  : 1
ErrorCode        :
ExecutionDate    : 2/20/2015 8:31:11 AM
ExecutionSummary : temp_status_change
InstanceId       : i-0000293ffd8c57862
Name             : AWS-UpdateSSMAgent
OutputUrl        :
Status           : Pending
```

Beispiel 2: In diesem Beispiel wird der Status der Instance-Zuordnung für die angegebene Instance-ID überprüft und außerdem der Ausführungsstatus dieser Zuordnungen angezeigt

```
Get-SSMInstanceAssociationsStatus -InstanceId i-012e3cb4df567e8aa | ForEach-
Object {Get-SSMAssociationExecution -AssociationId .AssociationId}
```

Ausgabe:

```
AssociationId         : 512a34a5-c678-1234-1234-12345678db9e
AssociationVersion    : 2
CreatedTime           : 3/2/2019 8:53:29 AM
DetailedStatus        :
ExecutionId           : 512a34a5-c678-1234-1234-12345678db9e
LastExecutionDate     : 1/1/0001 12:00:00 AM
ResourceCountByStatus : {Success=9}
Status                : Success
```

- Einzelheiten zur API finden Sie unter DescribeInstanceAssociationsStatus AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeInstanceInformation`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeInstanceInformation` verwendet wird.

CLI

AWS CLI

Beispiel 1: So beschreiben Sie Informationen zu verwalteten Instances

Im folgenden `describe-instance-information`-Beispiel werden Details zu jeder Ihrer verwalteten Instances abgerufen.

```
aws ssm describe-instance-information
```

Beispiel 2: Um Informationen über eine bestimmte verwaltete Instance zu beschreiben

Das folgende `describe-instance-information`-Beispiel zeigt Details der verwalteten Instance `i-028ea792daEXAMPLE`.

```
aws ssm describe-instance-information \
    --filters "Key=InstanceIds,Values=i-028ea792daEXAMPLE"
```

Beispiel 3: So beschreiben Sie Informationen zu verwalteten Instances mit einem bestimmten Tag-Schlüssel

Das folgende `describe-instance-information`-Beispiel zeigt Details für verwaltete Instances, die über den Tag-Schlüssel `DEV` verfügen.

```
aws ssm describe-instance-information \
    --filters "Key=tag-key,Values=DEV"
```

Ausgabe:

```
{
    "InstanceInformationList": [
        {
            "InstanceId": "i-028ea792daEXAMPLE",
            "PingStatus": "Online",
            "LastPingDateTime": 1582221233.421,
            "AgentVersion": "2.3.842.0",
            "IsLatestVersion": true,
            "PlatformType": "Linux",
            "PlatformName": "SLES",
            "PlatformVersion": "15.1",
            "ResourceType": "EC2Instance",
            "IPAddress": "192.0.2.0",
            "ComputerName": "ip-198.51.100.0.us-east-2.compute.internal",
            "AssociationStatus": "Success",
            "LastAssociationExecutionDate": 1582220806.0,
            "LastSuccessfulAssociationExecutionDate": 1582220806.0,
            "AssociationOverview": {
                "DetailedStatus": "Success",
                "InstanceAssociationStatusAggregatedCount": {
                    "Success": 2
                }
            }
        }
    ]
}
```

Weitere Informationen finden Sie unter Verwaltete Instances im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeInstanceInformationin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dieses Beispiel zeigt Details zu jeder Ihrer Instances.

```
Get-SSMInstanceInformation
```

Ausgabe:

```
ActivationId                          :
AgentVersion                          : 2.0.672.0
AssociationOverview                   :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                     : Success
ComputerName                          : ip-172-31-44-222.us-
west-2.compute.internal
IamRole                               :
InstanceId                            : i-0cb2b964d3e14fd9f
IPAddress                             : 172.31.44.222
IsLatestVersion                       : True
LastAssociationExecutionDate          : 2/24/2017 3:18:09 AM
LastPingDateTime                      : 2/24/2017 3:35:03 AM
LastSuccessfulAssociationExecutionDate : 2/24/2017 3:18:09 AM
Name                                  :
PingStatus                            : ConnectionLost
PlatformName                          : Amazon Linux AMI
PlatformType                          : Linux
PlatformVersion                       : 2016.09
RegistrationDate                      : 1/1/0001 12:00:00 AM
ResourceType                          : EC2Instance
```

Beispiel 2: Dieses Beispiel zeigt, wie der Parameter -Filter verwendet wird, um Ergebnisse nur nach den AWS Systems Manager Manager-Instanzen in der Region **`us-east-1`** mit dem Wert **`AgentVersion`** von **`2.2.800.0`** zu filtern. Eine Liste der gültigen -Filter-Schlüsselwerte finden Sie im InstanceInformation API-Referenzthema (https://docs.aws.amazon.com/systems-manager/latest/APIReference/API_ InstanceInformation .html #systemsmanager -Type- - InstanceInformation). ActivationId

```
$Filters = @{
    Key="AgentVersion"
    Values="2.2.800.0"
}
Get-SSMInstanceInformation -Region us-east-1 -Filter $Filters
```

Ausgabe:

```
ActivationId                          :
AgentVersion                          : 2.2.800.0
AssociationOverview                   :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
```

```
AssociationStatus                       : Success
ComputerName                            : EXAMPLE-EXAMPLE.WORKGROUP
IamRole                                 :
InstanceId                              : i-EXAMPLEb0792d98ce
IPAddress                               : 10.0.0.01
IsLatestVersion                         : False
LastAssociationExecutionDate            : 8/16/2018 12:02:50 AM
LastPingDateTime                        : 8/16/2018 7:40:27 PM
LastSuccessfulAssociationExecutionDate : 8/16/2018 12:02:50 AM
Name                                    :
PingStatus                              : Online
PlatformName                            : Microsoft Windows Server 2016 Datacenter
PlatformType                            : Windows
PlatformVersion                         : 10.0.14393
RegistrationDate                        : 1/1/0001 12:00:00 AM
ResourceType                            : EC2Instance

ActivationId                            :
AgentVersion                            : 2.2.800.0
AssociationOverview                     :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                       : Success
ComputerName                            : EXAMPLE-EXAMPLE.WORKGROUP
IamRole                                 :
InstanceId                              : i-EXAMPLEac7501d023
IPAddress                               : 10.0.0.02
IsLatestVersion                         : False
LastAssociationExecutionDate            : 8/16/2018 12:00:20 AM
LastPingDateTime                        : 8/16/2018 7:40:35 PM
LastSuccessfulAssociationExecutionDate : 8/16/2018 12:00:20 AM
Name                                    :
PingStatus                              : Online
PlatformName                            : Microsoft Windows Server 2016 Datacenter
PlatformType                            : Windows
PlatformVersion                         : 10.0.14393
RegistrationDate                        : 1/1/0001 12:00:00 AM
ResourceType                            : EC2Instance
```

Beispiel 3: Dieses Beispiel zeigt, wie der InstanceInformationFilterList Parameter - verwendet wird, um Ergebnisse nur nach den AWS Systems Manager Manager-Instanzen in **`PlatformTypes`** der Region **`us-east-1`** mit **`Windows`** oder zu filtern**`Linux`**. Eine Liste der gültigen InstanceInformationFilterList Schlüsselwerte finden Sie im InstanceInformationFilter

API-Referenzthema (https://docs.aws.amazon.com/systems-manager/latest/APIReference/API_ InstanceInformationFilter .html).

```
$Filters = @{
   Key="PlatformTypes"
   ValueSet=("Windows","Linux")
}
Get-SSMInstanceInformation -Region us-east-1 -InstanceInformationFilterList
 $Filters
```

Ausgabe:

```
ActivationId                          :
AgentVersion                          : 2.2.800.0
AssociationOverview                   :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                     : Success
ComputerName                          : EXAMPLE-EXAMPLE.WORKGROUP
IamRole                               :
InstanceId                            : i-EXAMPLEb0792d98ce
IPAddress                             : 10.0.0.27
IsLatestVersion                       : False
LastAssociationExecutionDate          : 8/16/2018 12:02:50 AM
LastPingDateTime                      : 8/16/2018 7:40:27 PM
LastSuccessfulAssociationExecutionDate : 8/16/2018 12:02:50 AM
Name                                  :
PingStatus                            : Online
PlatformName                          : Ubuntu Server 18.04 LTS
PlatformType                          : Linux
PlatformVersion                       : 18.04
RegistrationDate                      : 1/1/0001 12:00:00 AM
ResourceType                          : EC2Instance

ActivationId                          :
AgentVersion                          : 2.2.800.0
AssociationOverview                   :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                     : Success
ComputerName                          : EXAMPLE-EXAMPLE.WORKGROUP
IamRole                               :
InstanceId                            : i-EXAMPLEac7501d023
IPAddress                             : 10.0.0.100
IsLatestVersion                       : False
```

```
LastAssociationExecutionDate           : 8/16/2018 12:00:20 AM
LastPingDateTime                       : 8/16/2018 7:40:35 PM
LastSuccessfulAssociationExecutionDate : 8/16/2018 12:00:20 AM
Name                                   :
PingStatus                             : Online
PlatformName                           : Microsoft Windows Server 2016 Datacenter
PlatformType                           : Windows
PlatformVersion                        : 10.0.14393
RegistrationDate                       : 1/1/0001 12:00:00 AM
ResourceType                           : EC2Instance
```

Beispiel 4: In diesem Beispiel werden von SSM verwaltete Instanzen und Exporte InstanceId LastPingDateTime sowie PlatformName in eine CSV-Datei aufgeführt. PingStatus

```
Get-SSMInstanceInformation | Select-Object InstanceId, PingStatus,
 LastPingDateTime, PlatformName | Export-Csv Instance-details.csv -
NoTypeInformation
```

- Einzelheiten zur API finden Sie unter DescribeInstanceInformation AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dieses Beispiel zeigt Details zu jeder Ihrer Instances.

```
Get-SSMInstanceInformation
```

Ausgabe:

```
ActivationId                           :
AgentVersion                           : 2.0.672.0
AssociationOverview                    :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                      : Success
ComputerName                           : ip-172-31-44-222.us-
west-2.compute.internal
IamRole                                :
InstanceId                             : i-0cb2b964d3e14fd9f
IPAddress                              : 172.31.44.222
IsLatestVersion                        : True
LastAssociationExecutionDate           : 2/24/2017 3:18:09 AM
LastPingDateTime                       : 2/24/2017 3:35:03 AM
LastSuccessfulAssociationExecutionDate : 2/24/2017 3:18:09 AM
```

```
Name                                    :
PingStatus                              : ConnectionLost
PlatformName                            : Amazon Linux AMI
PlatformType                            : Linux
PlatformVersion                         : 2016.09
RegistrationDate                        : 1/1/0001 12:00:00 AM
ResourceType                            : EC2Instance
```

Beispiel 2: Dieses Beispiel zeigt, wie der Parameter -Filter verwendet wird, um Ergebnisse nur nach den AWS Systems Manager Manager-Instanzen in der Region **`us-east-1`** mit dem Wert **`AgentVersion`** von **`2.2.800.0`** zu filtern. Eine Liste der gültigen -Filter-Schlüsselwerte finden Sie im InstanceInformation API-Referenzthema (https://docs.aws.amazon.com/systems-manager/latest/APIReference/API_ InstanceInformation .html #systemsmanager -Type- - InstanceInformation). ActivationId

```
$Filters = @{
    Key="AgentVersion"
    Values="2.2.800.0"
}
Get-SSMInstanceInformation -Region us-east-1 -Filter $Filters
```

Ausgabe:

```
ActivationId                            :
AgentVersion                            : 2.2.800.0
AssociationOverview                     :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                       : Success
ComputerName                            : EXAMPLE-EXAMPLE.WORKGROUP
IamRole                                 :
InstanceId                              : i-EXAMPLEb0792d98ce
IPAddress                               : 10.0.0.01
IsLatestVersion                         : False
LastAssociationExecutionDate            : 8/16/2018 12:02:50 AM
LastPingDateTime                        : 8/16/2018 7:40:27 PM
LastSuccessfulAssociationExecutionDate : 8/16/2018 12:02:50 AM
Name                                    :
PingStatus                              : Online
PlatformName                            : Microsoft Windows Server 2016 Datacenter
PlatformType                            : Windows
PlatformVersion                         : 10.0.14393
RegistrationDate                        : 1/1/0001 12:00:00 AM
```

```
ResourceType                            : EC2Instance

ActivationId                            :
AgentVersion                            : 2.2.800.0
AssociationOverview                     :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                       : Success
ComputerName                            : EXAMPLE-EXAMPLE.WORKGROUP
IamRole                                 :
InstanceId                              : i-EXAMPLEac7501d023
IPAddress                               : 10.0.0.02
IsLatestVersion                         : False
LastAssociationExecutionDate            : 8/16/2018 12:00:20 AM
LastPingDateTime                        : 8/16/2018 7:40:35 PM
LastSuccessfulAssociationExecutionDate : 8/16/2018 12:00:20 AM
Name                                    :
PingStatus                              : Online
PlatformName                            : Microsoft Windows Server 2016 Datacenter
PlatformType                            : Windows
PlatformVersion                         : 10.0.14393
RegistrationDate                        : 1/1/0001 12:00:00 AM
ResourceType                            : EC2Instance
```

Beispiel 3: Dieses Beispiel zeigt, wie der InstanceInformationFilterList Parameter - verwendet wird, um Ergebnisse nur nach den AWS Systems Manager Manager-Instanzen in **`PlatformTypes`** der Region **`us-east-1`** mit **`Windows`** oder zu filtern**`Linux`**. Eine Liste der gültigen InstanceInformationFilterList Schlüsselwerte finden Sie im InstanceInformationFilter API-Referenzthema (https://docs.aws.amazon.com/systems-manager/latest/APIReference/API_ InstanceInformationFilter .html).

```
$Filters = @{
   Key="PlatformTypes"
   ValueSet=("Windows","Linux")
}
Get-SSMInstanceInformation -Region us-east-1 -InstanceInformationFilterList
 $Filters
```

Ausgabe:

```
ActivationId                            :
AgentVersion                            : 2.2.800.0
```

```
AssociationOverview                     :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                       : Success
ComputerName                            : EXAMPLE-EXAMPLE.WORKGROUP
IamRole                                 :
InstanceId                              : i-EXAMPLEb0792d98ce
IPAddress                               : 10.0.0.27
IsLatestVersion                         : False
LastAssociationExecutionDate            : 8/16/2018 12:02:50 AM
LastPingDateTime                        : 8/16/2018 7:40:27 PM
LastSuccessfulAssociationExecutionDate : 8/16/2018 12:02:50 AM
Name                                    :
PingStatus                              : Online
PlatformName                            : Ubuntu Server 18.04 LTS
PlatformType                            : Linux
PlatformVersion                         : 18.04
RegistrationDate                        : 1/1/0001 12:00:00 AM
ResourceType                            : EC2Instance

ActivationId                            :
AgentVersion                            : 2.2.800.0
AssociationOverview                     :
 Amazon.SimpleSystemsManagement.Model.InstanceAggregatedAssociationOverview
AssociationStatus                       : Success
ComputerName                            : EXAMPLE-EXAMPLE.WORKGROUP
IamRole                                 :
InstanceId                              : i-EXAMPLEac7501d023
IPAddress                               : 10.0.0.100
IsLatestVersion                         : False
LastAssociationExecutionDate            : 8/16/2018 12:00:20 AM
LastPingDateTime                        : 8/16/2018 7:40:35 PM
LastSuccessfulAssociationExecutionDate : 8/16/2018 12:00:20 AM
Name                                    :
PingStatus                              : Online
PlatformName                            : Microsoft Windows Server 2016 Datacenter
PlatformType                            : Windows
PlatformVersion                         : 10.0.14393
RegistrationDate                        : 1/1/0001 12:00:00 AM
ResourceType                            : EC2Instance
```

Beispiel 4: In diesem Beispiel werden von SSM verwaltete Instanzen und Exporte InstanceId LastPingDateTime sowie PlatformName in eine CSV-Datei aufgeführt. PingStatus

```
Get-SSMInstanceInformation | Select-Object InstanceId, PingStatus,
 LastPingDateTime, PlatformName | Export-Csv Instance-details.csv -
NoTypeInformation
```

- Einzelheiten zur API finden Sie unter DescribeInstanceInformation AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeInstancePatchStates`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeInstancePatchStates` verwendet wird.

CLI

AWS CLI

Um die Status der Patch-Zusammenfassung für Instances abzurufen

In diesem `describe-instance-patch-states`-Beispiel werden die Status der Patch-Zusammenfassung für eine Instance abgerufen.

```
aws ssm describe-instance-patch-states \
    --instance-ids "i-1234567890abcdef0"
```

Ausgabe:

```
{
    "InstancePatchStates": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "PatchGroup": "my-patch-group",
            "BaselineId": "pb-0713accee01234567",
            "SnapshotId": "521c3536-930c-4aa9-950e-01234567abcd",
            "CriticalNonCompliantCount": 2,
            "SecurityNonCompliantCount": 2,
            "OtherNonCompliantCount": 1,
            "InstalledCount": 123,
            "InstalledOtherCount": 334,
```

```
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 1,
            "FailedCount": 2,
            "UnreportedNotApplicableCount": 11,
            "NotApplicableCount": 2063,
            "OperationStartTime": "2021-05-03T11:00:56-07:00",
            "OperationEndTime": "2021-05-03T11:01:09-07:00",
            "Operation": "Scan",
            "LastNoRebootInstallOperationTime": "2020-06-14T12:17:41-07:00",
            "RebootOption": "RebootIfNeeded"
        }
    ]
}
```

Weitere Informationen finden Sie unter Über Patch-Compliance im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeInstancePatchStatesin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden die Status der Patch-Zusammenfassung für eine Instance abgerufen.

```
Get-SSMInstancePatchState -InstanceId "i-08ee91c0b17045407"
```

Beispiel 2: In diesem Beispiel werden die Status der Patch-Zusammenfassung für zwei Instances abgerufen.

```
Get-SSMInstancePatchState -InstanceId "i-08ee91c0b17045407","i-09a618aec652973a9"
```

- Einzelheiten zur API finden Sie unter DescribeInstancePatchStates AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden die Status der Patch-Zusammenfassung für eine Instance abgerufen.

```
Get-SSMInstancePatchState -InstanceId "i-08ee91c0b17045407"
```

Beispiel 2: In diesem Beispiel werden die Status der Patch-Zusammenfassung für zwei Instances abgerufen.

```
Get-SSMInstancePatchState -InstanceId "i-08ee91c0b17045407","i-09a618aec652973a9"
```

- Einzelheiten zur API finden Sie unter DescribeInstancePatchStates AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeInstancePatchStatesForPatchGroup`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeInstancePatchStatesForPatchGroup` verwendet wird.

CLI

AWS CLI

Beispiel 1: Um die Instance-Status für eine Patchgruppe abzurufen

Im folgenden `describe-instance-patch-states-for-patch-group`-Beispiel werden Details zu den Status der Patchzusammenfassung pro Instance für die angegebene Patchgruppe abgerufen.

```
aws ssm describe-instance-patch-states-for-patch-group \
    --patch-group "Production"
```

Ausgabe:

```
{
    "InstancePatchStates": [
        {
            "InstanceId": "i-02573cafcfEXAMPLE",
            "PatchGroup": "Production",
            "BaselineId": "pb-0c10e65780EXAMPLE",
```

```
            "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
            "OwnerInformation": "",
            "InstalledCount": 32,
            "InstalledOtherCount": 1,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 2,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": 2671,
            "NotApplicableCount": 400,
            "OperationStartTime": "2021-08-04T11:03:50.590000-07:00",
            "OperationEndTime": "2021-08-04T11:04:21.555000-07:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 0
        },
        {
            "InstanceId": "i-0471e04240EXAMPLE",
            "PatchGroup": "Production",
            "BaselineId": "pb-09ca3fb51fEXAMPLE",
            "SnapshotId": "05d8ffb0-1bbe-4812-ba2d-d9b7bEXAMPLE",
            "OwnerInformation": "",
            "InstalledCount": 32,
            "InstalledOtherCount": 1,
            "InstalledPendingRebootCount": 0,
            "InstalledRejectedCount": 0,
            "MissingCount": 2,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": 2671,
            "NotApplicableCount": 400,
            "OperationStartTime": "2021-08-04T22:06:20.340000-07:00",
            "OperationEndTime": "2021-08-04T22:07:11.220000-07:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 0
        }
    ]
}
```

Beispiel 2: Um den Instance-Status für eine Patch-Gruppe mit mehr als fünf fehlenden Patches abzurufen

Das folgende `describe-instance-patch-states-for-patch-group`-Beispiel ruft Details zu den Patch-Zusammenfassungszuständen für die angegebene Patch-Gruppe für Instances mit mehr als fünf fehlenden Patches ab.

```
aws ssm describe-instance-patch-states-for-patch-group \
    --filters Key=MissingCount,Type=GreaterThan,Values=5 \
    --patch-group "Production"
```

Ausgabe:

```
{
    "InstancePatchStates": [
        {
            "InstanceId": "i-02573cafcfEXAMPLE",
            "PatchGroup": "Production",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
            "OwnerInformation": "",
            "InstalledCount": 46,
            "InstalledOtherCount": 4,
            "InstalledPendingRebootCount": 1,
            "InstalledRejectedCount": 1,
            "MissingCount": 7,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": 232,
            "NotApplicableCount": 654,
            "OperationStartTime": "2021-08-04T11:03:50.590000-07:00",
            "OperationEndTime": "2021-08-04T11:04:21.555000-07:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 1
        }
    ]
}
```

Beispiel 3: Um den Instance-Status für eine Patchgruppe mit weniger als zehn Instances abzurufen, für die ein Neustart erforderlich ist

Im folgenden `describe-instance-patch-states-for-patch-group`-Beispiel werden Details zum Status der Patch-Zusammenfassung für die angegebene Patchgruppe für Instances mit weniger als zehn Instances abgerufen, die einen Neustart erfordern.

```
aws ssm describe-instance-patch-states-for-patch-group \
    --filters Key=InstalledPendingRebootCount,Type=LessThan,Values=10 \
    --patch-group "Production"
```

Ausgabe:

```
{
    "InstancePatchStates": [
        {
            "InstanceId": "i-02573cafcfEXAMPLE",
            "BaselineId": "pb-0c10e65780EXAMPLE",
            "SnapshotId": "a3f5ff34-9bc4-4d2c-a665-4d1c1EXAMPLE",
            "PatchGroup": "Production",
            "OwnerInformation": "",
            "InstalledCount": 32,
            "InstalledOtherCount": 1,
            "InstalledPendingRebootCount": 4,
            "InstalledRejectedCount": 0,
            "MissingCount": 2,
            "FailedCount": 0,
            "UnreportedNotApplicableCount": 846,
            "NotApplicableCount": 212,
            "OperationStartTime": "2021-08-046T11:03:50.590000-07:00",
            "OperationEndTime": "2021-08-06T11:04:21.555000-07:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 0
        }
    ]
}
```

Weitere Informationen finden Sie unter Grundlegendes zu den Werten für den Patch-Kompatibilitätsstatus im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeInstancePatchStatesForPatchGroupin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden für eine Patch-Gruppe eine Übersicht über den Patch-Zustand auf der Ebene einzelner Instances abgerufen.

```
Get-SSMInstancePatchStatesForPatchGroup -PatchGroup "Production"
```

- Einzelheiten zur API finden Sie unter DescribeInstancePatchStatesForPatchGroup AWS - Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden für eine Patch-Gruppe eine Übersicht über den Patch-Zustand auf der Ebene einzelner Instances abgerufen.

```
Get-SSMInstancePatchStatesForPatchGroup -PatchGroup "Production"
```

- Einzelheiten zur API finden Sie unter DescribeInstancePatchStatesForPatchGroup AWS - Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeInstancePatches`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeInstancePatches` verwendet wird.

CLI

AWS CLI

Beispiel 1: Um die Details zum Patch-Status für eine Instance abzurufen

Das folgende `describe-instance-patches`-Beispiel ruft Details zu den Patches für die angegebene Instance ab.

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0"
```

Ausgabe:

```
{
    "Patches": [
        {
            "Title": "2019-01 Security Update for Adobe Flash Player for Windows
 Server 2016 for x64-based Systems (KB4480979)",
            "KBId": "KB4480979",
            "Classification": "SecurityUpdates",
            "Severity": "Critical",
            "State": "Installed",
            "InstalledTime": "2019-01-09T00:00:00+00:00"
        },
        {
            "Title": "",
            "KBId": "KB4481031",
            "Classification": "",
            "Severity": "",
            "State": "InstalledOther",
            "InstalledTime": "2019-02-08T00:00:00+00:00"
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

Beispiel 2: Um eine Liste mit Patches mit dem Status Fehlend für eine Instance abzurufen

Im folgenden `describe-instance-patches`-Beispiel werden Informationen über Patches abgerufen, die sich für die angegebene Instance im Status Missing befinden.

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0" \
    --filters Key=State,Values=Missing
```

Ausgabe:

```
{
    "Patches": [
        {
            "Title": "Windows Malicious Software Removal Tool x64 - February 2019
 (KB890830)",
            "KBId": "KB890830",
```

```
            "Classification": "UpdateRollups",
            "Severity": "Unspecified",
            "State": "Missing",
            "InstalledTime": "1970-01-01T00:00:00+00:00"
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

Weitere Informationen finden Sie unter Über Patch-Compliance-Status im AWS -Systems-Manager-Benutzerhandbuch.

Beispiel 3: Um eine Liste der Patches abzurufen, die seit einer bestimmten InstalledTime Instanz installiert wurden

Im folgenden Beispiel für `describe-instance-patches` werden Informationen über Patches abgerufen, die seit einem bestimmten Zeitpunkt für die angegebene Instance installiert wurden, indem die Verwendung von `--filters` und `--query` kombiniert wird.

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0" \
    --filters Key=State,Values=Installed \
    --query "Patches[?InstalledTime >= `2023-01-01T16:00:00`]"
```

Ausgabe:

```
{
    "Patches": [
        {
            "Title": "2023-03 Cumulative Update for Windows Server 2019 (1809)
 for x64-based Systems (KB5023702)",
            "KBId": "KB5023702",
            "Classification": "SecurityUpdates",
            "Severity": "Critical",
            "State": "Installed",
            "InstalledTime": "2023-03-16T11:00:00+00:00"
        },
        ...
    ],
    "NextToken": "--token string truncated--"
}
```

- Einzelheiten zur API finden Sie DescribeInstancePatchesin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden die Details zur Patch-Konformität für eine Instance abgerufen.

```
Get-SSMInstancePatch -InstanceId "i-08ee91c0b17045407"
```

- Einzelheiten zur API finden Sie unter DescribeInstancePatches AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden die Details zur Patch-Konformität für eine Instance abgerufen.

```
Get-SSMInstancePatch -InstanceId "i-08ee91c0b17045407"
```

- Einzelheiten zur API finden Sie unter DescribeInstancePatches AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeMaintenanceWindowExecutionTaskInvocations`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeMaintenanceWindowExecutionTaskInvocations` verwendet wird.

CLI

AWS CLI

So können Sie die spezifischen Aufgabenaufrufen für die Ausführung einer Aufgabe in einem Wartungsfenster ausführen

Im folgenden Beispiel `describe-maintenance-window-execution-task-invocations` werden die Aufrufe für die angegebene Aufgabe aufgeführt, die im Rahmen der Ausführung des angegebenen Wartungsfensters ausgeführt wurden.

```
aws ssm describe-maintenance-window-execution-task-invocations \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2a638355" \
    --task-id "ac0c6ae1-daa3-4a89-832e-d384503b6586"
```

Ausgabe:

```
{
    "WindowExecutionTaskInvocationIdentities": [
        {
            "Status": "SUCCESS",
            "Parameters": "{\"documentName\":\"AWS-RunShellScript\",
\"instanceIds\":[\"i-0000293ffd8c57862\"],\"parameters\":{\"commands\":[\"df\"]},
\"maxConcurrency\":\"1\",\"maxErrors\":\"1\"}",
            "InvocationId": "e274b6e1-fe56-4e32-bd2a-8073c6381d8b",
            "StartTime": 1487692834.723,
            "EndTime": 1487692834.871,
            "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2a638355",
            "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d384503b6586"
        }
    ]
}
```

Weitere Informationen finden Sie unter Informationen zu Aufgaben und Aufgabenausführungen (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeMaintenanceWindowExecutionTaskInvocationsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dieses Beispiel listet die Aufrufe einer Aufgabenausführungen als Teil einer Wartungsfenster-Ausführung auf.

```
Get-SSMMaintenanceWindowExecutionTaskInvocationList -TaskId "ac0c6ae1-
daa3-4a89-832e-d384503b6586" -WindowExecutionId "518d5565-5969-4cca-8f0e-
da3b2a638355"
```

Ausgabe:

```
EndTime           : 2/21/2017 4:00:34 PM
ExecutionId       :
InvocationId      : e274b6e1-fe56-4e32-bd2a-8073c6381d8b
OwnerInformation  :
Parameters        : {"documentName":"AWS-RunShellScript","instanceIds":
["i-0000293ffd8c57862"],"parameters":{"commands":["df"]},"maxConcurrency":"1",
                    "maxErrors":"1"}
StartTime         : 2/21/2017 4:00:34 PM
Status            : FAILED
StatusDetails     : The instance IDs list contains an invalid entry.
TaskExecutionId   : ac0c6ae1-daa3-4a89-832e-d384503b6586
WindowExecutionId : 518d5565-5969-4cca-8f0e-da3b2a638355
WindowTargetId    :
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowExecutionTaskInvocations AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dieses Beispiel listet die Aufrufe einer Aufgabenausführungen als Teil einer Wartungsfenster-Ausführung auf.

```
Get-SSMMaintenanceWindowExecutionTaskInvocationList -TaskId "ac0c6ae1-
daa3-4a89-832e-d384503b6586" -WindowExecutionId "518d5565-5969-4cca-8f0e-
da3b2a638355"
```

Ausgabe:

```
EndTime           : 2/21/2017 4:00:34 PM
ExecutionId       :
InvocationId      : e274b6e1-fe56-4e32-bd2a-8073c6381d8b
OwnerInformation  :
Parameters        : {"documentName":"AWS-RunShellScript","instanceIds":
["i-0000293ffd8c57862"],"parameters":{"commands":["df"]},"maxConcurrency":"1",
                    "maxErrors":"1"}
```

```
StartTime         : 2/21/2017 4:00:34 PM
Status            : FAILED
StatusDetails     : The instance IDs list contains an invalid entry.
TaskExecutionId   : ac0c6ae1-daa3-4a89-832e-d384503b6586
WindowExecutionId : 518d5565-5969-4cca-8f0e-da3b2a638355
WindowTargetId    :
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowExecutionTaskInvocations AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeMaintenanceWindowExecutionTasks`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeMaintenanceWindowExecutionTasks` verwendet wird.

CLI

AWS CLI

So können Sie alle Aufgaben auflisten, die mit der Ausführung eines Wartungsfensters verbunden sind

Im folgenden `ssm describe-maintenance-window-execution-tasks`-Beispiel werden die Aufgaben aufgeführt, die mit der Ausführung des angegebenen Wartungsfensters verknüpft sind.

```
aws ssm describe-maintenance-window-execution-tasks \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

Ausgabe:

```
{
    "WindowExecutionTaskIdentities": [
        {
            "Status": "SUCCESS",
            "TaskArn": "AWS-RunShellScript",
```

```
            "StartTime": 1487692834.684,
            "TaskType": "RUN_COMMAND",
            "EndTime": 1487692835.005,
            "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
            "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
        }
    ]
}
```

Weitere Informationen finden Sie unter Informationen zu Aufgaben und Aufgabenausführungen (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeMaintenanceWindowExecutionTasksin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dieses Beispiel listet die Aufgaben auf, die mit der Ausführung eines Wartungsfensters verbunden sind.

```
Get-SSMMaintenanceWindowExecutionTaskList -WindowExecutionId
 "518d5565-5969-4cca-8f0e-da3b2a638355"
```

Ausgabe:

```
EndTime           : 2/21/2017 4:00:35 PM
StartTime         : 2/21/2017 4:00:34 PM
Status            : SUCCESS
TaskArn           : AWS-RunShellScript
TaskExecutionId   : ac0c6ae1-daa3-4a89-832e-d384503b6586
TaskType          : RUN_COMMAND
WindowExecutionId : 518d5565-5969-4cca-8f0e-da3b2a638355
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowExecutionTasks AWS - Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dieses Beispiel listet die Aufgaben auf, die mit der Ausführung eines Wartungsfensters verbunden sind.

```
Get-SSMMaintenanceWindowExecutionTaskList -WindowExecutionId
 "518d5565-5969-4cca-8f0e-da3b2a638355"
```

Ausgabe:

```
EndTime           : 2/21/2017 4:00:35 PM
StartTime         : 2/21/2017 4:00:34 PM
Status            : SUCCESS
TaskArn           : AWS-RunShellScript
TaskExecutionId   : ac0c6ae1-daa3-4a89-832e-d384503b6586
TaskType          : RUN_COMMAND
WindowExecutionId : 518d5565-5969-4cca-8f0e-da3b2a638355
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowExecutionTasks AWS - Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeMaintenanceWindowExecutions`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeMaintenanceWindowExecutions` verwendet wird.

CLI

AWS CLI

Beispiel 1: So listen Sie alle Ausführungen für ein Wartungsfenster auf

Das folgende `describe-maintenance-window-executions`-Beispiel listet alle Ausführungen für das angegebene Wartungsfenster auf.

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE"
```

Ausgabe:

```
{
    "WindowExecutions": [
```

```
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowExecutionId": "6027b513-64fe-4cf0-be7d-1191aEXAMPLE",
            "Status": "IN_PROGRESS",
            "StartTime": "2021-08-04T11:00:00.000000-07:00"

        },
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowExecutionId": "ff75b750-4834-4377-8f61-b3cadEXAMPLE",
            "Status": "SUCCESS",
            "StartTime": "2021-08-03T11:00:00.000000-07:00",
            "EndTime": "2021-08-03T11:37:21.450000-07:00"
        },
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowExecutionId": "9fac7dd9-ff21-42a5-96ad-bbc4bEXAMPLE",
            "Status": "FAILED",
            "StatusDetails": "One or more tasks in the orchestration failed.",
            "StartTime": "2021-08-02T11:00:00.000000-07:00",
            "EndTime": "2021-08-02T11:22:36.190000-07:00"
        }
    ]
}
```

Beispiel 2: So listen Sie alle Ausführungen für ein Wartungsfenster vor einem bestimmten Datum auf

Im folgenden `describe-maintenance-window-executions`-Beispiel werden alle Ausführungen für das angegebene Wartungsfenster vor dem angegebenen Datum aufgeführt.

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=ExecutedBefore,Values=2021-08-03T00:00:00Z"
```

Ausgabe:

```
{
    "WindowExecutions": [
        {
        "WindowId": "mw-ab12cd34eEXAMPLE",
        "WindowExecutionId": "9fac7dd9-ff21-42a5-96ad-bbc4bEXAMPLE",
        "Status": "FAILED",
```

```
        "StatusDetails": "One or more tasks in the orchestration failed.",
        "StartTime": "2021-08-02T11:00:00.000000-07:00",
        "EndTime": "2021-08-02T11:22:36.190000-07:00"
    }
    ]
}
```

Beispiel 3: So listen Sie alle Ausführungen für ein Wartungsfenster nach einem bestimmten Datum auf

Im folgenden `describe-maintenance-window-executions`-Beispiel werden alle Ausführungen für das angegebene Wartungsfenster nach dem angegebenen Datum aufgeführt.

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=ExecutedAfter,Values=2021-08-04T00:00:00Z"
```

Ausgabe:

```
{
    "WindowExecutions": [
        {
        "WindowId": "mw-ab12cd34eEXAMPLE",
        "WindowExecutionId": "6027b513-64fe-4cf0-be7d-1191aEXAMPLE",
        "Status": "IN_PROGRESS",
        "StartTime": "2021-08-04T11:00:00.000000-07:00"
        }
    ]
}
```

Weitere Informationen finden Sie unter Informationen zu Aufgaben und Aufgabenausführungen (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeMaintenanceWindowExecutionsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden alle Ausführungen für ein Wartungsfenster aufgeführt.

```
Get-SSMMaintenanceWindowExecutionList -WindowId "mw-03eb9db42890fb82d"
```

Ausgabe:

```
EndTime           : 2/20/2017 6:30:17 PM
StartTime         : 2/20/2017 6:30:16 PM
Status            : FAILED
StatusDetails     : One or more tasks in the orchestration failed.
WindowExecutionId : 6f3215cf-4101-4fa0-9b7b-9523269599c7
WindowId          : mw-03eb9db42890fb82d
```

Beispiel 2: In diesem Beispiel werden alle Ausführungen für ein Wartungsfenster vor einem bestimmten Datum aufgeführt.

```
$option1 = @{Key="ExecutedBefore";Values=@("2016-11-04T05:00:00Z")}
Get-SSMMaintenanceWindowExecutionList -WindowId "mw-03eb9db42890fb82d" -Filter
 $option1
```

Beispiel 3: In diesem Beispiel werden alle Ausführungen für ein Wartungsfenster nach einem bestimmten Datum aufgeführt.

```
$option1 = @{Key="ExecutedAfter";Values=@("2016-11-04T05:00:00Z")}
Get-SSMMaintenanceWindowExecutionList -WindowId "mw-03eb9db42890fb82d" -Filter
 $option1
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowExecutions AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden alle Ausführungen für ein Wartungsfenster aufgeführt.

```
Get-SSMMaintenanceWindowExecutionList -WindowId "mw-03eb9db42890fb82d"
```

Ausgabe:

```
EndTime           : 2/20/2017 6:30:17 PM
StartTime         : 2/20/2017 6:30:16 PM
Status            : FAILED
StatusDetails     : One or more tasks in the orchestration failed.
```

```
WindowExecutionId : 6f3215cf-4101-4fa0-9b7b-9523269599c7
WindowId          : mw-03eb9db42890fb82d
```

Beispiel 2: In diesem Beispiel werden alle Ausführungen für ein Wartungsfenster vor einem bestimmten Datum aufgeführt.

```
$option1 = @{Key="ExecutedBefore";Values=@("2016-11-04T05:00:00Z")}
Get-SSMMaintenanceWindowExecutionList -WindowId "mw-03eb9db42890fb82d" -Filter
 $option1
```

Beispiel 3: In diesem Beispiel werden alle Ausführungen für ein Wartungsfenster nach einem bestimmten Datum aufgeführt.

```
$option1 = @{Key="ExecutedAfter";Values=@("2016-11-04T05:00:00Z")}
Get-SSMMaintenanceWindowExecutionList -WindowId "mw-03eb9db42890fb82d" -Filter
 $option1
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowExecutions AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeMaintenanceWindowTargets`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeMaintenanceWindowTargets` verwendet wird.

CLI

AWS CLI

Beispiel 1: So listen Sie alle Ziele für ein Wartungsfenster auf

Das folgende `describe-maintenance-window-targets`-Beispiel listet alle Ziele für das angegebene Wartungsfenster auf.

```
aws ssm describe-maintenance-window-targets \
```

```
    --window-id "mw-06cf17cbefEXAMPLE"
```

Ausgabe:

```
{
    "Targets": [
        {
            "ResourceType": "INSTANCE",
            "OwnerInformation": "Single instance",
            "WindowId": "mw-06cf17cbefEXAMPLE",
            "Targets": [
                {
                    "Values": [
                        "i-0000293ffdEXAMPLE"
                    ],
                    "Key": "InstanceIds"
                }
            ],
            "WindowTargetId": "350d44e6-28cc-44e2-951f-4b2c9EXAMPLE"
        },
        {
            "ResourceType": "INSTANCE",
            "OwnerInformation": "Two instances in a list",
            "WindowId": "mw-06cf17cbefEXAMPLE",
            "Targets": [
                {
                    "Values": [
                        "i-0000293ffdEXAMPLE",
                        "i-0cb2b964d3EXAMPLE"
                    ],
                    "Key": "InstanceIds"
                }
            ],
            "WindowTargetId": "e078a987-2866-47be-bedd-d9cf4EXAMPLE"
        }
    ]
}
```

Beispiel 2: So listen Sie alle Ziele für ein Wartungsfenster auf, die einem bestimmten Eigentümerinformationswert entsprechen

In diesem `describe-maintenance-window-targets`-Beispiel werden alle Ziele für ein Wartungsfenster aufgeführt.

```
aws ssm describe-maintenance-window-targets \
    --window-id "mw-0ecb1226ddEXAMPLE" \
    --filters "Key=OwnerInformation,Values=CostCenter1"
```

Ausgabe:

```
{
    "Targets": [
        {
            "WindowId": "mw-0ecb1226ddEXAMPLE",
            "WindowTargetId": "da89dcc3-7f9c-481d-ba2b-edcb7d0057f9",
            "ResourceType": "INSTANCE",
            "Targets": [
                {
                    "Key": "tag:Environment",
                    "Values": [
                        "Prod"
                    ]
                }
            ],
            "OwnerInformation": "CostCenter1",
            "Name": "ProdTarget1"
        }
    ]
}
```

Weitere Informationen finden Sie unter Informationen über Maintenance Windows (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeMaintenanceWindowTargetsunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden alle Ziele für ein Wartungsfenster aufgeführt.

```
Get-SSMMaintenanceWindowTarget -WindowId "mw-06cf17cbefcb4bf4f"
```

Ausgabe:

```
OwnerInformation : Single instance
ResourceType     : INSTANCE
Targets          : {InstanceIds}
WindowId         : mw-06cf17cbefcb4bf4f
WindowTargetId   : 350d44e6-28cc-44e2-951f-4b2c985838f6

OwnerInformation : Two instances in a list
ResourceType     : INSTANCE
Targets          : {InstanceIds}
WindowId         : mw-06cf17cbefcb4bf4f
WindowTargetId   : e078a987-2866-47be-bedd-d9cf49177d3a
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowTargets AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden alle Ziele für ein Wartungsfenster aufgeführt.

```
Get-SSMMaintenanceWindowTarget -WindowId "mw-06cf17cbefcb4bf4f"
```

Ausgabe:

```
OwnerInformation : Single instance
ResourceType     : INSTANCE
Targets          : {InstanceIds}
WindowId         : mw-06cf17cbefcb4bf4f
WindowTargetId   : 350d44e6-28cc-44e2-951f-4b2c985838f6

OwnerInformation : Two instances in a list
ResourceType     : INSTANCE
Targets          : {InstanceIds}
WindowId         : mw-06cf17cbefcb4bf4f
WindowTargetId   : e078a987-2866-47be-bedd-d9cf49177d3a
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowTargets AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeMaintenanceWindowTasks`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeMaintenanceWindowTasks` verwendet wird.

CLI

AWS CLI

Beispiel 1: So listen Sie alle Aufgaben für ein Wartungsfenster auf

Das folgende `describe-maintenance-window-tasks`-Beispiel listet alle Aufgaben für das angegebene Wartungsfenster auf.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-06cf17cbefEXAMPLE"
```

Ausgabe:

```
{
    "Tasks": [
        {
            "WindowId": "mw-06cf17cbefEXAMPLE",
            "WindowTaskId": "018b31c3-2d77-4b9e-bd48-c91edEXAMPLE",
            "TaskArn": "AWS-RestartEC2Instance",
            "TaskParameters": {},
            "Type": "AUTOMATION",
            "Description": "Restarting EC2 Instance for maintenance",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "My-Automation-Example-Task",
            "Priority": 0,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ]
        },
        {
```

```
            "WindowId": "mw-06cf17cbefEXAMPLE",
            "WindowTaskId": "1943dee0-0a17-4978-9bf4-3cc2fEXAMPLE",
            "TaskArn": "AWS-DisableS3BucketPublicReadWrite",
            "TaskParameters": {},
            "Type": "AUTOMATION",
            "Description": "Automation task to disable read/write access on
 public S3 buckets",
            "MaxConcurrency": "10",
            "MaxErrors": "5",
            "Name": "My-Disable-S3-Public-Read-Write-Access-Automation-Task",
            "Priority": 0,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ]
        }
    ]
}
```

Beispiel 2: Um alle Aufgaben für ein Wartungsfenster aufzulisten, das das RunPowerShellScript Befehlsdokument AWS- aufruft

Das folgende `describe-maintenance-window-tasks`-Beispiel listet alle Aufgaben für das angegebene `AWS-RunPowerShellScript`-Wartungsfenster auf.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=TaskArn,Values=AWS-RunPowerShellScript"
```

Ausgabe:

```
{
    "Tasks": [
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",
            "TaskArn": "AWS-RunPowerShellScript",
```

```
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 1,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "MyTask"
        }
    ]
}
```

Beispiel 3: So listen Sie alle Aufgaben für ein Wartungsfenster auf, die eine Priorität von 3 haben

Das folgende `describe-maintenance-window-tasks`-Beispiel listet alle Aufgaben für das angegebene Wartungsfenster auf, die eine `Priority` von 3 haben.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=Priority,Values=3"
```

Ausgabe:

```
{
    "Tasks": [
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",
            "TaskArn": "AWS-RunPowerShellScript",
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
```

```
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 3,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "MyRunCommandTask"
        },
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowTaskId": "ee45feff-ad65-4a6c-b478-5cab8EXAMPLE",
            "TaskArn": "AWS-RestartEC2Instance",
            "Type": "AUTOMATION",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 3,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "MaxConcurrency": "10",
            "MaxErrors": "5",
            "Name": "My-Automation-Task",
            "Description": "A description for my Automation task"
        }
    ]
}
```

Beispiel 4: So listen Sie alle Aufgaben für ein Wartungsfenster auf, die eine Priorität von 1 haben, und verwenden Sie Run Command

In diesem `describe-maintenance-window-tasks`-Beispiel werden alle Aufgaben für das angegebene Wartungsfenster aufgeführt, die einen `Priority` von 1 und `Run Command` nutzen.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=Priority,Values=1" "Key=TaskType,Values=RUN_COMMAND"
```

Ausgabe:

```
{
    "Tasks": [
        {
            "WindowId": "mw-ab12cd34eEXAMPLE",
            "WindowTaskId": "0d36e6b4-3a4f-411e-adcb-3558eEXAMPLE",
            "TaskArn": "AWS-RunPowerShellScript",
            "Type": "RUN_COMMAND",
            "Targets": [
                {
                    "Key": "WindowTargetIds",
                    "Values": [
                        "da89dcc3-7f9c-481d-ba2b-edcb7EXAMPLE"
                    ]
                }
            ],
            "TaskParameters": {},
            "Priority": 1,
            "ServiceRoleArn": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "MaxConcurrency": "1",
            "MaxErrors": "1",
            "Name": "MyRunCommandTask"
        }
    ]
}
```

Weitere Informationen finden Sie unter Informationen zu Wartungsfenstern (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeMaintenanceWindowTasksunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden alle Aufgaben für ein Wartungsfenster aufgeführt.

```
Get-SSMMaintenanceWindowTaskList -WindowId "mw-06cf17cbefcb4bf4f"
```

Ausgabe:

```
LoggingInfo    :
MaxConcurrency : 1
MaxErrors      : 1
Priority       : 10
ServiceRoleArn : arn:aws:iam::123456789012:role/MaintenanceWindowsRole
Targets        : {InstanceIds}
TaskArn        : AWS-RunShellScript
TaskParameters : {[commands,
 Amazon.SimpleSystemsManagement.Model.MaintenanceWindowTaskParameterValueExpression]}
Type           : RUN_COMMAND
WindowId       : mw-06cf17cbefcb4bf4f
WindowTaskId   : a23e338d-ff30-4398-8aa3-09cd052ebf17
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowTasks AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden alle Aufgaben für ein Wartungsfenster aufgeführt.

```
Get-SSMMaintenanceWindowTaskList -WindowId "mw-06cf17cbefcb4bf4f"
```

Ausgabe:

```
LoggingInfo    :
MaxConcurrency : 1
MaxErrors      : 1
Priority       : 10
ServiceRoleArn : arn:aws:iam::123456789012:role/MaintenanceWindowsRole
Targets        : {InstanceIds}
TaskArn        : AWS-RunShellScript
TaskParameters : {[commands,
 Amazon.SimpleSystemsManagement.Model.MaintenanceWindowTaskParameterValueExpression]}
```

```
Type            : RUN_COMMAND
WindowId        : mw-06cf17cbefcb4bf4f
WindowTaskId    : a23e338d-ff30-4398-8aa3-09cd052ebf17
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindowTasks AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribeMaintenanceWindows`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeMaintenanceWindows` verwendet wird.

CLI

AWS CLI

Beispiel 1: So listen Sie alle Wartungsfenster auf

Das folgende `describe-maintenance-windows` Beispiel listet alle Wartungsfenster in Ihrem AWS Konto in der aktuellen Region auf.

```
aws ssm describe-maintenance-windows
```

Ausgabe:

```
{
    "WindowIdentities": [
        {
            "WindowId": "mw-0ecb1226ddEXAMPLE",
            "Name": "MyMaintenanceWindow-1",
            "Enabled": true,
            "Duration": 2,
            "Cutoff": 1,
            "Schedule": "rate(180 minutes)",
            "NextExecutionTime": "2020-02-12T23:19:20.596Z"
        },
        {
            "WindowId": "mw-03eb9db428EXAMPLE",
```

```
            "Name": "MyMaintenanceWindow-2",
            "Enabled": true,
            "Duration": 3,
            "Cutoff": 1,
            "Schedule": "rate(7 days)",
            "NextExecutionTime": "2020-02-17T23:22:00.956Z"
        },
    ]
}
```

Beispiel 2: So listen Sie alle aktivierten Wartungsfenster auf

Das folgende `describe-maintenance-windows`-Beispiel listet alle aktivierten Wartungsfenster auf.

```
aws ssm describe-maintenance-windows \
    --filters "Key=Enabled,Values=true"
```

Beispiel 3: So listen Sie Wartungsfenster auf, die einem bestimmten Namen entsprechen

In diesem `describe-maintenance-windows`-Beispiel werden alle Wartungsfenster mit dem angegebenen Namen aufgeführt.

```
aws ssm describe-maintenance-windows \
    --filters "Key=Name,Values=MyMaintenanceWindow"
```

Weitere Informationen finden Sie unter Informationen über Maintenance Windows (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeMaintenanceWindowsunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden alle Wartungsfenster in Ihrem Konto aufgeführt.

```
Get-SSMMaintenanceWindowList
```

Ausgabe:

```
Cutoff   : 1
Duration : 4
Enabled  : True
Name     : My-First-Maintenance-Window
WindowId : mw-06d59c1a07c022145
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindows AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden alle Wartungsfenster in Ihrem Konto aufgeführt.

```
Get-SSMMaintenanceWindowList
```

Ausgabe:

```
Cutoff   : 1
Duration : 4
Enabled  : True
Name     : My-First-Maintenance-Window
WindowId : mw-06d59c1a07c022145
```

- Einzelheiten zur API finden Sie unter DescribeMaintenanceWindows AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeOpsItems`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeOpsItems` verwendet wird.

CLI

AWS CLI

Um eine Reihe von aufzulisten OpsItems

Im folgenden `describe-ops-items` Beispiel wird eine Liste aller offenen Konten OpsItems in Ihrem AWS Konto angezeigt.

```
aws ssm describe-ops-items \
    --ops-item-filters "Key=Status,Values=Open,Operator=Equal"
```

Ausgabe:

```
{
    "OpsItemSummaries": [
        {
            "CreatedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-
Role/fbf77cbe264a33509569f23e4EXAMPLE",
            "CreatedTime": "2020-03-14T17:02:46.375000-07:00",
            "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-
CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
            "LastModifiedTime": "2020-03-14T17:02:46.375000-07:00",
            "Source": "SSM",
            "Status": "Open",
            "OpsItemId": "oi-7cfc5EXAMPLE",
            "Title": "SSM Maintenance Window execution failed",
            "OperationalData": {
                "/aws/dedup": {
                    "Value": "{\"dedupString\":\"SSMOpsItems-SSM-maintenance-
window-execution-failed\"}",
                    "Type": "SearchableString"
                },
                "/aws/resources": {
                    "Value": "[{\"arn\":\"arn:aws:ssm:us-
east-2:111222333444:maintenancewindow/mw-034093d322EXAMPLE\"}]",
                    "Type": "SearchableString"
                }
            },
            "Category": "Availability",
            "Severity": "3"
        },
        {
            "CreatedBy": "arn:aws:sts::1112223233444:assumed-role/OpsItem-CWE-
Role/fbf77cbe264a33509569f23e4EXAMPLE",
            "CreatedTime": "2020-02-26T11:43:15.426000-08:00",
            "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-
CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
            "LastModifiedTime": "2020-02-26T11:43:15.426000-08:00",
            "Source": "EC2",
            "Status": "Open",
            "OpsItemId": "oi-6f966EXAMPLE",
```

```
            "Title": "EC2 instance stopped",
            "OperationalData": {
                "/aws/automations": {
                    "Value": "[ { \"automationType\": \"AWS:SSM:Automation\",
 \"automationId\": \"AWS-RestartEC2Instance\" } ]",
                    "Type": "SearchableString"
                },
                "/aws/dedup": {
                    "Value": "{\"dedupString\":\"SSMOpsItems-EC2-instance-stopped
\"}",
                    "Type": "SearchableString"
                },
                "/aws/resources": {
                    "Value": "[{\"arn\":\"arn:aws:ec2:us-
east-2:111222333444:instance/i-0beccfbc02EXAMPLE\"}]",
                    "Type": "SearchableString"
                }
            },
            "Category": "Availability",
            "Severity": "3"
        }
    ]
}
```

Weitere Informationen finden Sie unter Arbeiten mit OpsItems im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribeOpsItemsin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    /**
     * Describes AWS SSM OpsItems asynchronously.
     *
```

```
    * @param key The key to filter OpsItems by (e.g., OPS_ITEM_ID).
    *
    * This method initiates an asynchronous request to describe SSM OpsItems.
    * If the request is successful, it prints the title and status of each
OpsItem.
    * If an exception occurs, it handles the error appropriately.
    */
   public void describeOpsItems(String key) {
       OpsItemFilter filter = OpsItemFilter.builder()
               .key(OpsItemFilterKey.OPS_ITEM_ID)
               .values(key)
               .operator(OpsItemFilterOperator.EQUAL)
               .build();

       DescribeOpsItemsRequest itemsRequest = DescribeOpsItemsRequest.builder()
               .maxResults(10)
               .opsItemFilters(filter)
               .build();

       CompletableFuture<Void> future = CompletableFuture.runAsync(() -> {
           getAsyncClient().describeOpsItems(itemsRequest)
                   .thenAccept(itemsResponse -> {
                       List<OpsItemSummary> items =
itemsResponse.opsItemSummaries();
                       for (OpsItemSummary item : items) {
                           System.out.println("The item title is " +
item.title() + " and the status is " + item.status().toString());
                       }
                   })
                   .exceptionally(ex -> {
                       throw new CompletionException(ex);
                   }).join();
       }).exceptionally(ex -> {
           Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
           if (cause instanceof SsmException) {
               throw new RuntimeException("SSM error: " + cause.getMessage(),
cause);
           } else {
               throw new RuntimeException("Unexpected error: " +
cause.getMessage(), cause);
           }
       });
```

```
        try {
            future.join();
        } catch (CompletionException ex) {
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
 ex.getCause() : ex;
        }
    }
```

- Einzelheiten zur API finden Sie DescribeOpsItemsin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import {
  OpsItemFilterOperator,
  OpsItemFilterKey,
  paginateDescribeOpsItems,
  SSMClient,
} from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * Describe SSM OpsItems.
 * @param {{ opsItemId: string }}
 */
export const main = async ({ opsItemId }) => {
  const client = new SSMClient({});
  try {
    const describeOpsItemsPaginated = [];
    for await (const page of paginateDescribeOpsItems(
      { client },
      {
```

```
        OpsItemFilters: {
          Key: OpsItemFilterKey.OPSITEM_ID,
          Operator: OpsItemFilterOperator.EQUAL,
          Values: opsItemId,
        },
      },
    )) {
      describeOpsItemsPaginated.push(...page.OpsItemSummaries);
    }
    console.log("Here are the ops items:");
    console.log(describeOpsItemsPaginated);
    return { OpsItemSummaries: describeOpsItemsPaginated };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "MissingParameter") {
      console.warn(`${caught.message}. Did you provide this value?`);
    }
    throw caught;
  }
};
```

- Einzelheiten zur API finden Sie DescribeOpsItemsin der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class OpsItemWrapper:
    """Encapsulates AWS Systems Manager OpsItem actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
```

```
        self.ssm_client = ssm_client
        self.id = None

    @classmethod
    def from_client(cls):
        """
        :return: A OpsItemWrapper instance.
        """
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def describe(self):
        """
        Describe an OpsItem.
        """
        try:
            paginator = self.ssm_client.get_paginator("describe_ops_items")
            ops_items = []
            for page in paginator.paginate(
                OpsItemFilters=[
                    {"Key": "OpsItemId", "Values": [self.id], "Operator":
"Equal"}
                ]
            ):
                ops_items.extend(page["OpsItemSummaries"])

            for item in ops_items:
                print(
                    f"The item title is {item['Title']} and the status is
{item['Status']}"
                )
            return len(ops_items) > 0
        except ClientError as err:
            logger.error(
                "Couldn't describe ops item %s. Here's why: %s: %s",
                self.id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Einzelheiten zur API finden Sie DescribeOpsItemsin AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeParameters`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeParameters` verwendet wird.

CLI

AWS CLI

Beispiel 1: So können Sie alle Parameter auflisten

Das folgende `describe-parameters` Beispiel listet alle Parameter im aktuellen AWS Konto und in der Region auf.

```
aws ssm describe-parameters
```

Ausgabe:

```
{
    "Parameters": [
        {
            "Name": "MySecureStringParameter",
            "Type": "SecureString",
            "KeyId": "alias/aws/ssm",
            "LastModifiedDate": 1582155479.205,
            "LastModifiedUser": "arn:aws:sts::111222333444:assumed-role/Admin/
Richard-Roe-Managed",
            "Description": "This is a SecureString parameter",
            "Version": 2,
            "Tier": "Advanced",
            "Policies": [
                {
                    "PolicyText": "{\"Type\":\"Expiration\",\"Version\":\"1.0\",
\"Attributes\":{\"Timestamp\":\"2020-07-07T22:30:00Z\"}}",
                    "PolicyType": "Expiration",
                    "PolicyStatus": "Pending"
```

```
                },
                {
                    "PolicyText": "{\"Type\":\"ExpirationNotification\",\"Version
\":\"1.0\",\"Attributes\":{\"Before\":\"12\",\"Unit\":\"Hours\"}}",
                    "PolicyType": "ExpirationNotification",
                    "PolicyStatus": "Pending"
                }
            ]
        },
        {
            "Name": "MyStringListParameter",
            "Type": "StringList",
            "LastModifiedDate": 1582154764.222,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is a StringList parameter",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "LastModifiedDate": 1582154711.976,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Alejandro-
Rosalez",
            "Description": "This is a String parameter",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "latestAmi",
            "Type": "String",
            "LastModifiedDate": 1580862415.521,
            "LastModifiedUser": "arn:aws:sts::111222333444:assumed-role/lambda-
ssm-role/Automation-UpdateSSM-Param",
            "Version": 3,
            "Tier": "Standard",
            "Policies": []
        }
    ]
}
```

Beispiel 2: Um alle Parameter aufzulisten, die bestimmten Metadaten entsprechen

In diesem `describe-parameters`-Beispiel werden alle Parameter aufgeführt, die einem Filter entsprechen.

aws ssm describe-parameters --filters „Key=Type, Values=“ StringList

Ausgabe:

```
{
    "Parameters": [
        {
            "Name": "MyStringListParameter",
            "Type": "StringList",
            "LastModifiedDate": 1582154764.222,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is a StringList parameter",
            "Version": 1,
            "Tier": "Standard",
            "Policies": []
        }
    ]
}
```

Weitere Informationen finden Sie unter Suchen nach Systems-Manager-Parametern im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie in der Befehlsreferenz. DescribeParametersAWS CLI

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.ssm.SsmClient;
import software.amazon.awssdk.services.ssm.model.GetParameterRequest;
```

```
import software.amazon.awssdk.services.ssm.model.GetParameterResponse;
import software.amazon.awssdk.services.ssm.model.SsmException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class GetParameter {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <paraName>

                Where:
                    paraName - The name of the parameter.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String paraName = args[0];
        Region region = Region.US_EAST_1;
        SsmClient ssmClient = SsmClient.builder()
                .region(region)
                .build();

        getParaValue(ssmClient, paraName);
        ssmClient.close();
    }

    public static void getParaValue(SsmClient ssmClient, String paraName) {
        try {
            GetParameterRequest parameterRequest = GetParameterRequest.builder()
                    .name(paraName)
                    .build();
```

```
            GetParameterResponse parameterResponse =
 ssmClient.getParameter(parameterRequest);
            System.out.println("The parameter value is " +
 parameterResponse.parameter().value());

        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie DescribeParametersin der AWS SDK for Java 2.x API-Referenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dieses Beispiel listet alle Parameter auf.

```
Get-SSMParameterList
```

Ausgabe:

```
Description      :
KeyId            :
LastModifiedDate : 3/3/2017 6:58:23 PM
LastModifiedUser : arn:aws:iam::123456789012:user/admin
Name             : Welcome
Type             : String
```

- Einzelheiten zur API finden Sie unter DescribeParameters AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dieses Beispiel listet alle Parameter auf.

```
Get-SSMParameterList
```

Ausgabe:

```
Description      :
KeyId            :
LastModifiedDate : 3/3/2017 6:58:23 PM
LastModifiedUser : arn:aws:iam::123456789012:user/admin
Name             : Welcome
Type             : String
```

- Einzelheiten zur API finden Sie unter DescribeParameters AWS -Tools für PowerShellCmdlet-Referenz (V5).

Rust

SDK für Rust

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
async fn show_parameters(client: &Client) -> Result<(), Error> {
    let resp = client.describe_parameters().send().await?;

    for param in resp.parameters() {
        println!("  {}", param.name().unwrap_or_default());
    }

    Ok(())
}
```

- Einzelheiten zur API finden Sie DescribeParametersin der API-Referenz zum AWS SDK für Rust.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden dieses Dienstes mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribePatchBaselines`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribePatchBaselines` verwendet wird.

CLI

AWS CLI

Beispiel 1: So können Sie alle Patch-Baselines auflisten

Im folgenden Beispiel `describe-patch-baselines` werden Details für alle Patch-Baselines in Ihrem Konto in der aktuellen Region abgerufen.

```
aws ssm describe-patch-baselines
```

Ausgabe:

```
{
    "BaselineIdentities": [
        {
            "BaselineName": "AWS-SuseDefaultPatchBaseline",
            "DefaultBaseline": true,
            "BaselineDescription": "Default Patch Baseline for Suse Provided by
 AWS.",
            "BaselineId": "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-0123fdb36e334a3b2",
            "OperatingSystem": "SUSE"
        },
        {
            "BaselineName": "AWS-DefaultPatchBaseline",
            "DefaultBaseline": false,
            "BaselineDescription": "Default Patch Baseline Provided by AWS.",
            "BaselineId": "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-020d361a05defe4ed",
            "OperatingSystem": "WINDOWS"
        },
        ...
        {
            "BaselineName": "MyWindowsPatchBaseline",
            "DefaultBaseline": true,
            "BaselineDescription": "My patch baseline for EC2 instances for
 Windows Server",
            "BaselineId": "pb-0ad00e0dd7EXAMPLE",
```

```
            "OperatingSystem": "WINDOWS"
        }
    ]
}
```

Beispiel 2: Um alle Patch-Baselines aufzulisten, die bereitgestellt werden von AWS

Das folgende `describe-patch-baselines` Beispiel listet alle Patch-Baselines auf, die von bereitgestellt werden. AWS

```
aws ssm describe-patch-baselines \
    --filters "Key=OWNER,Values=[AWS]"
```

Beispiel 3: So können Sie alle Patch-Baselines auflisten, die Ihnen gehören

Im folgenden `describe-patch-baselines`-Beispiel werden alle benutzerdefinierten Patch-Baselines aufgeführt, die in Ihrem Konto in der aktuellen Region erstellt wurden.

```
aws ssm describe-patch-baselines \
    --filters "Key=OWNER,Values=[Self]"
```

Weitere Informationen finden Sie unter Über vordefinierte und benutzerdefinierte Patch-Baselines im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribePatchBaselinesin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dieses Beispiel listet alle Patch-Baselines auf.

```
Get-SSMPatchBaseline
```

Ausgabe:

```
BaselineDescription                                             BaselineId
                                                  BaselineName
-------------------                                             ----------
                                                  ------------
Default Patch Baseline Provided by AWS.                         arn:aws:ssm:us-
west-2:123456789012:patchbaseline/pb-04fb4ae6142167966 AWS-DefaultP...
```

```
Baseline containing all updates approved for production systems
 pb-045f10b4f382baeda
 Production-B...
Baseline containing all updates approved for production systems
 pb-0a2f1059b670ebd31
 Production-B...
```

Beispiel 2: Dieses Beispiel listet alle Patch-Baselines auf, die von bereitgestellt werden. AWS Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$filter1 = @{Key="OWNER";Values=@("AWS")}
```

Ausgabe:

```
Get-SSMPatchBaseline -Filter $filter1
```

Beispiel 3: In diesem Beispiel werden alle Patch-Baselines mit Ihnen als Eigentümer aufgeführt. Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$filter1 = @{Key="OWNER";Values=@("Self")}
```

Ausgabe:

```
Get-SSMPatchBaseline -Filter $filter1
```

Beispiel 4: Bei PowerShell Version 2 müssen Sie New-Object verwenden, um jedes Tag zu erstellen.

```
$filter1 = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchOrchestratorFilter
$filter1.Key = "OWNER"
$filter1.Values = "AWS"

Get-SSMPatchBaseline -Filter $filter1
```

Ausgabe:

```
BaselineDescription                    BaselineId
                            BaselineName             DefaultBaselin
```

```
                                                    e
-------------------                    ----------
                              ------------            --------------
Default Patch Baseline Provided by AWS. arn:aws:ssm:us-
west-2:123456789012:patchbaseline/pb-04fb4ae6142167966 AWS-DefaultPatchBaseline
 True
```

- Einzelheiten zur API finden Sie unter DescribePatchBaselines AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dieses Beispiel listet alle Patch-Baselines auf.

```
Get-SSMPatchBaseline
```

Ausgabe:

```
BaselineDescription                                             BaselineId
                                                  BaselineName
-------------------                                             ----------
                                                  ------------
Default Patch Baseline Provided by AWS.                         arn:aws:ssm:us-
west-2:123456789012:patchbaseline/pb-04fb4ae6142167966 AWS-DefaultP...
Baseline containing all updates approved for production systems
 pb-045f10b4f382baeda
 Production-B...
Baseline containing all updates approved for production systems
 pb-0a2f1059b670ebd31
 Production-B...
```

Beispiel 2: Dieses Beispiel listet alle Patch-Baselines auf, die von bereitgestellt werden. AWS Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$filter1 = @{Key="OWNER";Values=@("AWS")}
```

Ausgabe:

```
Get-SSMPatchBaseline -Filter $filter1
```

Beispiel 3: In diesem Beispiel werden alle Patch-Baselines mit Ihnen als Eigentümer aufgeführt. Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$filter1 = @{Key="OWNER";Values=@("Self")}
```

Ausgabe:

```
Get-SSMPatchBaseline -Filter $filter1
```

Beispiel 4: Bei PowerShell Version 2 müssen Sie New-Object verwenden, um jedes Tag zu erstellen.

```
$filter1 = New-Object
 Amazon.SimpleSystemsManagement.Model.PatchOrchestratorFilter
$filter1.Key = "OWNER"
$filter1.Values = "AWS"

Get-SSMPatchBaseline -Filter $filter1
```

Ausgabe:

```
BaselineDescription                     BaselineId
                            BaselineName            DefaultBaselin


                                                    e
------------------                      ----------
                            ------------            --------------
Default Patch Baseline Provided by AWS. arn:aws:ssm:us-
west-2:123456789012:patchbaseline/pb-04fb4ae6142167966 AWS-DefaultPatchBaseline
 True
```

- Einzelheiten zur API finden Sie unter DescribePatchBaselines AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribePatchGroupState`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribePatchGroupState` verwendet wird.

CLI

AWS CLI

So rufen Sie den Status einer Patchgruppe ab

Im folgenden Beispiel `describe-patch-group-state` wird die allgemeine Zusammenfassung der Patch-Konformität für eine Patchgruppe abgerufen.

```
aws ssm describe-patch-group-state \
    --patch-group "Production"
```

Ausgabe:

```
{
    "Instances": 21,
    "InstancesWithCriticalNonCompliantPatches": 1,
    "InstancesWithFailedPatches": 2,
    "InstancesWithInstalledOtherPatches": 3,
    "InstancesWithInstalledPatches": 21,
    "InstancesWithInstalledPendingRebootPatches": 2,
    "InstancesWithInstalledRejectedPatches": 1,
    "InstancesWithMissingPatches": 3,
    "InstancesWithNotApplicablePatches": 4,
    "InstancesWithOtherNonCompliantPatches": 1,
    "InstancesWithSecurityNonCompliantPatches": 1,
    "InstancesWithUnreportedNotApplicablePatches": 2
}
```

Weitere Informationen finden Sie unter About patch groups < https://docs.aws.amazon.com/systems-manager/ latest/userguide/sysman -patch-patchgroups.html>__ und Understanding Patch Compliance State Values im Systems Manager Manager-Benutzerhandbuch.AWS

- Einzelheiten zur API finden Sie in der Befehlsreferenz. DescribePatchGroupStateAWS CLI

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird die allgemeine Zusammenfassung der Patch-Konformität für eine Patch-Gruppe abgerufen.

```
Get-SSMPatchGroupState -PatchGroup "Production"
```

Ausgabe:

```
Instances                           : 4
InstancesWithFailedPatches          : 1
InstancesWithInstalledOtherPatches : 4
InstancesWithInstalledPatches       : 3
InstancesWithMissingPatches         : 0
InstancesWithNotApplicablePatches   : 0
```

- Einzelheiten zur API finden Sie unter DescribePatchGroupState AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird die allgemeine Zusammenfassung der Patch-Konformität für eine Patch-Gruppe abgerufen.

```
Get-SSMPatchGroupState -PatchGroup "Production"
```

Ausgabe:

```
Instances                           : 4
InstancesWithFailedPatches          : 1
InstancesWithInstalledOtherPatches : 4
InstancesWithInstalledPatches       : 3
InstancesWithMissingPatches         : 0
InstancesWithNotApplicablePatches   : 0
```

- Einzelheiten zur API finden Sie unter DescribePatchGroupState AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`DescribePatchGroups`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `DescribePatchGroups` verwendet wird.

CLI

AWS CLI

So zeigen Sie Registrierungen für Patch-Gruppen an

Das folgende `describe-patch-groups`-Beispiel listet die Patch-Gruppenregistrierungen auf.

```
aws ssm describe-patch-groups
```

Ausgabe:

```
{
    "Mappings": [
        {
            "PatchGroup": "Production",
            "BaselineIdentity": {
                "BaselineId": "pb-0123456789abcdef0",
                "BaselineName": "ProdPatching",
                "OperatingSystem": "WINDOWS",
                "BaselineDescription": "Patches for Production",
                "DefaultBaseline": false
            }
        },
        {
            "PatchGroup": "Development",
            "BaselineIdentity": {
                "BaselineId": "pb-0713accee01234567",
                "BaselineName": "DevPatching",
                "OperatingSystem": "WINDOWS",
                "BaselineDescription": "Patches for Development",
                "DefaultBaseline": true
            }
```

```
        },
        ...
    ]
}
```

Weitere Informationen finden Sie unter Erstellen einer Patchgruppe < https://docs.aws.amazon.com/systems-manager/ latest/userguide/sysman - patch-group-tagging .html>__ und Hinzufügen einer Patchgruppe zu einer Patch-Baseline im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie DescribePatchGroupsin AWS CLI der Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden die Registrierungen der Patchgruppen aufgeführt.

```
Get-SSMPatchGroup
```

Ausgabe:

```
BaselineIdentity                                          PatchGroup
----------------                                          ----------
Amazon.SimpleSystemsManagement.Model.PatchBaselineIdentity Production
```

- Einzelheiten zur API finden Sie unter DescribePatchGroups AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden die Registrierungen der Patchgruppen aufgeführt.

```
Get-SSMPatchGroup
```

Ausgabe:

```
BaselineIdentity                                          PatchGroup
----------------                                          ----------
Amazon.SimpleSystemsManagement.Model.PatchBaselineIdentity Production
```

- Einzelheiten zur API finden Sie unter DescribePatchGroups AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`GetAutomationExecution`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetAutomationExecution` verwendet wird.

CLI

AWS CLI

So zeigen Sie Details zu einer Automatisierungsausführung an

Im folgenden `get-automation-execution`-Beispiel werden detaillierte Informationen zu einer Automation-Ausführung angezeigt.

```
aws ssm get-automation-execution \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Ausgabe:

```
{
    "AutomationExecution": {
        "AutomationExecutionId": "73c8eef8-f4ee-4a05-820c-e354fEXAMPLE",
        "DocumentName": "AWS-StartEC2Instance",
        "DocumentVersion": "1",
        "ExecutionStartTime": 1583737233.748,
        "ExecutionEndTime": 1583737234.719,
        "AutomationExecutionStatus": "Success",
        "StepExecutions": [
            {
                "StepName": "startInstances",
                "Action": "aws:changeInstanceState",
                "ExecutionStartTime": 1583737234.134,
                "ExecutionEndTime": 1583737234.672,
                "StepStatus": "Success",
                "Inputs": {
```

```
                    "DesiredState": "\"running\"",
                    "InstanceIds": "[\"i-0cb99161f6EXAMPLE\"]"
                },
                "Outputs": {
                    "InstanceStates": [
                        "running"
                    ]
                },
                "StepExecutionId": "95e70479-cf20-4d80-8018-7e4e2EXAMPLE",
                "OverriddenParameters": {}
            }
        ],
        "StepExecutionsTruncated": false,
        "Parameters": {
            "AutomationAssumeRole": [
                ""
            ],
            "InstanceId": [
                "i-0cb99161f6EXAMPLE"
            ]
        },
        "Outputs": {},
        "Mode": "Auto",
        "ExecutedBy": "arn:aws:sts::29884EXAMPLE:assumed-role/mw_service_role/
OrchestrationService",
        "Targets": [],
        "ResolvedTargets": {
            "ParameterValues": [],
            "Truncated": false
        }
    }
}
```

Weitere Informationen finden Sie unter Exemplarische Vorgehensweise: Patchen eines Linux-AMI (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie GetAutomationExecutionin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden die Details einer Automation-Ausführung angezeigt.

```
Get-SSMAutomationExecution -AutomationExecutionId "4105a4fc-
f944-11e6-9d32-8fb2db27a909"
```

Ausgabe:

```
AutomationExecutionId     : 4105a4fc-f944-11e6-9d32-8fb2db27a909
AutomationExecutionStatus : Failed
DocumentName              : AWS-UpdateLinuxAmi
DocumentVersion           : 1
ExecutionEndTime          : 2/22/2017 9:17:08 PM
ExecutionStartTime        : 2/22/2017 9:17:02 PM
FailureMessage            : Step launchInstance failed maximum allowed times. You
 are not authorized to perform this operation. Encoded
                            authorization failure message:
 B_V2QyyN7NhSZQYpmVzpEc4oSnj2GLTNYnXUHsTbqJkNMoDgubmbtthLmZyaiUYekORIrA42-
fv1x-04q5Fjff6glh
                            Yb6TI5b0GQeeNrpwNvpDzmO-
PSR1swlAbg9fdM9BcNjyrznspUkWpuKu9EC1Ou6v3OXU1KC9nZ7mPlWMFZNkSioQqpwWEvMw-
GZktsQzm67qOhUhBNOLWYhbS
                            pkfiqzY-5nw3S0obx30fhd3EJa5O_-
GjV_a0nFXQJa70ik40bFOrEh3MtCSbrQT6--DvFy_FQ8TKvkIXadyVskeJI84XOF5WmA60f1pi5GI08i-
nRfZS6oDeU

 gELBjjoFKD8s3L2aI0B6umWVxnQOjqhQRxwJ53b54sZJ2PW3v_mtg9-q0CK0ezS3xfh_y0ilaUGOAZG-
xjQFuvU_JZedWpla3xi-MZsmblAifBI
                            (Service: AmazonEC2; Status Code: 403; Error Code:
 UnauthorizedOperation; Request ID:
                            6a002f94-ba37-43fd-99e6-39517715fce5)
Outputs                   : {[createImage.ImageId,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
Parameters                : {[AutomationAssumeRole,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]], [InstanceIamRole,

 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]], [SourceAmiId,

 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
StepExecutions            : {launchInstance, updateOSSoftware, stopInstance,
 createImage...}
```

Beispiel 2: In diesem Beispiel werden die Schrittdetails für die angegebene Automatisierungsausführungs-ID aufgeführt

```
Get-SSMAutomationExecution -AutomationExecutionId e1d2bad3-4567-8901-
ae23-456c7c8901be | Select-Object -ExpandProperty StepExecutions | Select-Object
 StepName, Action, StepStatus, ValidNextSteps
```

Ausgabe:

```
StepName                  Action                   StepStatus ValidNextSteps
--------                  ------                   ---------- --------------
LaunchInstance            aws:runInstances         Success
 {OSCompatibilityCheck}
OSCompatibilityCheck      aws:runCommand           Success    {RunPreUpdateScript}
RunPreUpdateScript        aws:runCommand           Success    {UpdateEC2Config}
UpdateEC2Config           aws:runCommand           Cancelled  {}
UpdateSSMAgent            aws:runCommand           Pending    {}
UpdateAWSPVDriver         aws:runCommand           Pending    {}
UpdateAWSEnaNetworkDriver aws:runCommand           Pending    {}
UpdateAWSNVMe             aws:runCommand           Pending    {}
InstallWindowsUpdates     aws:runCommand           Pending    {}
RunPostUpdateScript       aws:runCommand           Pending    {}
RunSysprepGeneralize      aws:runCommand           Pending    {}
StopInstance              aws:changeInstanceState  Pending    {}
CreateImage               aws:createImage          Pending    {}
TerminateInstance         aws:changeInstanceState  Pending    {}
```

- Einzelheiten zur API finden Sie unter GetAutomationExecution AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden die Details einer Automation-Ausführung angezeigt.

```
Get-SSMAutomationExecution -AutomationExecutionId "4105a4fc-
f944-11e6-9d32-8fb2db27a909"
```

Ausgabe:

```
AutomationExecutionId     : 4105a4fc-f944-11e6-9d32-8fb2db27a909
AutomationExecutionStatus : Failed
DocumentName              : AWS-UpdateLinuxAmi
DocumentVersion           : 1
ExecutionEndTime          : 2/22/2017 9:17:08 PM
ExecutionStartTime        : 2/22/2017 9:17:02 PM
```

```
FailureMessage            : Step launchInstance failed maximum allowed times. You
 are not authorized to perform this operation. Encoded
                            authorization failure message:
 B_V2QyyN7NhSZQYpmVzpEc4oSnj2GLTNYnXUHsTbqJkNMoDgubmbtthLmZyaiUYekORIrA42-
fv1x-04q5Fjff6glh
                            Yb6TI5b0GQeeNrpwNvpDzmO-
PSR1swlAbg9fdM9BcNjyrznspUkWpuKu9EC1Ou6v3OXU1KC9nZ7mPlWMFZNkSioQqpwWEvMw-
GZktsQzm67qOhUhBNOLWYhbS
                            pkfiqzY-5nw3S0obx30fhd3EJa5O_-
GjV_a0nFXQJa70ik40bFOrEh3MtCSbrQT6--DvFy_FQ8TKvkIXadyVskeJI84XOF5WmA60f1pi5GI08i-
nRfZS6oDeU

 gELBjjoFKD8s3L2aI0B6umWVxnQOjqhQRxwJ53b54sZJ2PW3v_mtg9-q0CK0ezS3xfh_y0ilaUGOAZG-
xjQFuvU_JZedWpla3xi-MZsmblAifBI
                            (Service: AmazonEC2; Status Code: 403; Error Code:
 UnauthorizedOperation; Request ID:
                            6a002f94-ba37-43fd-99e6-39517715fce5)
Outputs                   : {[createImage.ImageId,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
Parameters                : {[AutomationAssumeRole,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]], [InstanceIamRole,

 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]], [SourceAmiId,

 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
StepExecutions            : {launchInstance, updateOSSoftware, stopInstance,
 createImage...}
```

Beispiel 2: In diesem Beispiel werden die Schrittdetails für die angegebene Automatisierungsausführungs-ID aufgeführt

```
Get-SSMAutomationExecution -AutomationExecutionId e1d2bad3-4567-8901-
ae23-456c7c8901be | Select-Object -ExpandProperty StepExecutions | Select-Object
 StepName, Action, StepStatus, ValidNextSteps
```

Ausgabe:

```
StepName                  Action                 StepStatus ValidNextSteps
--------                  ------                 ---------- --------------
LaunchInstance            aws:runInstances       Success
 {OSCompatibilityCheck}
OSCompatibilityCheck      aws:runCommand         Success    {RunPreUpdateScript}
RunPreUpdateScript        aws:runCommand         Success    {UpdateEC2Config}
```

```
UpdateEC2Config           aws:runCommand          Cancelled  {}
UpdateSSMAgent            aws:runCommand          Pending    {}
UpdateAWSPVDriver         aws:runCommand          Pending    {}
UpdateAWSEnaNetworkDriver aws:runCommand          Pending    {}
UpdateAWSNVMe             aws:runCommand          Pending    {}
InstallWindowsUpdates     aws:runCommand          Pending    {}
RunPostUpdateScript       aws:runCommand          Pending    {}
RunSysprepGeneralize      aws:runCommand          Pending    {}
StopInstance              aws:changeInstanceState Pending    {}
CreateImage               aws:createImage         Pending    {}
TerminateInstance         aws:changeInstanceState Pending    {}
```

- Einzelheiten zur API finden Sie unter GetAutomationExecution AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`GetCommandInvocation`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetCommandInvocation` verwendet wird.

CLI

AWS CLI

So zeigen Sie die Details eines Befehlsaufrufs an

Das folgende `get-command-invocation`-Beispiel listet alle Aufrufe des angegebenen Befehls auf der angegebenen Instance auf.

```
aws ssm get-command-invocation \
    --command-id "ef7fdfd8-9b57-4151-a15c-db9a12345678" \
    --instance-id "i-1234567890abcdef0"
```

Ausgabe:

```
{
    "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",
    "InstanceId": "i-1234567890abcdef0",
```

```
    "Comment": "b48291dd-ba76-43e0-b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-
d6ce8EXAMPLE",
    "DocumentName": "AWS-UpdateSSMAgent",
    "DocumentVersion": "",
    "PluginName": "aws:updateSsmAgent",
    "ResponseCode": 0,
    "ExecutionStartDateTime": "2020-02-19T18:18:03.419Z",
    "ExecutionElapsedTime": "PT0.091S",
    "ExecutionEndDateTime": "2020-02-19T18:18:03.419Z",
    "Status": "Success",
    "StatusDetails": "Success",
    "StandardOutputContent": "Updating amazon-ssm-agent from 2.3.842.0 to latest
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-
east-2/ssm-agent-manifest.json\namazon-ssm-agent 2.3.842.0 has already been
 installed, update skipped\n",
    "StandardOutputUrl": "",
    "StandardErrorContent": "",
    "StandardErrorUrl": "",
    "CloudWatchOutputConfig": {
        "CloudWatchLogGroupName": "",
        "CloudWatchOutputEnabled": false
    }
}
```

Weitere Informationen finden Sie unter Befehlsstatus verstehen im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie GetCommandInvocationin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden die Details eines Befehls angezeigt, der auf einer Instance ausgeführt wurde.

```
Get-SSMCommandInvocationDetail -InstanceId "i-0cb2b964d3e14fd9f" -CommandId
 "b8eac879-0541-439d-94ec-47a80d554f44"
```

Ausgabe:

```
CommandId              : b8eac879-0541-439d-94ec-47a80d554f44
```

```
Comment                : IP config
DocumentName           : AWS-RunShellScript
ExecutionElapsedTime   : PT0.004S
ExecutionEndDateTime   : 2017-02-22T20:13:16.651Z
ExecutionStartDateTime : 2017-02-22T20:13:16.651Z
InstanceId             : i-0cb2b964d3e14fd9f
PluginName             : aws:runShellScript
ResponseCode           : 0
StandardErrorContent   :
StandardErrorUrl       :
StandardOutputContent  :
StandardOutputUrl      :
Status                 : Success
StatusDetails          : Success
```

- Einzelheiten zur API finden Sie unter GetCommandInvocation AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden die Details eines Befehls angezeigt, der auf einer Instance ausgeführt wurde.

```
Get-SSMCommandInvocationDetail -InstanceId "i-0cb2b964d3e14fd9f" -CommandId
 "b8eac879-0541-439d-94ec-47a80d554f44"
```

Ausgabe:

```
CommandId              : b8eac879-0541-439d-94ec-47a80d554f44
Comment                : IP config
DocumentName           : AWS-RunShellScript
ExecutionElapsedTime   : PT0.004S
ExecutionEndDateTime   : 2017-02-22T20:13:16.651Z
ExecutionStartDateTime : 2017-02-22T20:13:16.651Z
InstanceId             : i-0cb2b964d3e14fd9f
PluginName             : aws:runShellScript
ResponseCode           : 0
StandardErrorContent   :
StandardErrorUrl       :
StandardOutputContent  :
StandardOutputUrl      :
Status                 : Success
StatusDetails          : Success
```

- Einzelheiten zur API finden Sie unter GetCommandInvocation AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`GetConnectionStatus`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetConnectionStatus` verwendet wird.

CLI

AWS CLI

Um den Verbindungsstatus einer verwalteten Instance anzuzeigen

In diesem `get-connection-status`-Beispiel wird der Verbindungsstatus der angegebenen verwalteten Instance zurückgegeben.

```
aws ssm get-connection-status \
    --target i-1234567890abcdef0
```

Ausgabe:

```
{
    "Target": "i-1234567890abcdef0",
    "Status": "connected"
}
```

- Einzelheiten zur API finden Sie GetConnectionStatusin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird der Session-Manager-Verbindungsstatus für eine Instance abgerufen, um festzustellen, ob sie verbunden und bereit ist, Session-Manager-Verbindungen zu empfangen.

```
Get-SSMConnectionStatus -Target i-0a1caf234f12d3dc4
```

Ausgabe:

```
Status    Target
------    ------
Connected i-0a1caf234f12d3dc4
```

- Einzelheiten zur API finden Sie unter GetConnectionStatus AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird der Session-Manager-Verbindungsstatus für eine Instance abgerufen, um festzustellen, ob sie verbunden und bereit ist, Session-Manager-Verbindungen zu empfangen.

```
Get-SSMConnectionStatus -Target i-0a1caf234f12d3dc4
```

Ausgabe:

```
Status    Target
------    ------
Connected i-0a1caf234f12d3dc4
```

- Einzelheiten zur API finden Sie unter GetConnectionStatus AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`GetDefaultPatchBaseline`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetDefaultPatchBaseline` verwendet wird.

CLI

AWS CLI

Beispiel 1: So zeigen Sie die Standard-Windows-Patch-Baseline an

Im folgenden Beispiel `get-default-patch-baseline` werden Details für die Standard-Patch-Baseline für Windows Server abgerufen.

```
aws ssm get-default-patch-baseline
```

Ausgabe:

```
{
  "BaselineId": "pb-0713accee01612345",
  "OperatingSystem": "WINDOWS"
}
```

Beispiel 2: So zeigen Sie die Standard-Patch-Baseline für Amazon Linux an

Im folgenden Beispiel `get-default-patch-baseline` werden Details für die Standard-Patch-Baseline für Amazon Linux abgerufen.

```
aws ssm get-default-patch-baseline \
    --operating-system AMAZON_LINUX
```

Ausgabe:

```
{
    "BaselineId": "pb-047c6eb9c8fc12345",
    "OperatingSystem": "AMAZON_LINUX"
}
```

Weitere Informationen finden Sie unter Über vordefinierte und benutzerdefinierte Patch-Baselines < https://docs.aws.amazon.com/systems-manager/ latest/userguide/sysman -patch-baselines.html>__ und Eine bestehende Patch-Baseline als Standard festlegen im Systems Manager Manager-Benutzerhandbuch.AWS

- GetDefaultPatchBaselineEinzelheiten AWS CLI zur API finden Sie in der Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird die Standard-Patch-Baseline angezeigt.

```
Get-SSMDefaultPatchBaseline
```

Ausgabe:

```
arn:aws:ssm:us-west-2:123456789012:patchbaseline/pb-04fb4ae6142167966
```

- Einzelheiten zur API finden Sie unter GetDefaultPatchBaseline AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird die Standard-Patch-Baseline angezeigt.

```
Get-SSMDefaultPatchBaseline
```

Ausgabe:

```
arn:aws:ssm:us-west-2:123456789012:patchbaseline/pb-04fb4ae6142167966
```

- Einzelheiten zur API finden Sie unter GetDefaultPatchBaseline AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`GetDeployablePatchSnapshotForInstance`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetDeployablePatchSnapshotForInstance` verwendet wird.

CLI

AWS CLI

Um den aktuellen Snapshot für die Patch-Baseline abzurufen, verwendet eine Instance

Im folgenden `get-deployable-patch-snapshot-for-instance`-Beispiel werden Details für den aktuellen Snapshot für die angegebene Patch-Baseline abgerufen, die

von einer Instance verwendet wird. Dieser Befehl muss von der Instance aus mit den Anmeldeinformationen der Instance ausgeführt werden. Um sicherzustellen, dass er die Instance-Anmeldeinformationen verwendet, führen Sie `aws configure` aus und geben Sie nur die Region Ihrer Instance an. Lassen Sie die Felder `Access Key` und `Secret Key` leer.

Tipp: Verwenden Sie `uuidgen`, um eine `snapshot-id` zu generieren.

```
aws ssm get-deployable-patch-snapshot-for-instance \
    --instance-id "i-1234567890abcdef0" \
    --snapshot-id "521c3536-930c-4aa9-950e-01234567abcd"
```

Ausgabe:

```
{
    "InstanceId": "i-1234567890abcdef0",
    "SnapshotId": "521c3536-930c-4aa9-950e-01234567abcd",
    "Product": "AmazonLinux2018.03",
    "SnapshotDownloadUrl": "https://patch-baseline-snapshot-us-
east-1.s3.amazonaws.com/
ed85194ef27214f5984f28b4d664d14f7313568fea7d4b6ac6c10ad1f729d7e7-773304212436/
AMAZON_LINUX-521c3536-930c-4aa9-950e-01234567abcd?X-Amz-
Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20190215T164031Z&X-Amz-
SignedHeaders=host&X-Amz-Expires=86400&X-Amz-Credential=AKIAJ5C56P35AEBRX2QQ
%2F20190215%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-
Signature=efaaaf6e3878e77f48a6697e015efdbda9c426b09c5822055075c062f6ad2149"
}
```

Weitere Informationen finden Sie unter Parametername: Snapshot-ID im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie GetDeployablePatchSnapshotForInstancein der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird der aktuelle Snapshot für die von einer Instance verwendete Patch-Baseline angezeigt. Dieser Befehl muss von der Instance aus mit den Anmeldeinformationen der Instance ausgeführt werden. Um sicherzustellen, dass die Instance-Anmeldedaten verwendet werden, übergibt das Beispiel ein

`Amazon.Runtime.InstanceProfileAWSCredentials`-Objekt an den Anmeldeinformationen-Parameter.

```
$credentials = [Amazon.Runtime.InstanceProfileAWSCredentials]::new()
Get-SSMDeployablePatchSnapshotForInstance -SnapshotId "4681775b-098f-4435-
a956-0ef33373ac11" -InstanceId "i-0cb2b964d3e14fd9f" -Credentials $credentials
```

Ausgabe:

```
InstanceId          SnapshotDownloadUrl
----------          -------------------
i-0cb2b964d3e14fd9f https://patch-baseline-snapshot-us-west-2.s3-us-
west-2.amazonaws.com/853d0d3db0f0cafe...1692/4681775b-098f-4435...
```

Beispiel 2: Dieses Beispiel zeigt, wie man die volle Version bekommt SnapshotDownloadUrl. Dieser Befehl muss von der Instance aus mit den Anmeldeinformationen der Instance ausgeführt werden. Um sicherzustellen, dass die Instanzanmeldedaten verwendet werden, konfiguriert das Beispiel die PowerShell Sitzung für die Verwendung eines **`Amazon.Runtime.InstanceProfileAWSCredentials`** Objekts.

```
Set-AWSCredential -Credential
 ([Amazon.Runtime.InstanceProfileAWSCredentials]::new())
(Get-SSMDeployablePatchSnapshotForInstance -SnapshotId "4681775b-098f-4435-
a956-0ef33373ac11" -InstanceId "i-0cb2b964d3e14fd9f").SnapshotDownloadUrl
```

Ausgabe:

```
https://patch-baseline-snapshot-us-west-2.s3-us-
west-2.amazonaws.com/853d0d3db0f0cafe...
```

- Einzelheiten zur API finden Sie unter GetDeployablePatchSnapshotForInstance AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird der aktuelle Snapshot für die von einer Instance verwendete Patch-Baseline angezeigt. Dieser Befehl muss von der Instance aus mit den Anmeldeinformationen der Instance ausgeführt werden. Um sicherzustellen, dass die Instance-Anmeldedaten verwendet werden, übergibt das Beispiel ein

`Amazon.Runtime.InstanceProfileAWSCredentials`-Objekt an den Anmeldeinformationen-Parameter.

```
$credentials = [Amazon.Runtime.InstanceProfileAWSCredentials]::new()
Get-SSMDeployablePatchSnapshotForInstance -SnapshotId "4681775b-098f-4435-
a956-0ef33373ac11" -InstanceId "i-0cb2b964d3e14fd9f" -Credentials $credentials
```

Ausgabe:

```
InstanceId          SnapshotDownloadUrl
----------          -------------------
i-0cb2b964d3e14fd9f https://patch-baseline-snapshot-us-west-2.s3-us-
west-2.amazonaws.com/853d0d3db0f0cafe...1692/4681775b-098f-4435...
```

Beispiel 2: Dieses Beispiel zeigt, wie man das volle SnapshotDownloadUrl bekommt. Dieser Befehl muss von der Instance aus mit den Anmeldeinformationen der Instance ausgeführt werden. Um sicherzustellen, dass die Instanzanmeldedaten verwendet werden, konfiguriert das Beispiel die PowerShell Sitzung für die Verwendung eines **`Amazon.Runtime.InstanceProfileAWSCredentials`** Objekts.

```
Set-AWSCredential -Credential
 ([Amazon.Runtime.InstanceProfileAWSCredentials]::new())
(Get-SSMDeployablePatchSnapshotForInstance -SnapshotId "4681775b-098f-4435-
a956-0ef33373ac11" -InstanceId "i-0cb2b964d3e14fd9f").SnapshotDownloadUrl
```

Ausgabe:

```
https://patch-baseline-snapshot-us-west-2.s3-us-
west-2.amazonaws.com/853d0d3db0f0cafe...
```

- Einzelheiten zur API finden Sie unter GetDeployablePatchSnapshotForInstance AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`GetDocument`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetDocument` verwendet wird.

CLI

AWS CLI

So rufen Sie den Inhalt des Dokuments ab

Im folgenden `get-document`-Beispiel wird der Inhalt eines Systems-Manager-Dokuments angezeigt.

```
aws ssm get-document \
    --name "AWS-RunShellScript"
```

Ausgabe:

```
{
    "Name": "AWS-RunShellScript",
    "DocumentVersion": "1",
    "Status": "Active",
    "Content": "{\n    \"schemaVersion\":\"1.2\",\n    \"description\":\"Run
 a shell script or specify the commands to run.\",\n    \"parameters\":{\n
        \"commands\":{\n            \"type\":\"StringList\",\n
 \"description\":\"(Required) Specify a shell script or a command to run.\",
\n            \"minItems\":1,\n            \"displayType\":\"textarea\"\n
      },\n        \"workingDirectory\":{\n            \"type\":\"String\",\n
            \"default\":\"\",\n            \"description\":\"(Optional) The
 path to the working directory on your instance.\",\n            \"maxChars
\":4096\n        },\n        \"executionTimeout\":{\n            \"type\":
\"String\",\n            \"default\":\"3600\",\n            \"description
\":\"(Optional) The time in seconds for a command to complete before it is
 considered to have failed. Default is 3600 (1 hour). Maximum is 172800 (48
 hours).\",\n            \"allowedPattern\":\"([1-9][0-9]{0,4})|(1[0-6][0-9]
{4})|(17[0-1][0-9]{3})|(172[0-7][0-9]{2})|(172800)\"\n        }\n    },\n
 \"runtimeConfig\":{\n        \"aws:runShellScript\":{\n            \"properties
\":[\n                {\n                    \"id\":\"0.aws:runShellScript
\",\n                    \"runCommand\":\"{{ commands }}\",\n
      \"workingDirectory\":\"{{ workingDirectory }}\",\n
 \"timeoutSeconds\":\"{{ executionTimeout }}\"\n                }\n
  ]\n        }\n    }\n}\n",
    "DocumentType": "Command",
    "DocumentFormat": "JSON"
}
```

Weitere Informationen finden Sie unter AWS -Systems-Manager-Dokumenten im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie GetDocumentin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird der Inhalt eines Dokuments zurückgegeben.

```
Get-SSMDocument -Name "RunShellScript"
```

Ausgabe:

```
Content
-------
{...
```

Beispiel 2: In diesem Beispiel wird der vollständige Inhalt eines Dokuments angezeigt.

```
(Get-SSMDocument -Name "RunShellScript").Content
{
   "schemaVersion":"2.0",
   "description":"Run an updated script",
   "parameters":{
      "commands":{
         "type":"StringList",
         "description":"(Required) Specify a shell script or a command to run.",
         "minItems":1,
         "displayType":"textarea"
      }
   },
   "mainSteps":[
      {
         "action":"aws:runShellScript",
         "name":"runShellScript",
         "inputs":{
            "commands":"{{ commands }}"
         }
      },
      {
```

```
         "action":"aws:runPowerShellScript",
         "name":"runPowerShellScript",
         "inputs":{
            "commands":"{{ commands }}"
         }
      }
   ]
}
```

- Einzelheiten zur API finden Sie unter GetDocument AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird der Inhalt eines Dokuments zurückgegeben.

```
Get-SSMDocument -Name "RunShellScript"
```

Ausgabe:

```
Content
-------
{...
```

Beispiel 2: In diesem Beispiel wird der vollständige Inhalt eines Dokuments angezeigt.

```
(Get-SSMDocument -Name "RunShellScript").Content
{
   "schemaVersion":"2.0",
   "description":"Run an updated script",
   "parameters":{
      "commands":{
         "type":"StringList",
         "description":"(Required) Specify a shell script or a command to run.",
         "minItems":1,
         "displayType":"textarea"
      }
   },
   "mainSteps":[
      {
         "action":"aws:runShellScript",
         "name":"runShellScript",
```

```
            "inputs":{
               "commands":"{{ commands }}"
            }
         },
         {
            "action":"aws:runPowerShellScript",
            "name":"runPowerShellScript",
            "inputs":{
               "commands":"{{ commands }}"
            }
         }
      ]
}
```

- Einzelheiten zur API finden Sie unter GetDocument AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`GetInventory`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetInventory` verwendet wird.

CLI

AWS CLI

Um Ihr Inventar einzusehen

In diesem Beispiel werden die benutzerdefinierten Metadaten für Ihr Inventar abgerufen.

Befehl:

```
aws ssm get-inventory
```

Ausgabe:

```
{
  "Entities": [
```

```
      {
          "Data": {
              "AWS:InstanceInformation": {
                  "Content": [
                      {
                          "ComputerName": "ip-172-31-44-222.us-
west-2.compute.internal",
                          "InstanceId": "i-0cb2b964d3e14fd9f",
                          "IpAddress": "172.31.44.222",
                          "AgentType": "amazon-ssm-agent",
                          "ResourceType": "EC2Instance",
                          "AgentVersion": "2.0.672.0",
                          "PlatformVersion": "2016.09",
                          "PlatformName": "Amazon Linux AMI",
                          "PlatformType": "Linux"
                      }
                  ],
                  "TypeName": "AWS:InstanceInformation",
                  "SchemaVersion": "1.0",
                  "CaptureTime": "2017-02-20T18:03:58Z"
              }
          },
          "Id": "i-0cb2b964d3e14fd9f"
      }
  ]
}
```

- Einzelheiten zur API finden Sie GetInventoryin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden die benutzerdefinierten Metadaten für Ihr Inventar abgerufen.

```
Get-SSMInventory
```

Ausgabe:

```
Data
     Id
```

```
----
     --
{[AWS:InstanceInformation,
 Amazon.SimpleSystemsManagement.Model.InventoryResultItem]} i-0cb2b964d3e14fd9f
```

- Einzelheiten zur API finden Sie unter GetInventory AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden die benutzerdefinierten Metadaten für Ihr Inventar abgerufen.

```
Get-SSMInventory
```

Ausgabe:

```
Data
     Id
----
     --
{[AWS:InstanceInformation,
 Amazon.SimpleSystemsManagement.Model.InventoryResultItem]} i-0cb2b964d3e14fd9f
```

- Einzelheiten zur API finden Sie unter GetInventory AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`GetInventorySchema`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetInventorySchema` verwendet wird.

CLI

AWS CLI

So zeigen Sie Ihr Inventarschema an

Dieses Beispiel gibt eine Liste mit Inventartypnamen für das Konto zurück.

Befehl:

```
aws ssm get-inventory-schema
```

Ausgabe:

```
{
  "Schemas": [
      {
          "TypeName": "AWS:AWSComponent",
          "Version": "1.0",
          "Attributes": [
              {
                  "Name": "Name",
                  "DataType": "STRING"
              },
              {
                  "Name": "ApplicationType",
                  "DataType": "STRING"
              },
              {
                  "Name": "Publisher",
                  "DataType": "STRING"
              },
              {
                  "Name": "Version",
                  "DataType": "STRING"
              },
              {
                  "Name": "InstalledTime",
                  "DataType": "STRING"
              },
              {
                  "Name": "Architecture",
                  "DataType": "STRING"
              },
              {
                  "Name": "URL",
                  "DataType": "STRING"
              }
          ]
      },
      ...
```

```
  ],
  "NextToken": "--token string truncated--"
}
```

So zeigen Sie das Inventarschema für einen bestimmten Inventartyp an

In diesem Beispiel wird das Inventarschema für den AWS Inventartyp „AWS Komponente“ zurückgegeben.

Befehl:

```
aws ssm get-inventory-schema --type-name "AWS:AWSComponent"
```

- Einzelheiten zur API finden Sie GetInventorySchemain der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird eine Liste von Inventartypnamen für das Konto zurückgegeben.

```
Get-SSMInventorySchema
```

- Einzelheiten zur API finden Sie unter GetInventorySchema AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird eine Liste von Inventartypnamen für das Konto zurückgegeben.

```
Get-SSMInventorySchema
```

- Einzelheiten zur API finden Sie unter GetInventorySchema AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **GetMaintenanceWindow** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetMaintenanceWindow` verwendet wird.

CLI

AWS CLI

Anzeigen von Informationen zu Wartungsfenstern

Im folgenden `get-maintenance-window`-Beispiel werden Details zum angegebenen Wartungsfenster abgerufen.

```
aws ssm get-maintenance-window \
    --window-id "mw-03eb9db428EXAMPLE"
```

Ausgabe:

```
{
    "AllowUnassociatedTargets": true,
    "CreatedDate": 1515006912.957,
    "Cutoff": 1,
    "Duration": 6,
    "Enabled": true,
    "ModifiedDate": 2020-01-01T10:04:04.099Z,
    "Name": "My-Maintenance-Window",
    "Schedule": "rate(3 days)",
    "WindowId": "mw-03eb9db428EXAMPLE",
    "NextExecutionTime": "2020-02-25T00:08:15.099Z"
}
```

Weitere Informationen finden Sie unter Informationen zu Wartungsfenstern (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie GetMaintenanceWindowunter AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden Details zu einem Wartungsfenster abgerufen.

```
Get-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d"
```

Ausgabe:

```
AllowUnassociatedTargets : False
CreatedDate              : 2/20/2017 6:14:05 PM
Cutoff                   : 1
Duration                 : 2
Enabled                  : True
ModifiedDate             : 2/20/2017 6:14:05 PM
Name                     : TestMaintWin
Schedule                 : cron(0 */30 * * * ? *)
WindowId                 : mw-03eb9db42890fb82d
```

- Einzelheiten zur API finden Sie unter GetMaintenanceWindow AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden Details zu einem Wartungsfenster abgerufen.

```
Get-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d"
```

Ausgabe:

```
AllowUnassociatedTargets : False
CreatedDate              : 2/20/2017 6:14:05 PM
Cutoff                   : 1
Duration                 : 2
Enabled                  : True
ModifiedDate             : 2/20/2017 6:14:05 PM
Name                     : TestMaintWin
Schedule                 : cron(0 */30 * * * ? *)
WindowId                 : mw-03eb9db42890fb82d
```

- Einzelheiten zur API finden Sie unter GetMaintenanceWindow AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **GetMaintenanceWindowExecution** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetMaintenanceWindowExecution` verwendet wird.

CLI

AWS CLI

So erhalten Sie Informationen zur Ausführung einer Wartungsfensteraufgabe

Das folgende `get-maintenance-window-execution`-Beispiel listet Informationen zu einer Aufgabe auf, die im Rahmen der Ausführung des angegebenen Wartungsfensters ausgeführt wurde.

```
aws ssm get-maintenance-window-execution \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

Ausgabe:

```
{
    "Status": "SUCCESS",
    "TaskIds": [
        "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
    ],
    "StartTime": 1487692834.595,
    "EndTime": 1487692835.051,
    "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
}
```

Weitere Informationen finden Sie unter Informationen zu Aufgaben und Aufgabenausführungen (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie GetMaintenanceWindowExecutionin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dieses Beispiel listet Informationen über eine Aufgabenausführungen als Teil einer Wartungsfenster-Ausführung auf.

```
Get-SSMMaintenanceWindowExecution -WindowExecutionId "518d5565-5969-4cca-8f0e-
da3b2a638355"
```

Ausgabe:

```
EndTime           : 2/21/2017 4:00:35 PM
StartTime         : 2/21/2017 4:00:34 PM
Status            : FAILED
StatusDetails     : One or more tasks in the orchestration failed.
TaskIds           : {ac0c6ae1-daa3-4a89-832e-d384503b6586}
WindowExecutionId : 518d5565-5969-4cca-8f0e-da3b2a638355
```

- Einzelheiten zur API finden Sie unter GetMaintenanceWindowExecution AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dieses Beispiel listet Informationen über eine Aufgabenausführungen als Teil einer Wartungsfenster-Ausführung auf.

```
Get-SSMMaintenanceWindowExecution -WindowExecutionId "518d5565-5969-4cca-8f0e-
da3b2a638355"
```

Ausgabe:

```
EndTime           : 2/21/2017 4:00:35 PM
StartTime         : 2/21/2017 4:00:34 PM
Status            : FAILED
StatusDetails     : One or more tasks in the orchestration failed.
TaskIds           : {ac0c6ae1-daa3-4a89-832e-d384503b6586}
WindowExecutionId : 518d5565-5969-4cca-8f0e-da3b2a638355
```

- Einzelheiten zur API finden Sie unter GetMaintenanceWindowExecution AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **GetMaintenanceWindowExecutionTask** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetMaintenanceWindowExecutionTask` verwendet wird.

CLI

AWS CLI

So erhalten Sie Informationen zur Ausführung einer Wartungsfensteraufgabe

Das folgende `get-maintenance-window-execution-task`-Beispiel listet Informationen zu einer Aufgabe auf, die Teil der Ausführung des angegebenen Wartungsfensters ist.

```
aws ssm get-maintenance-window-execution-task \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE" \
    --task-id "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
```

Ausgabe:

```
{
    "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
    "TaskExecutionId": "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE",
    "TaskArn": "AWS-RunPatchBaseline",
    "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
    "Type": "RUN_COMMAND",
    "TaskParameters": [
        {
            "BaselineOverride": {
                "Values": [
                    ""
                ]
            },
            "InstallOverrideList": {
                "Values": [
                    ""
                ]
            },
            "Operation": {
                "Values": [
                    "Scan"
```

```
                ]
            },
            "RebootOption": {
                "Values": [
                    "RebootIfNeeded"
                ]
            },
            "SnapshotId": {
                "Values": [
                    "{{ aws:ORCHESTRATION_ID }}"
                ]
            },
            "aws:InstanceId": {
                "Values": [
                    "i-02573cafcfEXAMPLE",
                    "i-0471e04240EXAMPLE",
                    "i-07782c72faEXAMPLE"
                ]
            }
        }
    ],
    "Priority": 1,
    "MaxConcurrency": "1",
    "MaxErrors": "3",
    "Status": "SUCCESS",
    "StartTime": "2021-08-04T11:45:35.088000-07:00",
    "EndTime": "2021-08-04T11:53:09.079000-07:00"
}
```

Weitere Informationen finden Sie unter Informationen zu Aufgaben und Aufgabenausführungen (AWS CLI) anzeigen im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie GetMaintenanceWindowExecutionTaskin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden Informationen zu einer Aufgabe aufgeführt, die Teil einer Ausführung im Rahmen eines Wartungsfensters war.

```
Get-SSMMaintenanceWindowExecutionTask -TaskId "ac0c6ae1-daa3-4a89-832e-
d384503b6586" -WindowExecutionId "518d5565-5969-4cca-8f0e-da3b2a638355"
```

Ausgabe:

```
EndTime           : 2/21/2017 4:00:35 PM
MaxConcurrency    : 1
MaxErrors         : 1
Priority          : 10
ServiceRole       : arn:aws:iam::123456789012:role/MaintenanceWindowsRole
StartTime         : 2/21/2017 4:00:34 PM
Status            : FAILED
StatusDetails     : The maximum error count was exceeded.
TaskArn           : AWS-RunShellScript
TaskExecutionId   : ac0c6ae1-daa3-4a89-832e-d384503b6586
TaskParameters    :
 {Amazon.Runtime.Internal.Util.AlwaysSendDictionary`2[System.String,Amazon.SimpleSystemsM
                     meterValueExpression]}
Type              : RUN_COMMAND
WindowExecutionId : 518d5565-5969-4cca-8f0e-da3b2a638355
```

- Einzelheiten zur API finden Sie unter GetMaintenanceWindowExecutionTask AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden Informationen zu einer Aufgabe aufgeführt, die Teil einer Ausführung im Rahmen eines Wartungsfensters war.

```
Get-SSMMaintenanceWindowExecutionTask -TaskId "ac0c6ae1-daa3-4a89-832e-
d384503b6586" -WindowExecutionId "518d5565-5969-4cca-8f0e-da3b2a638355"
```

Ausgabe:

```
EndTime           : 2/21/2017 4:00:35 PM
MaxConcurrency    : 1
MaxErrors         : 1
Priority          : 10
ServiceRole       : arn:aws:iam::123456789012:role/MaintenanceWindowsRole
StartTime         : 2/21/2017 4:00:34 PM
Status            : FAILED
```

```
StatusDetails     : The maximum error count was exceeded.
TaskArn           : AWS-RunShellScript
TaskExecutionId   : ac0c6ae1-daa3-4a89-832e-d384503b6586
TaskParameters    :
 {Amazon.Runtime.Internal.Util.AlwaysSendDictionary`2[System.String,Amazon.SimpleSystemsM
                   meterValueExpression]}
Type              : RUN_COMMAND
WindowExecutionId : 518d5565-5969-4cca-8f0e-da3b2a638355
```

- Einzelheiten zur API finden Sie unter GetMaintenanceWindowExecutionTask AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GetParameter`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `GetParameter` verwendet wird.

CLI

AWS CLI

Beispiel 1: So zeigen Sie den Wert eines Parameters an

Das folgende `get-parameter`-Beispiel listet den Wert für den angegebenen Einzelparameter auf.

```
aws ssm get-parameter \
    --name "MyStringParameter"
```

Ausgabe:

```
{
    "Parameter": {
        "Name": "MyStringParameter",
        "Type": "String",
        "Value": "Veni",
        "Version": 1,
        "LastModifiedDate": 1530018761.888,
```

```
        "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/MyStringParameter"
        "DataType": "text"
    }
}
```

Weitere Informationen finden Sie unter Arbeiten mit Parameter Store im AWS -Systems-Manager-Benutzerhandbuch.

Beispiel 2: Um den Wert eines SecureString Parameters zu entschlüsseln

Im folgenden Beispiel `get-parameter` wird der Wert des angegebenen `SecureString`-Parameters entschlüsselt.

```
aws ssm get-parameter \
    --name "MySecureStringParameter" \
    --with-decryption
```

Ausgabe:

```
{
    "Parameter": {
        "Name": "MySecureStringParameter",
        "Type": "SecureString",
        "Value": "16679b88-310b-4895-a943-e0764EXAMPLE",
        "Version": 2,
        "LastModifiedDate": 1582155479.205,
        "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/
MySecureStringParameter"
        "DataType": "text"
    }
}
```

Weitere Informationen finden Sie unter Arbeiten mit Parameter Store im AWS -Systems-Manager-Benutzerhandbuch.

Beispiel 3: So können den Wert eines Parameters mithilfe von Beschriftungen anzeigen

Das folgende `get-parameter`-Beispiel listet den Wert für den angegebenen Einzelparameter mit einer angegebenen Bezeichnung auf.

```
aws ssm get-parameter \
```

```
    --name "MyParameter:label"
```

Ausgabe:

```
{
    "Parameter": {
        "Name": "MyParameter",
        "Type": "String",
        "Value": "parameter version 2",
        "Version": 2,
        "Selector": ":label",
        "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",
        "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",
        "DataType": "text"
    }
}
```

Weitere Informationen finden Sie unter Arbeiten mit Parameterbeschriftungen im AWS - Systems-Manager-Benutzerhandbuch.

Beispiel 4: So können den Wert eines Parameters mithilfe von Versionen anzeigen

Das folgende `get-parameter`-Beispiel listet den Wert für die angegebene Einzelparameter-Version auf.

```
aws ssm get-parameter \
    --name "MyParameter:2"
```

Ausgabe:

```
{
    "Parameter": {
        "Name": "MyParameter",
        "Type": "String",
        "Value": "parameter version 2",
        "Version": 2,
        "Selector": ":2",
        "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",
        "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",
        "DataType": "text"
    }
}
```

Weitere Informationen finden Sie unter Arbeiten mit Parameterbeschriftungen im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie GetParameterin der AWS CLI Befehlsreferenz.

Rust

SDK für Rust

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    pub async fn list_path(&self, path: &str) -> Result<Vec<Parameter>, EC2Error>
{
        let maybe_params: Vec<Result<Parameter, _>> = TryFlatMap::new(
            self.inner
                .get_parameters_by_path()
                .path(path)
                .into_paginator()
                .send(),
        )
        .flat_map(|item| item.parameters.unwrap_or_default())
        .collect()
        .await;
        // Fail on the first error
        let params = maybe_params
            .into_iter()
            .collect::<Result<Vec<Parameter>, _>>()?;
        Ok(params)
    }
```

- Einzelheiten zur API finden Sie GetParameterin der API-Referenz zum AWS SDK für Rust.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden dieses Dienstes mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`GetParameterHistory`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetParameterHistory` verwendet wird.

CLI

AWS CLI

Um einen Werteverlauf für einen Parameter abzurufen

Das folgende `get-parameter-history`-Beispiel listet den Verlauf der Änderungen für den angegebenen Parameter auf, einschließlich seines Werts.

```
aws ssm get-parameter-history \
    --name "MyStringParameter"
```

Ausgabe:

```
{
    "Parameters": [
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "LastModifiedDate": 1582154711.976,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is the first version of my String parameter",
            "Value": "Veni",
            "Version": 1,
            "Labels": [],
            "Tier": "Standard",
            "Policies": []
        },
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "LastModifiedDate": 1582156093.471,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is the second version of my String parameter",
            "Value": "Vidi",
            "Version": 2,
            "Labels": [],
            "Tier": "Standard",
            "Policies": []
```

```
        },
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "LastModifiedDate": 1582156117.545,
            "LastModifiedUser": "arn:aws:iam::111222333444:user/Mary-Major",
            "Description": "This is the third version of my String parameter",
            "Value": "Vici",
            "Version": 3,
            "Labels": [],
            "Tier": "Standard",
            "Policies": []
        }
    ]
}
```

Weitere Informationen finden Sie unter Arbeiten mit Parameterversionen im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie GetParameterHistoryin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird der Werteverlauf für einen Parameter aufgeführt.

```
Get-SSMParameterHistory -Name "Welcome"
```

Ausgabe:

```
Description      :
KeyId            :
LastModifiedDate : 3/3/2017 6:55:25 PM
LastModifiedUser : arn:aws:iam::123456789012:user/admin
Name             : Welcome
Type             : String
Value            : helloWorld
```

- Einzelheiten zur API finden Sie unter GetParameterHistory AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird der Werteverlauf für einen Parameter aufgeführt.

```
Get-SSMParameterHistory -Name "Welcome"
```

Ausgabe:

```
Description      :
KeyId            :
LastModifiedDate : 3/3/2017 6:55:25 PM
LastModifiedUser : arn:aws:iam::123456789012:user/admin
Name             : Welcome
Type             : String
Value            : helloWorld
```

- Einzelheiten zur API finden Sie unter GetParameterHistory AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`GetParameters`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetParameters` verwendet wird.

CLI

AWS CLI

Beispiel 1: So können Sie die Werte für einen Parameter auflisten

Das folgende `get-parameters`-Beispiel listet die Werte für die drei angegebenen Parameter auf.

```
aws ssm get-parameters \
    --names "MyStringParameter" "MyStringListParameter" "MyInvalidParameterName"
```

Ausgabe:

```
{
    "Parameters": [
        {
            "Name": "MyStringListParameter",
            "Type": "StringList",
            "Value": "alpha,beta,gamma",
            "Version": 1,
            "LastModifiedDate": 1582154764.222,
            "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/
MyStringListParameter"
            "DataType": "text"
        },
        {
            "Name": "MyStringParameter",
            "Type": "String",
            "Value": "Vici",
            "Version": 3,
            "LastModifiedDate": 1582156117.545,
            "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/
MyStringParameter"
            "DataType": "text"
        }
    ],
    "InvalidParameters": [
        "MyInvalidParameterName"
    ]
}
```

Weitere Informationen finden Sie unter Arbeiten mit Parameter Store im AWS -Systems-Manager-Benutzerhandbuch.

Beispiel 2: So können Sie Namen und Werte mehrerer Parameter mit der Option ``--query`` auflisten

Im folgenden Beispiel `get-parameters` werden die Namen und Werte für die angegebenen Parameter angezeigt.

```
aws ssm get-parameters \
    --names MyStringParameter MyStringListParameter \
    --query "Parameters[*].{Name:Name,Value:Value}"
```

Ausgabe:

```
[
    {
        "Name": "MyStringListParameter",
        "Value": "alpha,beta,gamma"
    },
    {
        "Name": "MyStringParameter",
        "Value": "Vidi"
    }
]
```

Weitere Informationen finden Sie unter Arbeiten mit Parameter Store im AWS -Systems-Manager-Benutzerhandbuch.

Beispiel 3: So können den Wert eines Parameters mithilfe von Beschriftungen anzeigen

Das folgende `get-parameter`-Beispiel listet den Wert für den angegebenen Einzelparameter mit einer angegebenen Bezeichnung auf.

```
aws ssm get-parameter \
    --name "MyParameter:label"
```

Ausgabe:

```
{
    "Parameters": [
        {
            "Name": "MyLabelParameter",
            "Type": "String",
            "Value": "parameter by label",
            "Version": 1,
            "Selector": ":label",
            "LastModifiedDate": "2021-07-12T09:49:15.865000-07:00",
            "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/MyParameter",
            "DataType": "text"
        },
        {
            "Name": "MyVersionParameter",
            "Type": "String",
            "Value": "parameter by version",
            "Version": 2,
            "Selector": ":2",
```

```
            "LastModifiedDate": "2021-03-24T16:20:28.236000-07:00",
            "ARN": "arn:aws:ssm:us-west-2:786973925828:parameter/unlabel-param",
            "DataType": "text"
        }
    ],
    "InvalidParameters": []
}
```

Weitere Informationen finden Sie unter Arbeiten mit Parameterbeschriftungen im AWS - Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie GetParametersin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden die Werte für einen Parameter aufgeführt.

```
Get-SSMParameterValue -Name "Welcome"
```

Ausgabe:

```
InvalidParameters Parameters
----------------- ----------
{}                {Welcome}
```

Beispiel 2: In diesem Beispiel werden die Details des Werts aufgeführt.

```
(Get-SSMParameterValue -Name "Welcome").Parameters
```

Ausgabe:

```
Name    Type   Value
----    ----   -----
Welcome String Good day, Sunshine!
```

- Einzelheiten zur API finden Sie unter GetParameters AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden die Werte für einen Parameter aufgeführt.

```
Get-SSMParameterValue -Name "Welcome"
```

Ausgabe:

```
InvalidParameters Parameters
----------------- ----------
{}                {Welcome}
```

Beispiel 2: In diesem Beispiel werden die Details des Werts aufgeführt.

```
(Get-SSMParameterValue -Name "Welcome").Parameters
```

Ausgabe:

```
Name    Type   Value
----    ----   -----
Welcome String Good day, Sunshine!
```

- Einzelheiten zur API finden Sie unter GetParameters AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`GetPatchBaseline`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetPatchBaseline` verwendet wird.

CLI

AWS CLI

So zeigen Sie eine Patch-Baseline an

Das folgende `get-patch-baseline`-Beispiel ruft die Details für die angegebene Patch-Baseline ab.

```
aws ssm get-patch-baseline \
    --baseline-id "pb-0123456789abcdef0"
```

Ausgabe:

```
{
    "BaselineId": "pb-0123456789abcdef0",
    "Name": "WindowsPatching",
    "OperatingSystem": "WINDOWS",
    "GlobalFilters": {
        "PatchFilters": []
    },
    "ApprovalRules": {
        "PatchRules": [
            {
                "PatchFilterGroup": {
                    "PatchFilters": [
                        {
                            "Key": "PRODUCT",
                            "Values": [
                                "WindowsServer2016"
                            ]
                        }
                    ]
                },
                "ComplianceLevel": "CRITICAL",
                "ApproveAfterDays": 0,
                "EnableNonSecurity": false
            }
        ]
    },
    "ApprovedPatches": [],
    "ApprovedPatchesComplianceLevel": "UNSPECIFIED",
    "ApprovedPatchesEnableNonSecurity": false,
    "RejectedPatches": [],
    "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY",
    "PatchGroups": [
        "QA",
        "DEV"
    ],
```

```
    "CreatedDate": 1550244180.465,
    "ModifiedDate": 1550244180.465,
    "Description": "Patches for Windows Servers",
    "Sources": []
}
```

Weitere Informationen finden Sie unter Über Patch-Baselines im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie GetPatchBaselinein der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden die Details für eine Patch-Baseline angezeigt.

```
Get-SSMPatchBaselineDetail -BaselineId "pb-03da896ca3b68b639"
```

Ausgabe:

```
ApprovalRules   : Amazon.SimpleSystemsManagement.Model.PatchRuleGroup
ApprovedPatches : {}
BaselineId      : pb-03da896ca3b68b639
CreatedDate     : 3/3/2017 5:02:19 PM
Description     : Baseline containing all updates approved for production systems
GlobalFilters   : Amazon.SimpleSystemsManagement.Model.PatchFilterGroup
ModifiedDate    : 3/3/2017 5:02:19 PM
Name            : Production-Baseline
PatchGroups     : {}
RejectedPatches : {}
```

- Einzelheiten zur API finden Sie unter GetPatchBaseline AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden die Details für eine Patch-Baseline angezeigt.

```
Get-SSMPatchBaselineDetail -BaselineId "pb-03da896ca3b68b639"
```

Ausgabe:

```
ApprovalRules   : Amazon.SimpleSystemsManagement.Model.PatchRuleGroup
ApprovedPatches : {}
BaselineId      : pb-03da896ca3b68b639
CreatedDate     : 3/3/2017 5:02:19 PM
Description     : Baseline containing all updates approved for production systems
GlobalFilters   : Amazon.SimpleSystemsManagement.Model.PatchFilterGroup
ModifiedDate    : 3/3/2017 5:02:19 PM
Name            : Production-Baseline
PatchGroups     : {}
RejectedPatches : {}
```

- Einzelheiten zur API finden Sie unter GetPatchBaseline AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`GetPatchBaselineForPatchGroup`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `GetPatchBaselineForPatchGroup` verwendet wird.

CLI

AWS CLI

So zeigen Sie die Patch-Baseline für eine Patch-Gruppe an

Das folgende `get-patch-baseline-for-patch-group`-Beispiel ruft die Details über die Patch-Baseline für die angegebene Patchgruppe ab.

```
aws ssm get-patch-baseline-for-patch-group \
    --patch-group "DEV"
```

Ausgabe:

```
{
    "PatchGroup": "DEV",
    "BaselineId": "pb-0123456789abcdef0",
    "OperatingSystem": "WINDOWS"
```

```
}
```

Weitere Informationen finden Sie unter Erstellen einer Patchgruppe < https://docs.aws.amazon.com/systems-manager/ latest/userguide/sysman - patch-group-tagging .html>__ und Hinzufügen einer Patchgruppe zu einer Patch-Baseline im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie GetPatchBaselineForPatchGroupin AWS CLI der Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird die Patch-Baseline für eine Patchgruppe angezeigt.

```
Get-SSMPatchBaselineForPatchGroup -PatchGroup "Production"
```

Ausgabe:

```
BaselineId           PatchGroup
----------           ----------
pb-045f10b4f382baeda Production
```

- Einzelheiten zur API finden Sie unter GetPatchBaselineForPatchGroup AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird die Patch-Baseline für eine Patchgruppe angezeigt.

```
Get-SSMPatchBaselineForPatchGroup -PatchGroup "Production"
```

Ausgabe:

```
BaselineId           PatchGroup
----------           ----------
pb-045f10b4f382baeda Production
```

- Einzelheiten zur API finden Sie unter GetPatchBaselineForPatchGroup AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`ListAssociationVersions`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `ListAssociationVersions` verwendet wird.

CLI

AWS CLI

So listen Sie alle Versionen einer Zuordnung für eine bestimmte Zuordnungs-ID auf

Das folgende `list-association-versions`-Beispiel listet alle Versionen der angegebenen Zuordnungen auf.

```
aws ssm list-association-versions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Ausgabe:

```
{
"AssociationVersions": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "CreatedDate": 1550505536.726,
            "Name": "AWS-UpdateSSMAgent",
            "Parameters": {
                "allowDowngrade": [
                    "false"
                ],
                "version": [
                    ""
                ]
            },
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-1234567890abcdef0"
```

```
                    ]
                }
            ],
            "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
            "AssociationName": "UpdateSSMAgent"
        }
    ]
}
```

Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie ListAssociationVersionsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden alle Versionen der angegebenen Zuordnung abgerufen.

```
Get-SSMAssociationVersionList -AssociationId 123a45a0-c678-9012-3456-78901234db5e
```

Ausgabe:

```
AssociationId      : 123a45a0-c678-9012-3456-78901234db5e
AssociationName    :
AssociationVersion : 2
ComplianceSeverity :
CreatedDate        : 3/12/2019 9:21:01 AM
DocumentVersion    :
MaxConcurrency     :
MaxErrors          :
Name               : AWS-GatherSoftwareInventory
OutputLocation     :
Parameters         : {}
ScheduleExpression :
Targets            : {InstanceIds}

AssociationId      : 123a45a0-c678-9012-3456-78901234db5e
AssociationName    : test-case-1234567890
AssociationVersion : 1
ComplianceSeverity :
```

```
CreatedDate        : 3/2/2019 8:53:29 AM
DocumentVersion    :
MaxConcurrency     :
MaxErrors          :
Name               : AWS-GatherSoftwareInventory
OutputLocation     :
Parameters         : {}
ScheduleExpression : rate(30minutes)
Targets            : {InstanceIds}
```

- Einzelheiten zur API finden Sie unter ListAssociationVersions AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden alle Versionen der angegebenen Zuordnung abgerufen.

```
Get-SSMAssociationVersionList -AssociationId 123a45a0-c678-9012-3456-78901234db5e
```

Ausgabe:

```
AssociationId      : 123a45a0-c678-9012-3456-78901234db5e
AssociationName    :
AssociationVersion : 2
ComplianceSeverity :
CreatedDate        : 3/12/2019 9:21:01 AM
DocumentVersion    :
MaxConcurrency     :
MaxErrors          :
Name               : AWS-GatherSoftwareInventory
OutputLocation     :
Parameters         : {}
ScheduleExpression :
Targets            : {InstanceIds}

AssociationId      : 123a45a0-c678-9012-3456-78901234db5e
AssociationName    : test-case-1234567890
AssociationVersion : 1
ComplianceSeverity :
CreatedDate        : 3/2/2019 8:53:29 AM
DocumentVersion    :
MaxConcurrency     :
MaxErrors          :
```

```
Name               : AWS-GatherSoftwareInventory
OutputLocation     :
Parameters         : {}
ScheduleExpression : rate(30minutes)
Targets            : {InstanceIds}
```

- Einzelheiten zur API finden Sie unter ListAssociationVersions AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`ListAssociations`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `ListAssociations` verwendet wird.

CLI

AWS CLI

Beispiel 1: So können Sie Ihre Zuordnungen für eine bestimmte Instance auflisten

Das folgende Beispiel für Listenzuordnungen listet alle Verknüpfungen mit dem Update auf AssociationName. SSMAgent

```
aws ssm list-associations /
    --association-filter-list "key=AssociationName,value=UpdateSSMAgent"
```

Ausgabe:

```
{
    "Associations": [
        {
            "Name": "AWS-UpdateSSMAgent",
            "InstanceId": "i-1234567890abcdef0",
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "Targets": [
                {
```

```
                    "Key": "InstanceIds",
                    "Values": [
                        "i-016648b75dd622dab"
                    ]
                }
            ],
            "Overview": {
                "Status": "Pending",
                "DetailedStatus": "Associated",
                "AssociationStatusAggregatedCount": {
                    "Pending": 1
                }
            },
            "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
            "AssociationName": "UpdateSSMAgent"
        }
    ]
}
```

Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager im -Systems-Manager-Benutzerhandbuch.

Beispiel 2: So können Sie Ihre Verknüpfungen für ein bestimmtes Dokument auflisten

Das folgende Beispiel listet alle Verknüpfungen für das angegebene Dokument auf.

```
aws ssm list-associations /
    --association-filter-list "key=Name,value=AWS-UpdateSSMAgent"
```

Ausgabe:

```
{
    "Associations": [
        {
            "Name": "AWS-UpdateSSMAgent",
            "InstanceId": "i-1234567890abcdef0",
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
```

```
                        "i-1234567890abcdef0"
                    ]
                }
            ],
            "LastExecutionDate": 1550505828.548,
            "Overview": {
                "Status": "Success",
                "DetailedStatus": "Success",
                "AssociationStatusAggregatedCount": {
                    "Success": 1
                }
            },
            "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
            "AssociationName": "UpdateSSMAgent"
        },
    {
            "Name": "AWS-UpdateSSMAgent",
            "InstanceId": "i-9876543210abcdef0",
            "AssociationId": "fbc07ef7-b985-4684-b82b-0123456789ab",
            "AssociationVersion": "1",
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-9876543210abcdef0"
                    ]
                }
            ],
            "LastExecutionDate": 1550507531.0,
            "Overview": {
                "Status": "Success",
                "AssociationStatusAggregatedCount": {
                    "Success": 1
                }
            }
        }
    ]
}
```


Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager im -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie ListAssociationsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dieses Beispiel listet Details zu den Zuordnungen für eine Instance auf. Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$filter1 = @{Key="InstanceId";Value=@("i-0000293ffd8c57862")}
Get-SSMAssociationList -AssociationFilterList $filter1
```

Ausgabe:

```
AssociationId      : d8617c07-2079-4c18-9847-1655fc2698b0
DocumentVersion    :
InstanceId         : i-0000293ffd8c57862
LastExecutionDate  : 2/20/2015 8:31:11 AM
Name               : AWS-UpdateSSMAgent
Overview           : Amazon.SimpleSystemsManagement.Model.AssociationOverview
ScheduleExpression :
Targets            : {InstanceIds}
```

Beispiel 2: In diesem Beispiel werden alle Verknüpfungen für ein Konfigurationsdokument aufgeführt. Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$filter2 = @{Key="Name";Value=@("AWS-UpdateSSMAgent")}
Get-SSMAssociationList -AssociationFilterList $filter2
```

Ausgabe:

```
AssociationId      : d8617c07-2079-4c18-9847-1655fc2698b0
DocumentVersion    :
InstanceId         : i-0000293ffd8c57862
LastExecutionDate  : 2/20/2015 8:31:11 AM
Name               : AWS-UpdateSSMAgent
Overview           : Amazon.SimpleSystemsManagement.Model.AssociationOverview
ScheduleExpression :
Targets            : {InstanceIds}
```

Beispiel 3: Bei PowerShell Version 2 müssen Sie New-Object verwenden, um jeden Filter zu erstellen.

```
$filter1 = New-Object Amazon.SimpleSystemsManagement.Model.AssociationFilter
$filter1.Key = "InstanceId"
$filter1.Value = "i-0000293ffd8c57862"

Get-SSMAssociationList -AssociationFilterList $filter1
```

Ausgabe:

```
AssociationId      : d8617c07-2079-4c18-9847-1655fc2698b0
DocumentVersion    :
InstanceId         : i-0000293ffd8c57862
LastExecutionDate  : 2/20/2015 8:31:11 AM
Name               : AWS-UpdateSSMAgent
Overview           : Amazon.SimpleSystemsManagement.Model.AssociationOverview
ScheduleExpression :
Targets            : {InstanceIds}
```

- Einzelheiten zur API finden Sie unter ListAssociations AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dieses Beispiel listet Details zu den Zuordnungen für eine Instance auf. Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$filter1 = @{Key="InstanceId";Value=@("i-0000293ffd8c57862")}
Get-SSMAssociationList -AssociationFilterList $filter1
```

Ausgabe:

```
AssociationId      : d8617c07-2079-4c18-9847-1655fc2698b0
DocumentVersion    :
InstanceId         : i-0000293ffd8c57862
LastExecutionDate  : 2/20/2015 8:31:11 AM
Name               : AWS-UpdateSSMAgent
Overview           : Amazon.SimpleSystemsManagement.Model.AssociationOverview
ScheduleExpression :
Targets            : {InstanceIds}
```

Beispiel 2: In diesem Beispiel werden alle Verknüpfungen für ein Konfigurationsdokument aufgeführt. Die in diesem Beispiel verwendete Syntax erfordert PowerShell Version 3 oder höher.

```
$filter2 = @{Key="Name";Value=@("AWS-UpdateSSMAgent")}
Get-SSMAssociationList -AssociationFilterList $filter2
```

Ausgabe:

```
AssociationId      : d8617c07-2079-4c18-9847-1655fc2698b0
DocumentVersion    :
InstanceId         : i-0000293ffd8c57862
LastExecutionDate  : 2/20/2015 8:31:11 AM
Name               : AWS-UpdateSSMAgent
Overview           : Amazon.SimpleSystemsManagement.Model.AssociationOverview
ScheduleExpression :
Targets            : {InstanceIds}
```

Beispiel 3: Bei PowerShell Version 2 müssen Sie New-Object verwenden, um jeden Filter zu erstellen.

```
$filter1 = New-Object Amazon.SimpleSystemsManagement.Model.AssociationFilter
$filter1.Key = "InstanceId"
$filter1.Value = "i-0000293ffd8c57862"

Get-SSMAssociationList -AssociationFilterList $filter1
```

Ausgabe:

```
AssociationId      : d8617c07-2079-4c18-9847-1655fc2698b0
DocumentVersion    :
InstanceId         : i-0000293ffd8c57862
LastExecutionDate  : 2/20/2015 8:31:11 AM
Name               : AWS-UpdateSSMAgent
Overview           : Amazon.SimpleSystemsManagement.Model.AssociationOverview
ScheduleExpression :
Targets            : {InstanceIds}
```

- Einzelheiten zur API finden Sie unter ListAssociations AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ListCommandInvocations`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `ListCommandInvocations` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Kennenlernen der Grundlagen

CLI

AWS CLI

So listen Sie die Aufrufe eines bestimmten Befehls auf

Das folgende `list-command-invocations`-Beispiel listet alle Aufrufe eines Befehls auf.

```
aws ssm list-command-invocations \
    --command-id "ef7fdfd8-9b57-4151-a15c-db9a12345678" \
    --details
```

Ausgabe:

```
{
    "CommandInvocations": [
        {
            "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",
            "InstanceId": "i-02573cafcfEXAMPLE",
            "InstanceName": "",
            "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
            "DocumentName": "AWS-UpdateSSMAgent",
            "DocumentVersion": "",
            "RequestedDateTime": 1582136283.089,
            "Status": "Success",
            "StatusDetails": "Success",
            "StandardOutputUrl": "",
            "StandardErrorUrl": "",
            "CommandPlugins": [
                {
                    "Name": "aws:updateSsmAgent",
                    "Status": "Success",
                    "StatusDetails": "Success",
```

```
                    "ResponseCode": 0,
                    "ResponseStartDateTime": 1582136283.419,
                    "ResponseFinishDateTime": 1582136283.51,
                    "Output": "Updating amazon-ssm-agent from 2.3.842.0 to latest
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-
east-2/ssm-agent-manifest.json\namazon-ssm-agent 2.3.842.0 has already been
 installed, update skipped\n",
                    "StandardOutputUrl": "",
                    "StandardErrorUrl": "",
                    "OutputS3Region": "us-east-2",
                    "OutputS3BucketName": "",
                    "OutputS3KeyPrefix": ""
                }
            ],
            "ServiceRole": "",
            "NotificationConfig": {
                "NotificationArn": "",
                "NotificationEvents": [],
                "NotificationType": ""
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        },
        {
            "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",
            "InstanceId": "i-0471e04240EXAMPLE",
            "InstanceName": "",
            "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
            "DocumentName": "AWS-UpdateSSMAgent",
            "DocumentVersion": "",
            "RequestedDateTime": 1582136283.02,
            "Status": "Success",
            "StatusDetails": "Success",
            "StandardOutputUrl": "",
            "StandardErrorUrl": "",
            "CommandPlugins": [
                {
                    "Name": "aws:updateSsmAgent",
                    "Status": "Success",
                    "StatusDetails": "Success",
                    "ResponseCode": 0,
```

```
                    "ResponseStartDateTime": 1582136283.812,
                    "ResponseFinishDateTime": 1582136295.031,
                    "Output": "Updating amazon-ssm-agent from 2.3.672.0 to
 latest\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-
ssm-us-east-2/ssm-agent-manifest.json\nSuccessfully downloaded https://s3.us-
east-2.amazonaws.com/amazon-ssm-us-east-2/amazon-ssm-agent-updater/2.3.842.0/
amazon-ssm-agent-updater-snap-amd64.tar.gz\nSuccessfully downloaded https://
s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/amazon-ssm-agent/2.3.672.0/
amazon-ssm-agent-snap-amd64.tar.gz\nSuccessfully downloaded https://s3.us-
east-2.amazonaws.com/amazon-ssm-us-east-2/amazon-ssm-agent/2.3.842.0/amazon-ssm-
agent-snap-amd64.tar.gz\nInitiating amazon-ssm-agent update to 2.3.842.0\namazon-
ssm-agent updated successfully to 2.3.842.0",
                    "StandardOutputUrl": "",
                    "StandardErrorUrl": "",
                    "OutputS3Region": "us-east-2",
                    "OutputS3BucketName": "",
                    "OutputS3KeyPrefix": "8bee3135-398c-4d31-99b6-e42d2EXAMPLE/
i-0471e04240EXAMPLE/awsupdateSsmAgent"
                }
            ],
            "ServiceRole": "",
            "NotificationConfig": {
                "NotificationArn": "",
                "NotificationEvents": [],
                "NotificationType": ""
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        }
    ]
}
```


Weitere Informationen finden Sie unter Befehlsstatus verstehen im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie ListCommandInvocationsin der AWS CLI Befehlsreferenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import { paginateListCommandInvocations, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * List SSM command invocations on an instance.
 * @param {{ instanceId: string }}
 */
export const main = async ({ instanceId }) => {
  const client = new SSMClient({});
  try {
    const listCommandInvocationsPaginated = [];
    // The paginate function is a wrapper around the base command.
    const paginator = paginateListCommandInvocations(
      { client },
      {
        InstanceId: instanceId,
      },
    );
    for await (const page of paginator) {
      listCommandInvocationsPaginated.push(...page.CommandInvocations);
    }
    console.log("Here is the list of command invocations:");
    console.log(listCommandInvocationsPaginated);
    return { CommandInvocations: listCommandInvocationsPaginated };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ValidationError") {
      console.warn(`${caught.message}. Did you provide a valid instance ID?`);
    }
    throw caught;
  }
};
```

- Einzelheiten zur API finden Sie ListCommandInvocationsin der AWS SDK für JavaScript API-Referenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden alle Aufrufe eines Befehls aufgeführt.

```
Get-SSMCommandInvocation -CommandId "b8eac879-0541-439d-94ec-47a80d554f44" -
Detail $true
```

Ausgabe:

```
CommandId          : b8eac879-0541-439d-94ec-47a80d554f44
CommandPlugins     : {aws:runShellScript}
Comment            : IP config
DocumentName       : AWS-RunShellScript
InstanceId         : i-0cb2b964d3e14fd9f
InstanceName       :
NotificationConfig : Amazon.SimpleSystemsManagement.Model.NotificationConfig
RequestedDateTime  : 2/22/2017 8:13:16 PM
ServiceRole        :
StandardErrorUrl   :
StandardOutputUrl  :
Status             : Success
StatusDetails      : Success
TraceOutput        :
```

Beispiel 2: In diesem Beispiel wird der Aufruf der Befehls-ID CommandPlugins e1eb2e3c-ed4c-5123-45c1-234f5612345f aufgeführt

```
Get-SSMCommandInvocation -CommandId e1eb2e3c-ed4c-5123-45c1-234f5612345f -Detail:
$true | Select-Object -ExpandProperty CommandPlugins
```

Ausgabe:

```
Name                  : aws:runPowerShellScript
Output                : Completed 17.7 KiB/17.7 KiB (40.1 KiB/s) with 1 file(s)
 remainingdownload: s3://dd-aess-r-ctmer/KUMO.png to ..\..\programdata\KUMO.png
```

```
                          kumo available

OutputS3BucketName     :
OutputS3KeyPrefix      :
OutputS3Region         : eu-west-1
ResponseCode           : 0
ResponseFinishDateTime : 4/3/2019 11:53:23 AM
ResponseStartDateTime  : 4/3/2019 11:53:21 AM
StandardErrorUrl       :
StandardOutputUrl      :
Status                 : Success
StatusDetails          : Success
```

- Einzelheiten AWS -Tools für PowerShell zur API ListCommandInvocationsfinden Sie unter Cmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden alle Aufrufe eines Befehls aufgeführt.

```
Get-SSMCommandInvocation -CommandId "b8eac879-0541-439d-94ec-47a80d554f44" -
Detail $true
```

Ausgabe:

```
CommandId          : b8eac879-0541-439d-94ec-47a80d554f44
CommandPlugins     : {aws:runShellScript}
Comment            : IP config
DocumentName       : AWS-RunShellScript
InstanceId         : i-0cb2b964d3e14fd9f
InstanceName       :
NotificationConfig : Amazon.SimpleSystemsManagement.Model.NotificationConfig
RequestedDateTime  : 2/22/2017 8:13:16 PM
ServiceRole        :
StandardErrorUrl   :
StandardOutputUrl  :
Status             : Success
StatusDetails      : Success
TraceOutput        :
```

Beispiel 2: In diesem Beispiel wird der Aufruf der Befehls-ID CommandPlugins e1eb2e3c-ed4c-5123-45c1-234f5612345f aufgeführt

```
Get-SSMCommandInvocation -CommandId e1eb2e3c-ed4c-5123-45c1-234f5612345f -Detail:
$true | Select-Object -ExpandProperty CommandPlugins
```

Ausgabe:

```
Name                   : aws:runPowerShellScript
Output                 : Completed 17.7 KiB/17.7 KiB (40.1 KiB/s) with 1 file(s)
 remainingdownload: s3://dd-aess-r-ctmer/KUMO.png to ..\..\programdata\KUMO.png
                          kumo available

OutputS3BucketName     :
OutputS3KeyPrefix      :
OutputS3Region         : eu-west-1
ResponseCode           : 0
ResponseFinishDateTime : 4/3/2019 11:53:23 AM
ResponseStartDateTime  : 4/3/2019 11:53:21 AM
StandardErrorUrl       :
StandardOutputUrl      :
Status                 : Success
StatusDetails          : Success
```

- Einzelheiten AWS -Tools für PowerShell zur API ListCommandInvocationsfinden Sie unter Cmdlet-Referenz (V5).

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class DocumentWrapper:
    """Encapsulates AWS Systems Manager Document actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
```

```
        """
        self.ssm_client = ssm_client
        self.name = None

    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def list_command_invocations(self, instance_id):
        """
        Lists the commands for an instance.

        :param instance_id: The ID of the instance.
        :return: The list of commands.
        """
        try:
            paginator = self.ssm_client.get_paginator("list_command_invocations")
            command_invocations = []
            for page in paginator.paginate(InstanceId=instance_id):
                command_invocations.extend(page["CommandInvocations"])
            num_of_commands = len(command_invocations)
            print(
                f"{num_of_commands} command invocation(s) found for instance
{instance_id}."
            )

            if num_of_commands > 10:
                print("Displaying the first 10 commands:")
                num_of_commands = 10
            date_format = "%A, %d %B %Y %I:%M%p"
            for command in command_invocations[:num_of_commands]:
                print(
                    f"   The time of command invocation is
{command['RequestedDateTime'].strftime(date_format)}"
                )
        except ClientError as err:
            logger.error(
                "Couldn't list commands for %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
```

```
            raise
```

- Einzelheiten zur API finden Sie ListCommandInvocationsin AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`ListCommands`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `ListCommands` verwendet wird.

CLI

AWS CLI

Beispiel 1: So erhalten Sie den Status eines bestimmten Befehls

Im folgenden Beispiel `list-commands` wird der Status des angegebenen Befehls abgerufen und angezeigt.

```
aws ssm list-commands \
    --command-id "0831e1a8-a1ac-4257-a1fd-c831bEXAMPLE"
```

Beispiel 2: So erhalten Sie den Status von Befehlen, die nach einem bestimmten Datum angefordert wurden

Im folgenden Beispiel `list-commands` werden die Details von Befehlen abgerufen, die nach dem angegebenen Datum angefordert wurden.

```
aws ssm list-commands \
    --filter "key=InvokedAfter,value=2020-02-01T00:00:00Z"
```

Beispiel 3: Um alle Befehle aufzulisten, die in einem AWS Konto angefordert wurden

Das folgende `list-commands` Beispiel listet alle Befehle auf, die von Benutzern im aktuellen AWS Konto und in der Region angefordert wurden.

```
aws ssm list-commands
```

Ausgabe:

```
{
    "Commands": [
        {
            "CommandId": "8bee3135-398c-4d31-99b6-e42d2EXAMPLE",
            "DocumentName": "AWS-UpdateSSMAgent",
            "DocumentVersion": "",
            "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
            "ExpiresAfter": "2020-02-19T11:28:02.500000-08:00",
            "Parameters": {},
            "InstanceIds": [
                "i-028ea792daEXAMPLE",
                "i-02feef8c46EXAMPLE",
                "i-038613f3f0EXAMPLE",
                "i-03a530a2d4EXAMPLE",
                "i-083b678d37EXAMPLE",
                "i-0dee81debaEXAMPLE"
            ],
            "Targets": [],
            "RequestedDateTime": "2020-02-19T10:18:02.500000-08:00",
            "Status": "Success",
            "StatusDetails": "Success",
            "OutputS3BucketName": "",
            "OutputS3KeyPrefix": "",
            "MaxConcurrency": "50",
            "MaxErrors": "100%",
            "TargetCount": 6,
            "CompletedCount": 6,
            "ErrorCount": 0,
            "DeliveryTimedOutCount": 0,
            "ServiceRole": "",
            "NotificationConfig": {
                "NotificationArn": "",
                "NotificationEvents": [],
                "NotificationType": ""
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
```

```
            }
        }
        {
            "CommandId": "e9ade581-c03d-476b-9b07-26667EXAMPLE",
            "DocumentName": "AWS-FindWindowsUpdates",
            "DocumentVersion": "1",
            "Comment": "",
            "ExpiresAfter": "2020-01-24T12:37:31.874000-08:00",
            "Parameters": {
                "KbArticleIds": [
                    ""
                ],
                "UpdateLevel": [
                    "All"
                ]
            },
            "InstanceIds": [],
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-00ec29b21eEXAMPLE",
                        "i-09911ddd90EXAMPLE"
                    ]
                }
            ],
            "RequestedDateTime": "2020-01-24T11:27:31.874000-08:00",
            "Status": "Success",
            "StatusDetails": "Success",
            "OutputS3BucketName": "my-us-east-2-bucket",
            "OutputS3KeyPrefix": "my-rc-output",
            "MaxConcurrency": "50",
            "MaxErrors": "0",
            "TargetCount": 2,
            "CompletedCount": 2,
            "ErrorCount": 0,
            "DeliveryTimedOutCount": 0,
            "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "NotificationConfig": {
                "NotificationArn": "arn:aws:sns:us-east-2:111222333444:my-us-
east-2-notification-arn",
                "NotificationEvents": [
                    "All"
```

```
                ],
                "NotificationType": "Invocation"
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        }
        {
            "CommandId": "d539b6c3-70e8-4853-80e5-0ce4fEXAMPLE",
            "DocumentName": "AWS-RunPatchBaseline",
            "DocumentVersion": "1",
            "Comment": "",
            "ExpiresAfter": "2020-01-24T12:21:04.350000-08:00",
            "Parameters": {
                "InstallOverrideList": [
                    ""
                ],
                "Operation": [
                    "Install"
                ],
                "RebootOption": [
                    "RebootIfNeeded"
                ],
                "SnapshotId": [
                    ""
                ]
            },
            "InstanceIds": [],
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-00ec29b21eEXAMPLE",
                        "i-09911ddd90EXAMPLE"
                    ]
                }
            ],
            "RequestedDateTime": "2020-01-24T11:11:04.350000-08:00",
            "Status": "Success",
            "StatusDetails": "Success",
            "OutputS3BucketName": "my-us-east-2-bucket",
            "OutputS3KeyPrefix": "my-rc-output",
            "MaxConcurrency": "50",
```

```
            "MaxErrors": "0",
            "TargetCount": 2,
            "CompletedCount": 2,
            "ErrorCount": 0,
            "DeliveryTimedOutCount": 0,
            "ServiceRole": "arn:aws:iam::111222333444:role/aws-service-role/
ssm.amazonaws.com/AWSServiceRoleForAmazonSSM",
            "NotificationConfig": {
                "NotificationArn": "arn:aws:sns:us-east-2:111222333444:my-us-
east-2-notification-arn",
                "NotificationEvents": [
                    "All"
                ],
                "NotificationType": "Invocation"
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        }
    ]
}
```

Weitere Informationen finden Sie unter Run-Befehl mithilfe von Systems Manager in der AWS - Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie ListCommandsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden alle angeforderten Befehle aufgeführt.

```
Get-SSMCommand
```

Ausgabe:

```
CommandId          : 4b75a163-d39a-4d97-87c9-98ae52c6be35
Comment            : Apply association with id at update time: 4cc73e42-
d5ae-4879-84f8-57e09c0efcd0
CompletedCount     : 1
```

```
DocumentName       : AWS-RefreshAssociation
ErrorCount         : 0
ExpiresAfter       : 2/24/2017 3:19:08 AM
InstanceIds        : {i-0cb2b964d3e14fd9f}
MaxConcurrency     : 50
MaxErrors          : 0
NotificationConfig : Amazon.SimpleSystemsManagement.Model.NotificationConfig
OutputS3BucketName :
OutputS3KeyPrefix  :
OutputS3Region     :
Parameters         : {[associationIds,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
RequestedDateTime  : 2/24/2017 3:18:08 AM
ServiceRole        :
Status             : Success
StatusDetails      : Success
TargetCount        : 1
Targets            : {}
```

Beispiel 2: In diesem Beispiel wird der Status eines bestimmten Befehls abgerufen.

```
Get-SSMCommand -CommandId "4b75a163-d39a-4d97-87c9-98ae52c6be35"
```

Beispiel 3: In diesem Beispiel werden alle SSM-Befehle abgerufen, die nach dem 2019-04-01T 00:00:00 Z aufgerufen wurden

```
Get-SSMCommand -Filter @{Key="InvokedAfter";Value="2019-04-01T00:00:00Z"} |
 Select-Object CommandId, DocumentName, Status, RequestedDateTime | Sort-Object -
Property RequestedDateTime -Descending
```

Ausgabe:

```
CommandId                            DocumentName               Status
 RequestedDateTime
---------                            ------------               ------
 -----------------
edb1b23e-456a-7adb-aef8-90e-012ac34f AWS-RunPowerShellScript    Cancelled
 4/16/2019 5:45:23 AM
1a2dc3fb-4567-890d-a1ad-234b5d6bc7d9 AWS-ConfigureAWSPackage    Success
 4/6/2019 9:19:42 AM
12c3456c-7e90-4f12-1232-1234f5b67893 KT-Retrieve-Cloud-Type-Win Failed
 4/2/2019 4:13:07 AM
```

```
fe123b45-240c-4123-a2b3-234bdd567ecf AWS-RunInspecChecks        Failed
 4/1/2019 2:27:31 PM
1eb23aa4-567d-4123-12a3-4c1c2ab34561 AWS-RunPowerShellScript    Success
 4/1/2019 1:05:55 PM
1c2f3bb4-ee12-4bc1-1a23-12345eea123e AWS-RunInspecChecks        Failed
 4/1/2019 11:13:09 AM
```

- Einzelheiten zur API finden Sie unter ListCommands AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden alle angeforderten Befehle aufgeführt.

```
Get-SSMCommand
```

Ausgabe:

```
CommandId          : 4b75a163-d39a-4d97-87c9-98ae52c6be35
Comment            : Apply association with id at update time: 4cc73e42-
d5ae-4879-84f8-57e09c0efcd0
CompletedCount     : 1
DocumentName       : AWS-RefreshAssociation
ErrorCount         : 0
ExpiresAfter       : 2/24/2017 3:19:08 AM
InstanceIds        : {i-0cb2b964d3e14fd9f}
MaxConcurrency     : 50
MaxErrors          : 0
NotificationConfig : Amazon.SimpleSystemsManagement.Model.NotificationConfig
OutputS3BucketName :
OutputS3KeyPrefix  :
OutputS3Region     :
Parameters         : {[associationIds,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
RequestedDateTime  : 2/24/2017 3:18:08 AM
ServiceRole        :
Status             : Success
StatusDetails      : Success
TargetCount        : 1
Targets            : {}
```

Beispiel 2: In diesem Beispiel wird der Status eines bestimmten Befehls abgerufen.

```
Get-SSMCommand -CommandId "4b75a163-d39a-4d97-87c9-98ae52c6be35"
```

Beispiel 3: In diesem Beispiel werden alle SSM-Befehle abgerufen, die nach dem 2019-04-01T 00:00:00 Z aufgerufen wurden

```
Get-SSMCommand -Filter @{Key="InvokedAfter";Value="2019-04-01T00:00:00Z"} |
 Select-Object CommandId, DocumentName, Status, RequestedDateTime | Sort-Object -
Property RequestedDateTime -Descending
```

Ausgabe:

```
CommandId                            DocumentName               Status
 RequestedDateTime
---------                            ------------               ------
 -----------------
edb1b23e-456a-7adb-aef8-90e-012ac34f AWS-RunPowerShellScript    Cancelled
 4/16/2019 5:45:23 AM
1a2dc3fb-4567-890d-a1ad-234b5d6bc7d9 AWS-ConfigureAWSPackage    Success
 4/6/2019 9:19:42 AM
12c3456c-7e90-4f12-1232-1234f5b67893 KT-Retrieve-Cloud-Type-Win Failed
 4/2/2019 4:13:07 AM
fe123b45-240c-4123-a2b3-234bdd567ecf AWS-RunInspecChecks        Failed
 4/1/2019 2:27:31 PM
1eb23aa4-567d-4123-12a3-4c1c2ab34561 AWS-RunPowerShellScript    Success
 4/1/2019 1:05:55 PM
1c2f3bb4-ee12-4bc1-1a23-12345eea123e AWS-RunInspecChecks        Failed
 4/1/2019 11:13:09 AM
```

- Einzelheiten zur API finden Sie unter ListCommands AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`ListComplianceItems`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `ListComplianceItems` verwendet wird.

CLI

AWS CLI

So listen Sie Complianceelemente für eine bestimmte Instance auf

In diesem Beispiel werden alle Konformitätselemente für die angegebene Instance aufgeführt.

Befehl:

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-types "ManagedInstance"
```

Ausgabe:

```
{
  "ComplianceItems": [
      {
          "ComplianceType": "Association",
          "ResourceType": "ManagedInstance",
          "ResourceId": "i-1234567890abcdef0",
          "Id": "8dfe3659-4309-493a-8755-0123456789ab",
          "Title": "",
          "Status": "COMPLIANT",
          "Severity": "UNSPECIFIED",
          "ExecutionSummary": {
              "ExecutionTime": 1550408470.0
          },
          "Details": {
              "DocumentName": "AWS-GatherSoftwareInventory",
              "DocumentVersion": "1"
          }
      },
      {
          "ComplianceType": "Association",
          "ResourceType": "ManagedInstance",
          "ResourceId": "i-1234567890abcdef0",
          "Id": "e4c2ed6d-516f-41aa-aa2a-0123456789ab",
          "Title": "",
          "Status": "COMPLIANT",
          "Severity": "UNSPECIFIED",
          "ExecutionSummary": {
              "ExecutionTime": 1550508475.0
```

```
            },
            "Details": {
                "DocumentName": "AWS-UpdateSSMAgent",
                "DocumentVersion": "1"
            }
        },
                ...
    ],
    "NextToken": "--token string truncated--"
}
```

So listen Sie Complianceelemente für eine bestimmte Instance und Zuordnungs-ID auf

In diesem Beispiel werden alle Konformitätselemente für die angegebene Instance und Zuordnungs-ID aufgeführt.

Befehl:

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance" --
filters "Key=ComplianceType,Values=Association,Type=EQUAL" "Key=Id,Values=e4c2ed6d-516f-4
aa2a-0123456789ab,Type=EQUAL"
```

So listen Sie Compliance-Elemente für eine Instance nach einem bestimmten Datum und einer bestimmten Uhrzeit auf

In diesem Beispiel werden alle Compliance-Elemente für eine Instance nach dem angegebenen Datum und der angegebenen Uhrzeit aufgeführt.

Befehl:

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance" --
filters "Key=ExecutionTime,Values=2019-02-18T16:00:00Z,Type=GREATER_THAN"
```

- Einzelheiten zur API finden Sie ListComplianceItemsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird die Liste der Compliance-Elemente für die angegebene Ressourcen-ID und den angegebenen Ressourcentyp aufgeführt, wobei nach dem Compliance-Typ „Association“ gefiltert wird

```
Get-SSMComplianceItemList -ResourceId i-1a2caf345f67d0dc2 -ResourceType
 ManagedInstance -Filter @{Key="ComplianceType";Values="Association"}
```

Ausgabe:

```
ComplianceType   : Association
Details          : {[DocumentName, AWS-GatherSoftwareInventory],
 [DocumentVersion, 1]}
ExecutionSummary :
 Amazon.SimpleSystemsManagement.Model.ComplianceExecutionSummary
Id               : 123a45a1-c234-1234-1245-67891236db4e
ResourceId       : i-1a2caf345f67d0dc2
ResourceType     : ManagedInstance
Severity         : UNSPECIFIED
Status           : COMPLIANT
Title            :
```

- Einzelheiten zur API finden Sie unter ListComplianceItems AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird die Liste der Compliance-Elemente für die angegebene Ressourcen-ID und den angegebenen Ressourcentyp aufgeführt, wobei nach dem Compliance-Typ „Association“ gefiltert wird

```
Get-SSMComplianceItemList -ResourceId i-1a2caf345f67d0dc2 -ResourceType
 ManagedInstance -Filter @{Key="ComplianceType";Values="Association"}
```

Ausgabe:

```
ComplianceType   : Association
Details          : {[DocumentName, AWS-GatherSoftwareInventory],
 [DocumentVersion, 1]}
```

```
ExecutionSummary :
 Amazon.SimpleSystemsManagement.Model.ComplianceExecutionSummary
Id               : 123a45a1-c234-1234-1245-67891236db4e
ResourceId       : i-1a2caf345f67d0dc2
ResourceType     : ManagedInstance
Severity         : UNSPECIFIED
Status           : COMPLIANT
Title            :
```

- Einzelheiten zur API finden Sie unter ListComplianceItems AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`ListComplianceSummaries`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `ListComplianceSummaries` verwendet wird.

CLI

AWS CLI

So können Sie Konformitätszusammenfassungen für alle Compliance-Typen auflisten

In diesem Beispiel werden Konformitätszusammenfassungen für alle Compliance-Typen in Ihrem Konto aufgeführt.

Befehl:

```
aws ssm list-compliance-summaries
```

Ausgabe:

```
{
  "ComplianceSummaryItems": [
      {
          "ComplianceType": "Association",
          "CompliantSummary": {
              "CompliantCount": 2,
              "SeveritySummary": {
```

```
                "CriticalCount": 0,
                "HighCount": 0,
                "MediumCount": 0,
                "LowCount": 0,
                "InformationalCount": 0,
                "UnspecifiedCount": 2
            }
        },
        "NonCompliantSummary": {
            "NonCompliantCount": 0,
            "SeveritySummary": {
                "CriticalCount": 0,
                "HighCount": 0,
                "MediumCount": 0,
                "LowCount": 0,
                "InformationalCount": 0,
                "UnspecifiedCount": 0
            }
        }
    },
    {
        "ComplianceType": "Patch",
        "CompliantSummary": {
            "CompliantCount": 1,
            "SeveritySummary": {
                "CriticalCount": 0,
                "HighCount": 0,
                "MediumCount": 0,
                "LowCount": 0,
                "InformationalCount": 0,
                "UnspecifiedCount": 1
            }
        },
        "NonCompliantSummary": {
            "NonCompliantCount": 1,
            "SeveritySummary": {
                "CriticalCount": 1,
                "HighCount": 0,
                "MediumCount": 0,
                "LowCount": 0,
                "InformationalCount": 0,
                "UnspecifiedCount": 0
            }
        }
```

```
      },
                ...
  ],
  "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

So können Sie Compliance-Zusammenfassungen für einen bestimmten Konformitätstyp auflisten

In diesem Beispiel wird die Konformitätszusammenfassung für den Konformitätstyp Patch aufgelistet.

Befehl:

```
aws ssm list-compliance-summaries --
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- Einzelheiten zur API finden Sie ListComplianceSummariesin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dieses Beispiel gibt eine Sammelzahl konformer und nicht konformer Ressourcen für alle Konformitätstypen zurück.

```
Get-SSMComplianceSummaryList
```

Ausgabe:

```
ComplianceType CompliantSummary
 NonCompliantSummary
-------------- ----------------
 -------------------
FleetTotal     Amazon.SimpleSystemsManagement.Model.CompliantSummary
 Amazon.SimpleSystemsManagement.Model.NonCompliantSummary
Association    Amazon.SimpleSystemsManagement.Model.CompliantSummary
 Amazon.SimpleSystemsManagement.Model.NonCompliantSummary
Custom:InSpec  Amazon.SimpleSystemsManagement.Model.CompliantSummary
 Amazon.SimpleSystemsManagement.Model.NonCompliantSummary
```

```
Patch          Amazon.SimpleSystemsManagement.Model.CompliantSummary
 Amazon.SimpleSystemsManagement.Model.NonCompliantSummary
```

- Einzelheiten zur API finden Sie unter ListComplianceSummaries AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dieses Beispiel gibt eine Sammelzahl konformer und nicht konformer Ressourcen für alle Konformitätstypen zurück.

```
Get-SSMComplianceSummaryList
```

Ausgabe:

```
ComplianceType CompliantSummary
 NonCompliantSummary
-------------- ---------------
 -------------------
FleetTotal     Amazon.SimpleSystemsManagement.Model.CompliantSummary
 Amazon.SimpleSystemsManagement.Model.NonCompliantSummary
Association    Amazon.SimpleSystemsManagement.Model.CompliantSummary
 Amazon.SimpleSystemsManagement.Model.NonCompliantSummary
Custom:InSpec  Amazon.SimpleSystemsManagement.Model.CompliantSummary
 Amazon.SimpleSystemsManagement.Model.NonCompliantSummary
Patch          Amazon.SimpleSystemsManagement.Model.CompliantSummary
 Amazon.SimpleSystemsManagement.Model.NonCompliantSummary
```

- Einzelheiten zur API finden Sie unter ListComplianceSummaries AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`ListDocumentVersions`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `ListDocumentVersions` verwendet wird.

CLI

AWS CLI

Um Dokumentversionen aufzulisten

Das folgende `list-document-versions`-Beispiel listet alle Versionen eines Systems-Manager-Dokuments auf.

```
aws ssm list-document-versions \
    --name "Example"
```

Ausgabe:

```
{
    "DocumentVersions": [
        {
            "Name": "Example",
            "DocumentVersion": "1",
            "CreatedDate": 1583257938.266,
            "IsDefaultVersion": true,
            "DocumentFormat": "YAML",
            "Status": "Active"
        }
    ]
}
```

Weitere Informationen finden Sie unter Senden von Befehlen, die den Dokumentversionsparameter verwenden im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie ListDocumentVersionsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden alle Versionen eines Dokuments aufgeführt.

```
Get-SSMDocumentVersionList -Name "AWS-UpdateSSMAgent"
```

Ausgabe:

```
CreatedDate       : 6/1/2021 5:19:10 PM
DocumentFormat    : JSON
DocumentVersion   : 1
IsDefaultVersion  : True
Name              : AWS-UpdateSSMAgent
Status            : Active
```

- Einzelheiten zur API finden Sie unter ListDocumentVersions AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden alle Versionen eines Dokuments aufgeführt.

```
Get-SSMDocumentVersionList -Name "AWS-UpdateSSMAgent"
```

Ausgabe:

```
CreatedDate       : 6/1/2021 5:19:10 PM
DocumentFormat    : JSON
DocumentVersion   : 1
IsDefaultVersion  : True
Name              : AWS-UpdateSSMAgent
Status            : Active
```

- Einzelheiten zur API finden Sie unter ListDocumentVersions AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`ListDocuments`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `ListDocuments` verwendet wird.

CLI

AWS CLI

Beispiel 1: Dokumente auflisten

Das folgende `list-documents`-Beispiel listet Dokumente auf, die dem anfragenden Konto gehören und mit dem benutzerdefinierten Tag versehen sind.

```
aws ssm list-documents \
    --filters Key=Owner,Values=Self Key=tag:DocUse,Values=Testing
```

Ausgabe:

```
{
    "DocumentIdentifiers": [
        {
            "Name": "Example",
            "Owner": "29884EXAMPLE",
            "PlatformTypes": [
                "Windows",
                "Linux"
            ],
            "DocumentVersion": "1",
            "DocumentType": "Automation",
            "SchemaVersion": "0.3",
            "DocumentFormat": "YAML",
            "Tags": [
                {
                    "Key": "DocUse",
                    "Value": "Testing"
                }
            ]
        }
    ]
}
```

Weitere Informationen finden Sie unter AWS -Systems-Manager-Dokumenten im AWS -Systems-Manager-Benutzerhandbuch.

Beispiel 2: Geteilte Dokumente auflisten

Das folgende `list-documents` Beispiel listet gemeinsam genutzte Dokumente auf, einschließlich privater geteilter Dokumente, die nicht Eigentum von sind AWS.

```
aws ssm list-documents \
    --filters Key=Name,Values=sharedDocNamePrefix  Key=Owner,Values=Private
```

Ausgabe:

```
{
    "DocumentIdentifiers": [
        {
            "Name": "Example",
            "Owner": "12345EXAMPLE",
            "PlatformTypes": [
                "Windows",
                "Linux"
            ],
            "DocumentVersion": "1",
            "DocumentType": "Command",
            "SchemaVersion": "0.3",
            "DocumentFormat": "YAML",
            "Tags": []
        }
    ]
}
```

Weitere Informationen finden Sie unter AWS -Systems-Manager-Dokumenten im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie ListDocumentsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Listet alle Konfigurationsdokumente in Ihrem Konto auf.

```
Get-SSMDocumentList
```

Ausgabe:

```
DocumentType    : Command
DocumentVersion : 1
Name            : AWS-ApplyPatchBaseline
Owner           : Amazon
PlatformTypes   : {Windows}
SchemaVersion   : 1.2
```

```
DocumentType    : Command
DocumentVersion : 1
Name            : AWS-ConfigureAWSPackage
Owner           : Amazon
PlatformTypes   : {Windows, Linux}
SchemaVersion   : 2.0

DocumentType    : Command
DocumentVersion : 1
Name            : AWS-ConfigureCloudWatch
Owner           : Amazon
PlatformTypes   : {Windows}
SchemaVersion   : 1.2
...
```

Beispiel 2: In diesem Beispiel werden alle Automatisierungsdokumente abgerufen, deren Name mit „Platform“ übereinstimmt

```
Get-SSMDocumentList -DocumentFilterList @{Key="DocumentType";Value="Automation"}
 | Where-Object Name -Match "Platform"
```

Ausgabe:

```
DocumentFormat  : JSON
DocumentType    : Automation
DocumentVersion : 7
Name            : KT-Get-Platform
Owner           : 987654123456
PlatformTypes   : {Windows, Linux}
SchemaVersion   : 0.3
Tags            : {}
TargetType      :
VersionName     :
```

- Einzelheiten zur API finden Sie unter ListDocuments AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Listet alle Konfigurationsdokumente in Ihrem Konto auf.

```
Get-SSMDocumentList
```

Ausgabe:

```
DocumentType    : Command
DocumentVersion : 1
Name            : AWS-ApplyPatchBaseline
Owner           : Amazon
PlatformTypes   : {Windows}
SchemaVersion   : 1.2

DocumentType    : Command
DocumentVersion : 1
Name            : AWS-ConfigureAWSPackage
Owner           : Amazon
PlatformTypes   : {Windows, Linux}
SchemaVersion   : 2.0

DocumentType    : Command
DocumentVersion : 1
Name            : AWS-ConfigureCloudWatch
Owner           : Amazon
PlatformTypes   : {Windows}
SchemaVersion   : 1.2
...
```

Beispiel 2: In diesem Beispiel werden alle Automatisierungsdokumente abgerufen, deren Name mit „Platform“ übereinstimmt

```
Get-SSMDocumentList -DocumentFilterList @{Key="DocumentType";Value="Automation"}
 | Where-Object Name -Match "Platform"
```

Ausgabe:

```
DocumentFormat  : JSON
DocumentType    : Automation
DocumentVersion : 7
Name            : KT-Get-Platform
Owner           : 987654123456
PlatformTypes   : {Windows, Linux}
SchemaVersion   : 0.3
Tags            : {}
TargetType      :
VersionName     :
```

- Einzelheiten zur API finden Sie unter ListDocuments AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`ListInventoryEntries`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `ListInventoryEntries` verwendet wird.

CLI

AWS CLI

Beispiel 1: So zeigen Sie bestimmte Inventartypeinträge für eine Instance an

Das folgende `list-inventory-entries` Beispiel listet die Inventareinträge für den AWS Inventartyp:Application für eine bestimmte Instanz auf.

```
aws ssm list-inventory-entries \
    --instance-id "i-1234567890abcdef0" \
    --type-name "AWS:Application"
```

Ausgabe:

```
{
  "TypeName": "AWS:Application",
  "InstanceId": "i-1234567890abcdef0",
  "SchemaVersion": "1.1",
  "CaptureTime": "2019-02-15T12:17:55Z",
  "Entries": [
    {
      "Architecture": "i386",
      "Name": "Amazon SSM Agent",
      "PackageId": "{88a60be2-89a1-4df8-812a-80863c2a2b68}",
      "Publisher": "Amazon Web Services",
      "Version": "2.3.274.0"
    },
    {
      "Architecture": "x86_64",
```

```
      "InstalledTime": "2018-05-03T13:42:34Z",
      "Name": "AmazonCloudWatchAgent",
      "Publisher": "",
      "Version": "1.200442.0"
    }
  ]
}
```

Beispiel 2: So zeigen Sie benutzerdefinierte Inventareinträge an, die einer Instance zugewiesen sind

Das folgende `list-inventory-entries`-Beispiel listet einen benutzerdefinierten Inventareintrag auf, der einer Instance zugewiesen ist.

```
aws ssm list-inventory-entries \
    --instance-id "i-1234567890abcdef0" \
    --type-name "Custom:RackInfo"
```

Ausgabe:

```
{
  "TypeName": "Custom:RackInfo",
  "InstanceId": "i-1234567890abcdef0",
  "SchemaVersion": "1.0",
  "CaptureTime": "2021-05-22T10:01:01Z",
  "Entries": [
    {
      "RackLocation": "Bay B/Row C/Rack D/Shelf E"
    }
  ]
}
```

- Einzelheiten zur API finden Sie unter ListInventoryEntries AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden alle benutzerdefinierten Inventareinträge für eine Instance aufgeführt.

```
Get-SSMInventoryEntriesList -InstanceId "i-0cb2b964d3e14fd9f" -TypeName
 "Custom:RackInfo"
```

Ausgabe:

```
CaptureTime   : 2016-08-22T10:01:01Z
Entries       :
 {Amazon.Runtime.Internal.Util.AlwaysSendDictionary`2[System.String,System.String]}
InstanceId    : i-0cb2b964d3e14fd9f
NextToken     :
SchemaVersion : 1.0
TypeName      : Custom:RackInfo
```

Beispiel 2: In diesem Beispiel werden die Details aufgeführt.

```
(Get-SSMInventoryEntriesList -InstanceId "i-0cb2b964d3e14fd9f" -TypeName
 "Custom:RackInfo").Entries
```

Ausgabe:

```
Key          Value
---          -----
RackLocation Bay B/Row C/Rack D/Shelf E
```

- Einzelheiten zur API finden Sie unter ListInventoryEntries AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden alle benutzerdefinierten Inventareinträge für eine Instance aufgeführt.

```
Get-SSMInventoryEntriesList -InstanceId "i-0cb2b964d3e14fd9f" -TypeName
 "Custom:RackInfo"
```

Ausgabe:

```
CaptureTime   : 2016-08-22T10:01:01Z
Entries       :
 {Amazon.Runtime.Internal.Util.AlwaysSendDictionary`2[System.String,System.String]}
```

```
InstanceId    : i-0cb2b964d3e14fd9f
NextToken     :
SchemaVersion : 1.0
TypeName      : Custom:RackInfo
```

Beispiel 2: In diesem Beispiel werden die Details aufgeführt.

```
(Get-SSMInventoryEntriesList -InstanceId "i-0cb2b964d3e14fd9f" -TypeName
 "Custom:RackInfo").Entries
```

Ausgabe:

```
Key          Value
---          -----
RackLocation Bay B/Row C/Rack D/Shelf E
```

- Einzelheiten zur API finden Sie unter ListInventoryEntries AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`ListResourceComplianceSummaries`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `ListResourceComplianceSummaries` verwendet wird.

CLI

AWS CLI

So listen Sie die Anzahl der Compliance-Anforderungen auf Ressourcenebene auf

In diesem Beispiel werden die Compliance-Zusammenfassungszahlen auf Ressourcenebene aufgelistet.

Befehl:

```
aws ssm list-resource-compliance-summaries
```

Ausgabe:


```
{
  "ResourceComplianceSummaryItems": [
      {
          "ComplianceType": "Association",
          "ResourceType": "ManagedInstance",
          "ResourceId": "i-1234567890abcdef0",
          "Status": "COMPLIANT",
          "OverallSeverity": "UNSPECIFIED",
          "ExecutionSummary": {
              "ExecutionTime": 1550509273.0
          },
          "CompliantSummary": {
              "CompliantCount": 2,
              "SeveritySummary": {
                  "CriticalCount": 0,
                  "HighCount": 0,
                  "MediumCount": 0,
                  "LowCount": 0,
                  "InformationalCount": 0,
                  "UnspecifiedCount": 2
              }
          },
          "NonCompliantSummary": {
              "NonCompliantCount": 0,
              "SeveritySummary": {
                  "CriticalCount": 0,
                  "HighCount": 0,
                  "MediumCount": 0,
                  "LowCount": 0,
                  "InformationalCount": 0,
                  "UnspecifiedCount": 0
              }
          }
      },
      {
          "ComplianceType": "Patch",
          "ResourceType": "ManagedInstance",
          "ResourceId": "i-9876543210abcdef0",
          "Status": "COMPLIANT",
          "OverallSeverity": "UNSPECIFIED",
          "ExecutionSummary": {
              "ExecutionTime": 1550248550.0,
```

```
                "ExecutionId": "7abb6378-a4a5-4f10-8312-0123456789ab",
                "ExecutionType": "Command"
            },
            "CompliantSummary": {
                "CompliantCount": 397,
                "SeveritySummary": {
                    "CriticalCount": 0,
                    "HighCount": 0,
                    "MediumCount": 0,
                    "LowCount": 0,
                    "InformationalCount": 0,
                    "UnspecifiedCount": 397
                }
            },
            "NonCompliantSummary": {
                "NonCompliantCount": 0,
                "SeveritySummary": {
                    "CriticalCount": 0,
                    "HighCount": 0,
                    "MediumCount": 0,
                    "LowCount": 0,
                    "InformationalCount": 0,
                    "UnspecifiedCount": 0
                }
            }
        }
    ],
    "NextToken": "--token string truncated--"
}
```

So listen Sie die Compliance-Zusammenfassungen auf Ressourcenebene für einen bestimmten Konformitätstyp auf

In diesem Beispiel werden Konformitätszusammenfassungen auf Ressourcenebene für den Kompatibilitätstyp Patch aufgeführt.

Befehl:

```
aws ssm list-resource-compliance-summaries --
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- Einzelheiten zur API finden Sie ListResourceComplianceSummariesin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird eine Zusammenfassung der Anzahl auf Ressourcenebene abgerufen. Die Zusammenfassung enthält Informationen über den Status „konform“ und „nicht konform“ sowie detaillierte Angaben zum Schweregrad von Produkten, die „Windows10“ entsprechen. Da der MaxResult Standardwert 100 ist, wenn der Parameter nicht angegeben ist und dieser Wert nicht gültig ist, wird der MaxResult Parameter hinzugefügt und der Wert auf 50 gesetzt.

```
$FilterValues = @{
  "Key"="Product"
        "Type"="EQUAL"
        "Values"="Windows10"
}
        Get-SSMResourceComplianceSummaryList -Filter $FilterValues -MaxResult 50
```

- Einzelheiten zur API finden Sie unter ListResourceComplianceSummaries AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird eine Zusammenfassung der Anzahl auf Ressourcenebene abgerufen. Die Zusammenfassung enthält Informationen über den Status „konform“ und „nicht konform“ sowie detaillierte Angaben zum Schweregrad von Produkten, die „Windows10“ entsprechen. Da der MaxResult Standardwert 100 ist, wenn der Parameter nicht angegeben ist und dieser Wert nicht gültig ist, wird der MaxResult Parameter hinzugefügt und der Wert auf 50 gesetzt.

```
$FilterValues = @{
  "Key"="Product"
        "Type"="EQUAL"
        "Values"="Windows10"
}
        Get-SSMResourceComplianceSummaryList -Filter $FilterValues -MaxResult 50
```

- Einzelheiten zur API finden Sie unter ListResourceComplianceSummaries AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`ListTagsForResource`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `ListTagsForResource` verwendet wird.

CLI

AWS CLI

So listen Sie die auf eine Patch-Baseline angewendeten Tags auf

Das folgende `list-tags-for-resource`-Beispiel listet die Tags für eine Patch-Baseline auf.

```
aws ssm list-tags-for-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0123456789abcdef0"
```

Ausgabe:

```
{
    "TagList": [
        {
            "Key": "Environment",
            "Value": "Production"
        },
        {
            "Key": "Region",
            "Value": "EMEA"
        }
    ]
}
```

Weitere Informationen finden Sie unter AWS Ressourcen mit Tags versehen in der AWS allgemeinen Referenz.

- Einzelheiten zur API finden Sie ListTagsForResourcein der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dieses Beispiel listet die Tags für ein Wartungsfenster auf.

```
Get-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow"
```

Ausgabe:

```
Key   Value
---   -----
Stack Production
```

- Einzelheiten zur API finden Sie unter ListTagsForResource AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dieses Beispiel listet die Tags für ein Wartungsfenster auf.

```
Get-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow"
```

Ausgabe:

```
Key   Value
---   -----
Stack Production
```

- Einzelheiten zur API finden Sie unter ListTagsForResource AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`ModifyDocumentPermission`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `ModifyDocumentPermission` verwendet wird.

CLI

AWS CLI

So können Sie Dokumentberechtigungen ändern

Im folgenden `modify-document-permission`-Beispiel wird ein Systems-Manager-Dokument öffentlich freigegeben.

```
aws ssm modify-document-permission \
    --name "Example" \
    --permission-type "Share" \
    --account-ids-to-add "All"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Systems-Manager-Dokument teilen im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie ModifyDocumentPermissionin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden allen Konten für ein Dokument „Teilen“-Berechtigungen hinzugefügt. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Edit-SSMDocumentPermission -Name "RunShellScript" -PermissionType "Share" -
AccountIdsToAdd all
```

Beispiel 2: Dieses Beispiel fügt einem bestimmten Konto die Berechtigung zum „Freigeben“ eines Dokuments hinzu. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Edit-SSMDocumentPermission -Name "RunShellScriptNew" -PermissionType "Share" -
AccountIdsToAdd "123456789012"
```

- Einzelheiten zur API finden Sie unter ModifyDocumentPermission AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden allen Konten für ein Dokument „Teilen“-Berechtigungen hinzugefügt. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Edit-SSMDocumentPermission -Name "RunShellScript" -PermissionType "Share" -
AccountIdsToAdd all
```

Beispiel 2: Dieses Beispiel fügt einem bestimmten Konto die Berechtigung zum „Freigeben“ eines Dokuments hinzu. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Edit-SSMDocumentPermission -Name "RunShellScriptNew" -PermissionType "Share" -
AccountIdsToAdd "123456789012"
```

- Einzelheiten zur API finden Sie unter ModifyDocumentPermission AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`PutComplianceItems`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `PutComplianceItems` verwendet wird.

CLI

AWS CLI

Um einen Compliance-Typ und Compliance-Details zu einer bestimmten Instance zu registrieren

In diesem Beispiel wird der Konformitätstyp `Custom:AVCheck` für die angegebene verwaltete Instance registriert. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

Befehl:

```
aws ssm put-compliance-items --resource-id "i-1234567890abcdef0" --
resource-type "ManagedInstance" --compliance-type "Custom:AVCheck"
 --execution-summary "ExecutionTime=2019-02-18T16:00:00Z" --
items "Id=Version2.0,Title=ScanHost,Severity=CRITICAL,Status=COMPLIANT"
```

- Einzelheiten zur API finden Sie PutComplianceItemsin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird ein benutzerdefiniertes Compliance-Element für die angegebene verwaltete Instance geschrieben

```
$item = [Amazon.SimpleSystemsManagement.Model.ComplianceItemEntry]::new()
$item.Id = "07Jun2019-3"
$item.Severity="LOW"
$item.Status="COMPLIANT"
$item.Title="Fin-test-1 - custom"
Write-SSMComplianceItem -ResourceId mi-012dcb3ecea45b678 -ComplianceType
 Custom:VSSCompliant2 -ResourceType ManagedInstance -Item $item -
ExecutionSummary_ExecutionTime "07-Jun-2019"
```

- Einzelheiten zur API finden Sie unter PutComplianceItems AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird ein benutzerdefiniertes Compliance-Element für die angegebene verwaltete Instance geschrieben

```
$item = [Amazon.SimpleSystemsManagement.Model.ComplianceItemEntry]::new()
$item.Id = "07Jun2019-3"
$item.Severity="LOW"
$item.Status="COMPLIANT"
$item.Title="Fin-test-1 - custom"
Write-SSMComplianceItem -ResourceId mi-012dcb3ecea45b678 -ComplianceType
 Custom:VSSCompliant2 -ResourceType ManagedInstance -Item $item -
ExecutionSummary_ExecutionTime "07-Jun-2019"
```

- Einzelheiten zur API finden Sie unter PutComplianceItems AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`PutInventory`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `PutInventory` verwendet wird.

CLI

AWS CLI

So weisen Sie einer Instance Kundenmetadaten zu

In diesem Beispiel werden einer Instance Informationen zum Rack-Standort zugewiesen. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

Befehl (Linux):

```
aws ssm put-inventory --instance-id "i-016648b75dd622dab" --items
 '[{"TypeName": "Custom:RackInfo","SchemaVersion": "1.0","CaptureTime":
 "2019-01-22T10:01:01Z","Content":[{"RackLocation": "Bay B/Row C/Rack D/Shelf
 E"}]}]'
```

Befehl (Windows):

```
aws ssm put-inventory --instance-id "i-016648b75dd622dab" --
items "TypeName=Custom:RackInfo,SchemaVersion=1.0,CaptureTime=2019-01-22T10:01:01Z,Conten
 B/Row C/Rack D/Shelf F'}]"
```

- Einzelheiten zur API finden Sie PutInventoryin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden einer Instance Informationen zum Rack-Standort zugewiesen. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
$data = New-Object
 "System.Collections.Generic.Dictionary[System.String,System.String]"
$data.Add("RackLocation", "Bay B/Row C/Rack D/Shelf F")

$items = New-Object
 "System.Collections.Generic.List[System.Collections.Generic.Dictionary[System.String,
 System.String]]"
```

```
$items.Add($data)

$customInventoryItem = New-Object
 Amazon.SimpleSystemsManagement.Model.InventoryItem
$customInventoryItem.CaptureTime = "2016-08-22T10:01:01Z"
$customInventoryItem.Content = $items
$customInventoryItem.TypeName = "Custom:TestRackInfo2"
$customInventoryItem.SchemaVersion = "1.0"

$inventoryItems = @($customInventoryItem)

Write-SSMInventory -InstanceId "i-0cb2b964d3e14fd9f" -Item $inventoryItems
```

- Einzelheiten zur API finden Sie unter PutInventory AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden einer Instance Informationen zum Rack-Standort zugewiesen. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
$data = New-Object
 "System.Collections.Generic.Dictionary[System.String,System.String]"
$data.Add("RackLocation", "Bay B/Row C/Rack D/Shelf F")

$items = New-Object
 "System.Collections.Generic.List[System.Collections.Generic.Dictionary[System.String,
 System.String]]"
$items.Add($data)

$customInventoryItem = New-Object
 Amazon.SimpleSystemsManagement.Model.InventoryItem
$customInventoryItem.CaptureTime = "2016-08-22T10:01:01Z"
$customInventoryItem.Content = $items
$customInventoryItem.TypeName = "Custom:TestRackInfo2"
$customInventoryItem.SchemaVersion = "1.0"

$inventoryItems = @($customInventoryItem)

Write-SSMInventory -InstanceId "i-0cb2b964d3e14fd9f" -Item $inventoryItems
```

- Einzelheiten zur API finden Sie unter PutInventory AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`PutParameter`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `PutParameter` verwendet wird.

CLI

AWS CLI

Beispiel 1: So ändern Sie einen Parameterwert

Das folgende Beispiel `put-parameter` ändert den Wert des angegebenen Parameters.

```
aws ssm put-parameter \
    --name "MyStringParameter" \
    --type "String" \
    --value "Vici" \
    --overwrite
```

Ausgabe:

```
{
    "Version": 2,
    "Tier": "Standard"
}
```

Weitere Informationen finden Sie unter Erstellen eines Systems Manager Manager-Parameters (AWS CLI), Verwalten von Parameterschichten und Arbeiten mit Parameterrichtlinien im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So erstellen Sie einen erweiterten Parameter

Das folgende Beispiel `put-parameter` erstellt einen erweiterten Parameter.

```
aws ssm put-parameter \
    --name "MyAdvancedParameter" \
    --description "This is an advanced parameter" \
    --value "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
 eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
```

```
 veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
 consequat [truncated]" \
    --type "String" \
    --tier Advanced
```

Ausgabe:

```
{
    "Version": 1,
    "Tier": "Advanced"
}
```

Weitere Informationen finden Sie unter Erstellen eines Systems Manager Manager-Parameters (AWS CLI), Verwalten von Parameterschichten und Arbeiten mit Parameterrichtlinien im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 3: So konvertieren Sie einen Standardparameter in einen erweiterten Parameter

Im folgenden Beispiel für `put-parameter` wird ein vorhandener Standardparameter in einen erweiterten Parameter konvertiert.

```
aws ssm put-parameter \
    --name "MyConvertedParameter" \
    --value "abc123" \
    --type "String" \
    --tier Advanced \
    --overwrite
```

Ausgabe:

```
{
    "Version": 2,
    "Tier": "Advanced"
}
```

Weitere Informationen finden Sie unter Erstellen eines Systems Manager Manager-Parameters (AWS CLI), Verwalten von Parameterschichten und Arbeiten mit Parameterrichtlinien im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 4: So erstellen Sie einen Parameter mit angehängter Richtlinie

Im folgenden `put-parameter`-Beispiel wird ein erweiterter Parameter mit einer angehängten Parameterrichtlinie erstellt.

```
aws ssm put-parameter \
    --name "/Finance/Payroll/q2accesskey" \
    --value "P@sSwW)rd" \
    --type "SecureString" \
    --tier Advanced \
    --policies "[{\"Type\":\"Expiration\",\"Version\":\"1.0\",\"Attributes\":
{\"Timestamp\":\"2020-06-30T00:00:00.000Z\"}},{\"Type\":\"ExpirationNotification
\",\"Version\":\"1.0\",\"Attributes\":{\"Before\":\"5\",\"Unit\":\"Days\"}},
{\"Type\":\"NoChangeNotification\",\"Version\":\"1.0\",\"Attributes\":{\"After\":
\"60\",\"Unit\":\"Days\"}}]"
```

Ausgabe:

```
{
    "Version": 1,
    "Tier": "Advanced"
}
```

Weitere Informationen finden Sie unter Erstellen eines Systems Manager Manager-Parameters (AWS CLI), Verwalten von Parameterschichten und Arbeiten mit Parameterrichtlinien im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 5: So fügen Sie einem vorhandenen Parameter eine Richtlinie hinzu

Im folgenden `put-parameter`-Beispiel wird eine Richtlinie einem vorhandenen erweiterten Parameter angefügt.

```
aws ssm put-parameter \
    --name "/Finance/Payroll/q2accesskey" \
    --value "N3wP@sSwW)rd" \
    --type "SecureString" \
    --tier Advanced \
    --policies "[{\"Type\":\"Expiration\",\"Version\":\"1.0\",\"Attributes\":
{\"Timestamp\":\"2020-06-30T00:00:00.000Z\"}},{\"Type\":\"ExpirationNotification
\",\"Version\":\"1.0\",\"Attributes\":{\"Before\":\"5\",\"Unit\":\"Days\"}},
{\"Type\":\"NoChangeNotification\",\"Version\":\"1.0\",\"Attributes\":{\"After\":
\"60\",\"Unit\":\"Days\"}}]"
    --overwrite
```

Ausgabe:

```
{
    "Version": 2,
    "Tier": "Advanced"
}
```

Weitere Informationen finden Sie unter Erstellen eines Systems Manager Manager-Parameters (AWS CLI), Verwalten von Parameterschichten und Arbeiten mit Parameterrichtlinien im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie PutParameterin der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.ssm.SsmClient;
import software.amazon.awssdk.services.ssm.model.ParameterType;
import software.amazon.awssdk.services.ssm.model.PutParameterRequest;
import software.amazon.awssdk.services.ssm.model.SsmException;

public class PutParameter {

    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <paraName>

                Where:
                    paraName - The name of the parameter.
                    paraValue - The value of the parameter.
                """;
```

```
        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String paraName = args[0];
        String paraValue = args[1];
        Region region = Region.US_EAST_1;
        SsmClient ssmClient = SsmClient.builder()
                .region(region)
                .build();

        putParaValue(ssmClient, paraName, paraValue);
        ssmClient.close();
    }

    public static void putParaValue(SsmClient ssmClient, String paraName, String
 value) {
        try {
            PutParameterRequest parameterRequest = PutParameterRequest.builder()
                    .name(paraName)
                    .type(ParameterType.STRING)
                    .value(value)
                    .build();

            ssmClient.putParameter(parameterRequest);
            System.out.println("The parameter was successfully added.");

        } catch (SsmException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie PutParameterin der AWS SDK for Java 2.x API-Referenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird ein Parameter erstellt. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Write-SSMParameter -Name "Welcome" -Type "String" -Value "helloWorld"
```

Beispiel 2: In diesem Beispiel wird ein Parameter geändert. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Write-SSMParameter -Name "Welcome" -Type "String" -Value "Good day, Sunshine!" -
Overwrite $true
```

- Einzelheiten zur API finden Sie unter PutParameter AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird ein Parameter erstellt. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Write-SSMParameter -Name "Welcome" -Type "String" -Value "helloWorld"
```

Beispiel 2: In diesem Beispiel wird ein Parameter geändert. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Write-SSMParameter -Name "Welcome" -Type "String" -Value "Good day, Sunshine!" -
Overwrite $true
```

- Einzelheiten zur API finden Sie unter PutParameter AWS -Tools für PowerShellCmdlet-Referenz (V5).

Rust

SDK für Rust

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
async fn make_parameter(
    client: &Client,
    name: &str,
    value: &str,
    description: &str,
) -> Result<(), Error> {
    let resp = client
        .put_parameter()
        .overwrite(true)
        .r#type(ParameterType::String)
        .name(name)
        .value(value)
        .description(description)
        .send()
        .await?;

    println!("Success! Parameter now has version: {}", resp.version());

    Ok(())
}
```

- Einzelheiten zur API finden Sie PutParameterin der API-Referenz zum AWS SDK für Rust.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden dieses Dienstes mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`RegisterDefaultPatchBaseline`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `RegisterDefaultPatchBaseline` verwendet wird.

CLI

AWS CLI

Festlegen der Standard-Patch-Baseline

Im folgenden Beispiel `register-default-patch-baseline` wird die angegebene benutzerdefinierte Patch-Baseline als Standard-Patch-Baseline für den unterstützten Betriebssystemtyp registriert.

```
aws ssm register-default-patch-baseline \
    --baseline-id "pb-abc123cf9bEXAMPLE"
```

Ausgabe:

```
{
    "BaselineId":"pb-abc123cf9bEXAMPLE"
}
```

Im folgenden `register-default-patch-baseline` Beispiel wird die von AWS für CentOS bereitgestellte Standard-Patch-Baseline als Standard-Patch-Baseline registriert.

```
aws ssm register-default-patch-baseline \
    --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-0574b43a65ea646ed"
```

Ausgabe:

```
{
    "BaselineId":"pb-abc123cf9bEXAMPLE"
}
```

Weitere Informationen finden Sie unter Über vordefinierte und benutzerdefinierte Patch-Baselines im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter RegisterDefaultPatchBaseline AWS CLIBefehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird eine Patch-Baseline als Standard-Patch-Baseline registriert.

```
Register-SSMDefaultPatchBaseline -BaselineId "pb-03da896ca3b68b639"
```

Ausgabe:

```
pb-03da896ca3b68b639
```

- Einzelheiten zur API finden Sie unter RegisterDefaultPatchBaseline AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird eine Patch-Baseline als Standard-Patch-Baseline registriert.

```
Register-SSMDefaultPatchBaseline -BaselineId "pb-03da896ca3b68b639"
```

Ausgabe:

```
pb-03da896ca3b68b639
```

- Einzelheiten zur API finden Sie unter RegisterDefaultPatchBaseline AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`RegisterPatchBaselineForPatchGroup`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `RegisterPatchBaselineForPatchGroup` verwendet wird.

CLI

AWS CLI

So registrieren Sie eine Patch-Baseline für eine Patch-Gruppe

Im folgenden `register-patch-baseline-for-patch-group`-Beispiel wird eine Patch-Baseline für eine Patch-Gruppe registriert.

```
aws ssm register-patch-baseline-for-patch-group \
    --baseline-id "pb-045f10b4f382baeda" \
    --patch-group "Production"
```

Ausgabe:

```
{
    "BaselineId": "pb-045f10b4f382baeda",
    "PatchGroup": "Production"
}
```

Weitere Informationen finden Sie unter Erstellen einer Patchgruppe < https:// docs.aws.amazon.com/systems-manager/ latest/userguide/sysman - patch-group-tagging .html>__ und Hinzufügen einer Patchgruppe zu einer Patch-Baseline im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie RegisterPatchBaselineForPatchGroupin AWS CLI der Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dieses Beispiel registriert eine Patch-Baseline für eine Patch-Gruppe.

```
Register-SSMPatchBaselineForPatchGroup -BaselineId "pb-03da896ca3b68b639" -
PatchGroup "Production"
```

Ausgabe:

```
BaselineId           PatchGroup
----------           ----------
```

```
pb-03da896ca3b68b639 Production
```

- Einzelheiten zur API finden Sie unter RegisterPatchBaselineForPatchGroup AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dieses Beispiel registriert eine Patch-Baseline für eine Patch-Gruppe.

```
Register-SSMPatchBaselineForPatchGroup -BaselineId "pb-03da896ca3b68b639" -
PatchGroup "Production"
```

Ausgabe:

```
BaselineId           PatchGroup
----------           ----------
pb-03da896ca3b68b639 Production
```

- Einzelheiten zur API finden Sie unter RegisterPatchBaselineForPatchGroup AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`RegisterTargetWithMaintenanceWindow`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `RegisterTargetWithMaintenanceWindow` verwendet wird.

CLI

AWS CLI

Beispiel 1: So können Sie ein einzelnes Ziel mit einem Wartungsfenster registrieren

Im folgenden Beispiel `register-target-with-maintenance-window` wird eine Instance mit einem Wartungsfenster registriert.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
```

```
    --target "Key=InstanceIds,Values=i-0000293ffd8c57862" \
    --owner-information "Single instance" \
    --resource-type "INSTANCE"
```

Ausgabe:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Beispiel 2: Um mehrere Ziele mithilfe einer Instanz für ein Wartungsfenster zu registrieren IDs

Im folgenden `register-target-with-maintenance-window` Beispiel werden zwei Instanzen mit einem Wartungsfenster registriert, indem ihre Instanz angegeben wird IDs.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --target "Key=InstanceIds,Values=i-0000293ffd8c57862,i-0cb2b964d3e14fd9f" \
    --owner-information "Two instances in a list" \
    --resource-type "INSTANCE"
```

Ausgabe:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Beispiel 3: So registrieren Sie Ziele mit einem Wartungsfenster mithilfe von Ressourcentags

Im folgenden Beispiel `register-target-with-maintenance-window` werden Instances mit einem Wartungsfenster registriert, indem Ressourcen-Tags angegeben werden, die auf die Instances angewendet wurden.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-06cf17cbefcb4bf4f" \
    --targets "Key=tag:Environment,Values=Prod" "Key=Role,Values=Web" \
    --owner-information "Production Web Servers" \
    --resource-type "INSTANCE"
```

Ausgabe:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Beispiel 4: So registrieren Sie Ziele mithilfe einer Gruppe von Tag-Schlüsseln

Im folgenden `register-target-with-maintenance-window`-Beispiel werden Instances registriert, denen unabhängig von ihren Schlüsselwerten jeweils ein oder mehrere Tag-Schlüssel zugewiesen sind.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "INSTANCE" \
    --target "Key=tag-key,Values=Name,Instance-Type,CostCenter"
```

Ausgabe:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Beispiel 5: Registrieren von Zielen unter Verwendung eines Ressourcengruppennamens

Das folgende `register-target-with-maintenance-window`-Beispiel registriert eine angegebene Ressourcengruppe, unabhängig vom Typ der darin enthaltenen Ressourcen.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "RESOURCE_GROUP" \
    --target "Key=resource-groups:Name,Values=MyResourceGroup"
```

Ausgabe:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Weitere Informationen finden Sie unter Registrieren einer Zielinstanz mit dem Wartungsfenster (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie RegisterTargetWithMaintenanceWindowin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird eine Instance mit einem Wartungsfenster registriert.

```
$option1 = @{Key="InstanceIds";Values=@("i-0000293ffd8c57862")}
Register-SSMTargetWithMaintenanceWindow -WindowId "mw-06cf17cbefcb4bf4f" -Target
 $option1 -OwnerInformation "Single instance" -ResourceType "INSTANCE"
```

Ausgabe:

```
d8e47760-23ed-46a5-9f28-927337725398
```

Beispiel 2: In diesem Beispiel werden mehrere Instances mit einem Wartungsfenster registriert.

```
$option1 =
 @{Key="InstanceIds";Values=@("i-0000293ffd8c57862","i-0cb2b964d3e14fd9f")}
Register-SSMTargetWithMaintenanceWindow -WindowId "mw-06cf17cbefcb4bf4f" -Target
 $option1 -OwnerInformation "Single instance" -ResourceType "INSTANCE"
```

Ausgabe:

```
6ab5c208-9fc4-4697-84b7-b02a6cc25f7d
```

Beispiel 3: In diesem Beispiel wird mithilfe von EC2 Tags eine Instanz mit einem Wartungsfenster registriert.

```
$option1 = @{Key="tag:Environment";Values=@("Production")}
Register-SSMTargetWithMaintenanceWindow -WindowId "mw-06cf17cbefcb4bf4f" -Target
 $option1 -OwnerInformation "Production Web Servers" -ResourceType "INSTANCE"
```

Ausgabe:

```
2994977e-aefb-4a71-beac-df620352f184
```

- Einzelheiten zur API finden Sie unter RegisterTargetWithMaintenanceWindow AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird eine Instance mit einem Wartungsfenster registriert.

```
$option1 = @{Key="InstanceIds";Values=@("i-0000293ffd8c57862")}
Register-SSMTargetWithMaintenanceWindow -WindowId "mw-06cf17cbefcb4bf4f" -Target
 $option1 -OwnerInformation "Single instance" -ResourceType "INSTANCE"
```

Ausgabe:

```
d8e47760-23ed-46a5-9f28-927337725398
```

Beispiel 2: In diesem Beispiel werden mehrere Instances mit einem Wartungsfenster registriert.

```
$option1 =
 @{Key="InstanceIds";Values=@("i-0000293ffd8c57862","i-0cb2b964d3e14fd9f")}
Register-SSMTargetWithMaintenanceWindow -WindowId "mw-06cf17cbefcb4bf4f" -Target
 $option1 -OwnerInformation "Single instance" -ResourceType "INSTANCE"
```

Ausgabe:

```
6ab5c208-9fc4-4697-84b7-b02a6cc25f7d
```

Beispiel 3: In diesem Beispiel wird mithilfe von EC2 Tags eine Instanz mit einem Wartungsfenster registriert.

```
$option1 = @{Key="tag:Environment";Values=@("Production")}
Register-SSMTargetWithMaintenanceWindow -WindowId "mw-06cf17cbefcb4bf4f" -Target
 $option1 -OwnerInformation "Production Web Servers" -ResourceType "INSTANCE"
```

Ausgabe:

```
2994977e-aefb-4a71-beac-df620352f184
```

- Einzelheiten zur API finden Sie unter RegisterTargetWithMaintenanceWindow AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`RegisterTaskWithMaintenanceWindow`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `RegisterTaskWithMaintenanceWindow` verwendet wird.

CLI

AWS CLI

Beispiel 1: Um eine Automation-Aufgabe mit einem Wartungsfenster zu registrieren

Im folgenden Beispiel `register-task-with-maintenance-window` wird eine Automation-Aufgabe mit einem Wartungsfenster registriert, das auf eine Instance ausgerichtet ist.

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-082dcd7649EXAMPLE" \
    --targets Key=InstanceIds,Values=i-1234520122EXAMPLE \
    --task-arn AWS-RestartEC2Instance \
    --service-role-arn arn:aws:iam::111222333444:role/SSM --task-type AUTOMATION
 \
    --task-invocation-parameters "{\"Automation\":{\"DocumentVersion\":\"\$LATEST
\",\"Parameters\":{\"InstanceId\":[\"{{RESOURCE_ID}}\"]}}}" \
    --priority 0 \
    --max-concurrency 1 \
    --max-errors 1 \
    --name "AutomationExample" \
    --description "Restarting EC2 Instance for maintenance"
```

Ausgabe:

```
{
    "WindowTaskId":"11144444-5555-6666-7777-88888888"
}
```

Weitere Informationen finden Sie unter Registrieren einer Aufgabe im Wartungsfenster (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 2: So registrieren Sie eine Lambda-Aufgabe mit einem Wartungsfenster

Im folgenden Beispiel `register-task-with-maintenance-window` wird eine Lambda-Aufgabe mit einem Wartungsfenster registriert, das auf eine Instance ausgerichtet ist.

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-082dcd7649dee04e4" \
    --targets Key=InstanceIds,Values=i-12344d305eEXAMPLE \
    --task-arn arn:aws:lambda:us-east-1:111222333444:function:SSMTestLAMBDA \
    --service-role-arn arn:aws:iam::111222333444:role/SSM \
    --task-type LAMBDA \
    --task-invocation-parameters '{"Lambda":{"Payload":"{\"InstanceId\":
\"{{RESOURCE_ID}}\",\"targetType\":\"{{TARGET_TYPE}}\"}","Qualifier":"$LATEST"}}'
 \
    --priority 0 \
    --max-concurrency 10 \
    --max-errors 5 \
    --name "Lambda_Example" \
    --description "My Lambda Example"
```

Ausgabe:

```
{
    "WindowTaskId":"22244444-5555-6666-7777-88888888"
}
```

Weitere Informationen finden Sie unter Registrieren einer Aufgabe im Wartungsfenster (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 3: So registrieren Sie eine Run-Command-Aufgabe mit einem Wartungsfenster

Im folgenden Beispiel `register-task-with-maintenance-window` wird eine Run-Command-Aufgabe mit einem Wartungsfenster registriert, das auf eine Instance ausgerichtet ist.

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-082dcd7649dee04e4" \
    --targets "Key=InstanceIds,Values=i-12344d305eEXAMPLE" \
    --service-role-arn "arn:aws:iam::111222333444:role/SSM" \
    --task-type "RUN_COMMAND" \
    --name "SSMInstallPowerShellModule" \
    --task-arn "AWS-InstallPowerShellModule" \
    --task-invocation-parameters "{\"RunCommand\":{\"Comment\":\"\",
\"OutputS3BucketName\":\"runcommandlogs\",\"Parameters\":{\"commands\":[\"Get-
```

```
Module -ListAvailable\"],\"executionTimeout\":[\"3600\"],\"source\":[\"https:\/
\/gallery.technet.microsoft.com\/EZOut-33ae0fb7\/file\/110351\/1\/EZOut.zip\"],
\"workingDirectory\":[\"\\\\\"]},\"TimeoutSeconds\":600}}" \
    --max-concurrency 1 \
    --max-errors 1 \
    --priority 10
```

Ausgabe:

```
{
    "WindowTaskId":"33344444-5555-6666-7777-88888888"
}
```

Weitere Informationen finden Sie unter Registrieren einer Aufgabe im Wartungsfenster (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 4: So registrieren Sie eine Step-Functions-Aufgabe bei einem Wartungsfenster

Im folgenden Beispiel `register-task-with-maintenance-window` wird eine Step-Functions-Aufgabe mit einem Wartungsfenster registriert, das auf eine Instance ausgerichtet ist.

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-1234d787d6EXAMPLE" \
    --targets Key=WindowTargetIds,Values=12347414-69c3-49f8-95b8-ed2dcEXAMPLE \
    --task-arn arn:aws:states:us-
east-1:111222333444:stateMachine:SSMTestStateMachine \
    --service-role-arn arn:aws:iam::111222333444:role/MaintenanceWindows \
    --task-type STEP_FUNCTIONS \
    --task-invocation-parameters '{"StepFunctions":{"Input":"{\"InstanceId\":
\"{{RESOURCE_ID}}\"}"}}' \
    --priority 0 \
    --max-concurrency 10 \
    --max-errors 5 \
    --name "Step_Functions_Example" \
    --description "My Step Functions Example"
```

Ausgabe:

```
{
    "WindowTaskId":"44444444-5555-6666-7777-88888888"
}
```

Weitere Informationen finden Sie unter Registrieren einer Aufgabe im Wartungsfenster (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

Beispiel 5: So registrieren Sie eine Aufgabe mithilfe einer Windows-Wartungsziel-ID

Das folgende `register-task-with-maintenance-window`-Beispiel registriert eine Aufgabe mithilfe einer Wartungsfenster-Ziel-ID. Die Ziel-ID des Wartungsfensters war in der Ausgabe des `aws ssm register-target-with-maintenance-window`-Befehls enthalten. Sie können sie auch aus der Ausgabe des `aws ssm describe-maintenance-window-targets`-Befehls abrufen.

```
aws ssm register-task-with-maintenance-window \
    --targets "Key=WindowTargetIds,Values=350d44e6-28cc-44e2-951f-4b2c9EXAMPLE" \
    --task-arn "AWS-RunShellScript" \
    --service-role-arn "arn:aws:iam::111222333444:role/MaintenanceWindowsRole" \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --task-type "RUN_COMMAND" \
    --task-parameters  "{\"commands\":{\"Values\":[\"df\"]}}" \
    --max-concurrency 1 \
    --max-errors 1 \
    --priority 10
```

Ausgabe:

```
{
    "WindowTaskId":"33344444-5555-6666-7777-88888888"
}
```

Weitere Informationen finden Sie unter Registrieren einer Aufgabe im Wartungsfenster (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie RegisterTaskWithMaintenanceWindowin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird eine Aufgabe mit einem Wartungsfenster unter Verwendung einer Instance-ID registriert. Die Ausgabe ist die Aufgaben-ID.

```
$parameters = @{}
$parameterValues = New-Object
 Amazon.SimpleSystemsManagement.Model.MaintenanceWindowTaskParameterValueExpression
$parameterValues.Values = @("Install")
$parameters.Add("Operation", $parameterValues)

Register-SSMTaskWithMaintenanceWindow -WindowId "mw-03a342e62c96d31b0"
 -ServiceRoleArn "arn:aws:iam::123456789012:role/MaintenanceWindowsRole"
 -MaxConcurrency 1 -MaxError 1 -TaskArn "AWS-RunShellScript" -Target
 @{ Key="InstanceIds";Values="i-0000293ffd8c57862" } -TaskType "RUN_COMMAND" -
Priority 10 -TaskParameter $parameters
```

Ausgabe:

```
f34a2c47-ddfd-4c85-a88d-72366b69af1b
```

Beispiel 2: In diesem Beispiel wird eine Aufgabe mit einem Wartungsfenster unter Verwendung einer Ziel-ID registriert. Die Ausgabe ist die Aufgaben-ID.

```
$parameters = @{}
$parameterValues = New-Object
 Amazon.SimpleSystemsManagement.Model.MaintenanceWindowTaskParameterValueExpression
$parameterValues.Values = @("Install")
$parameters.Add("Operation", $parameterValues)

register-ssmtaskwithmaintenancewindow -WindowId "mw-03a342e62c96d31b0"
 -ServiceRoleArn "arn:aws:iam::123456789012:role/MaintenanceWindowsRole"
 -MaxConcurrency 1 -MaxError 1 -TaskArn "AWS-RunShellScript" -Target
 @{ Key="WindowTargetIds";Values="350d44e6-28cc-44e2-951f-4b2c985838f6" } -
TaskType "RUN_COMMAND" -Priority 10 -TaskParameter $parameters
```

Ausgabe:

```
f34a2c47-ddfd-4c85-a88d-72366b69af1b
```

Beispiel 3: In diesem Beispiel wird ein Parameterobjekt für das Run-Befehlsdokument **AWS-RunPowerShellScript** und eine Aufgabe mit einem bestimmten Wartungsfenster unter Verwendung der Ziel-ID erstellt. Die Rückgabeausgabe ist die Aufgaben-ID.

```
$parameters =
 [Collections.Generic.Dictionary[String,Collections.Generic.List[String]]]::new()
```

```
$parameters.Add("commands",@("ipconfig","dir env:\computername"))
$parameters.Add("executionTimeout",@(3600))

$props = @{
    WindowId = "mw-0123e4cce56ff78ae"
    ServiceRoleArn = "arn:aws:iam::123456789012:role/MaintenanceWindowsRole"
    MaxConcurrency = 1
    MaxError = 1
    TaskType = "RUN_COMMAND"
    TaskArn = "AWS-RunPowerShellScript"
    Target =
 @{Key="WindowTargetIds";Values="fe1234ea-56d7-890b-12f3-456b789bee0f"}
    Priority = 1
    RunCommand_Parameter = $parameters
    Name = "set-via-cmdlet"
}

Register-SSMTaskWithMaintenanceWindow @props
```

Ausgabe:

```
f1e2ef34-5678-12e3-456a-12334c5c6cbe
```

Beispiel 4: In diesem Beispiel wird eine AWS Systems Manager Automation-Aufgabe mithilfe eines Dokuments mit dem Namen registriert**`Create-Snapshots`**.

```
$automationParameters = @{}
$automationParameters.Add( "instanceId", @("{{ TARGET_ID }}") )
$automationParameters.Add( "AutomationAssumeRole",
 @("{arn:aws:iam::111111111111:role/AutomationRole}") )
$automationParameters.Add( "SnapshotTimeout", @("PT20M") )
Register-SSMTaskWithMaintenanceWindow -WindowId mw-123EXAMPLE456`
    -ServiceRoleArn "arn:aws:iam::123456789012:role/MW-Role"`
    -MaxConcurrency 1 -MaxError 1 -TaskArn "CreateVolumeSnapshots"`
    -Target @{ Key="WindowTargetIds";Values="4b5acdf4-946c-4355-
bd68-4329a43a5fd1" }`
    -TaskType "AUTOMATION"`
    -Priority 4`
    -Automation_DocumentVersion '$DEFAULT' -Automation_Parameter
 $automationParameters -Name "Create-Snapshots"
```

- Einzelheiten zur API finden Sie unter RegisterTaskWithMaintenanceWindow AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird eine Aufgabe mit einem Wartungsfenster unter Verwendung einer Instance-ID registriert. Die Ausgabe ist die Aufgaben-ID.

```
$parameters = @{}
$parameterValues = New-Object
 Amazon.SimpleSystemsManagement.Model.MaintenanceWindowTaskParameterValueExpression
$parameterValues.Values = @("Install")
$parameters.Add("Operation", $parameterValues)

Register-SSMTaskWithMaintenanceWindow -WindowId "mw-03a342e62c96d31b0"
 -ServiceRoleArn "arn:aws:iam::123456789012:role/MaintenanceWindowsRole"
 -MaxConcurrency 1 -MaxError 1 -TaskArn "AWS-RunShellScript" -Target
 @{ Key="InstanceIds";Values="i-0000293ffd8c57862" } -TaskType "RUN_COMMAND" -
Priority 10 -TaskParameter $parameters
```

Ausgabe:

```
f34a2c47-ddfd-4c85-a88d-72366b69af1b
```

Beispiel 2: In diesem Beispiel wird eine Aufgabe mit einem Wartungsfenster unter Verwendung einer Ziel-ID registriert. Die Ausgabe ist die Aufgaben-ID.

```
$parameters = @{}
$parameterValues = New-Object
 Amazon.SimpleSystemsManagement.Model.MaintenanceWindowTaskParameterValueExpression
$parameterValues.Values = @("Install")
$parameters.Add("Operation", $parameterValues)

register-ssmtaskwithmaintenancewindow -WindowId "mw-03a342e62c96d31b0"
 -ServiceRoleArn "arn:aws:iam::123456789012:role/MaintenanceWindowsRole"
 -MaxConcurrency 1 -MaxError 1 -TaskArn "AWS-RunShellScript" -Target
 @{ Key="WindowTargetIds";Values="350d44e6-28cc-44e2-951f-4b2c985838f6" } -
TaskType "RUN_COMMAND" -Priority 10 -TaskParameter $parameters
```

Ausgabe:

```
f34a2c47-ddfd-4c85-a88d-72366b69af1b
```

Beispiel 3: In diesem Beispiel wird ein Parameterobjekt für das Run-Befehlsdokument **AWS-RunPowerShellScript** und eine Aufgabe mit einem bestimmten Wartungsfenster unter Verwendung der Ziel-ID erstellt. Die Rückgabeausgabe ist die Aufgaben-ID.

```
$parameters =
 [Collections.Generic.Dictionary[String,Collections.Generic.List[String]]]::new()
$parameters.Add("commands",@("ipconfig","dir env:\computername"))
$parameters.Add("executionTimeout",@(3600))

$props = @{
    WindowId = "mw-0123e4cce56ff78ae"
    ServiceRoleArn = "arn:aws:iam::123456789012:role/MaintenanceWindowsRole"
    MaxConcurrency = 1
    MaxError = 1
    TaskType = "RUN_COMMAND"
    TaskArn = "AWS-RunPowerShellScript"
    Target =
 @{Key="WindowTargetIds";Values="fe1234ea-56d7-890b-12f3-456b789bee0f"}
    Priority = 1
    RunCommand_Parameter = $parameters
    Name = "set-via-cmdlet"
}

Register-SSMTaskWithMaintenanceWindow @props
```

Ausgabe:

```
f1e2ef34-5678-12e3-456a-12334c5c6cbe
```

Beispiel 4: In diesem Beispiel wird eine AWS Systems Manager Automation-Aufgabe mithilfe eines Dokuments mit dem Namen registriert**Create-Snapshots**.

```
$automationParameters = @{}
$automationParameters.Add( "instanceId", @("{{ TARGET_ID }}") )
$automationParameters.Add( "AutomationAssumeRole",
 @("{arn:aws:iam::111111111111:role/AutomationRole}") )
$automationParameters.Add( "SnapshotTimeout", @("PT20M") )
Register-SSMTaskWithMaintenanceWindow -WindowId mw-123EXAMPLE456`
    -ServiceRoleArn "arn:aws:iam::123456789012:role/MW-Role"`
    -MaxConcurrency 1 -MaxError 1 -TaskArn "CreateVolumeSnapshots"`
    -Target @{ Key="WindowTargetIds";Values="4b5acdf4-946c-4355-
bd68-4329a43a5fd1" }`
```

```
    -TaskType "AUTOMATION"`
    -Priority 4`
    -Automation_DocumentVersion '$DEFAULT' -Automation_Parameter
 $automationParameters -Name "Create-Snapshots"
```

- Einzelheiten zur API finden Sie unter RegisterTaskWithMaintenanceWindow AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`RemoveTagsFromResource`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `RemoveTagsFromResource` verwendet wird.

CLI

AWS CLI

Entfernen eines Tags aus einer Patch-Baseline

Das folgende `remove-tags-from-resource`-Beispiel entfernt Tags aus einer Patch-Baseline.

```
aws ssm remove-tags-from-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0123456789abcdef0" \
    --tag-keys "Region"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter AWS Ressourcen mit Tags versehen in der AWS allgemeinen Referenz.

- Einzelheiten zur API finden Sie RemoveTagsFromResourcein der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird ein Tag aus einem Wartungsfenster entfernt. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Remove-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow" -TagKey "Production"
```

- Einzelheiten zur API finden Sie unter RemoveTagsFromResource AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird ein Tag aus einem Wartungsfenster entfernt. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Remove-SSMResourceTag -ResourceId "mw-03eb9db42890fb82d" -ResourceType
 "MaintenanceWindow" -TagKey "Production"
```

- Einzelheiten zur API finden Sie unter RemoveTagsFromResource AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **SendCommand** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `SendCommand` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Kennenlernen der Grundlagen

CLI

AWS CLI

Beispiel 1: So führen Sie einen Befehl auf einer oder mehreren Remote-Instances aus

Im folgenden Beispiel `send-command` wird ein `echo`-Befehl auf einer Ziel-Instance ausgeführt.

```
aws ssm send-command \
    --document-name "AWS-RunShellScript" \
    --parameters 'commands=["echo HelloWorld"]' \
    --targets "Key=instanceids,Values=i-1234567890abcdef0" \
    --comment "echo HelloWorld"
```

Ausgabe:

```
{
    "Command": {
        "CommandId": "92853adf-ba41-4cd6-9a88-142d1EXAMPLE",
        "DocumentName": "AWS-RunShellScript",
        "DocumentVersion": "",
        "Comment": "echo HelloWorld",
        "ExpiresAfter": 1550181014.717,
        "Parameters": {
            "commands": [
                "echo HelloWorld"
            ]
        },
        "InstanceIds": [
            "i-0f00f008a2dcbefe2"
        ],
        "Targets": [],
        "RequestedDateTime": 1550173814.717,
        "Status": "Pending",
        "StatusDetails": "Pending",
        "OutputS3BucketName": "",
        "OutputS3KeyPrefix": "",
        "MaxConcurrency": "50",
        "MaxErrors": "0",
        "TargetCount": 1,
        "CompletedCount": 0,
        "ErrorCount": 0,
```

```
        "DeliveryTimedOutCount": 0,
        "ServiceRole": "",
        "NotificationConfig": {
            "NotificationArn": "",
            "NotificationEvents": [],
            "NotificationType": ""
        },
        "CloudWatchOutputConfig": {
            "CloudWatchLogGroupName": "",
            "CloudWatchOutputEnabled": false
        }
    }
}
```

Weitere Informationen finden Sie unter Run-Befehl mithilfe von Systems Manager in der AWS Systems-Manager-Benutzerhandbuch.

Beispiel 2: Um IP-Informationen über eine Instance abzurufen

Im folgenden `send-command`-Beispiel werden Informationen über die Instances zurückgegeben.

```
aws ssm send-command \
    --instance-ids "i-1234567890abcdef0" \
    --document-name "AWS-RunShellScript" \
    --comment "IP config" \
    --parameters "commands=ifconfig"
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter Run-Befehl mithilfe von Systems Manager in der AWS Systems-Manager-Benutzerhandbuch.

Beispiel 3: So führen Sie einen Befehl für Instances mit bestimmten Tags aus

Im folgenden Beispiel `send-command` wird ein Befehl auf Instances ausgeführt, die den Tag-Schlüssel „ENV“ und den Wert „Dev“ haben.

```
aws ssm send-command \
    --targets "Key=tag:ENV,Values=Dev" \
    --document-name "AWS-RunShellScript" \
    --parameters "commands=ifconfig"
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter Run-Befehl mithilfe von Systems Manager in der AWS Systems-Manager-Benutzerhandbuch.

Beispiel 4: So führen Sie einen Befehl aus, der SNS-Benachrichtigungen sendet

Im folgenden Beispiel `send-command` wird ein Befehl ausgeführt, der SNS-Benachrichtigungen für alle Benachrichtigungsereignisse und den `Command`-Benachrichtigungstyp sendet.

```
aws ssm send-command \
    --instance-ids "i-1234567890abcdef0" \
    --document-name "AWS-RunShellScript" \
    --comment "IP config" \
    --parameters "commands=ifconfig" \
    --service-role-arn "arn:aws:iam::123456789012:role/SNS_Role" \
    --notification-config "NotificationArn=arn:aws:sns:us-
east-1:123456789012:SNSTopicName,NotificationEvents=All,NotificationType=Command"
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter Run-Befehl mithilfe von Systems Manager in der AWS Systems-Manager-Benutzerhandbuch.

Beispiel 5: Um einen Befehl auszuführen, der an S3 ausgegeben wird und CloudWatch

Im folgenden `send-command` Beispiel wird ein Befehl ausgeführt, der Befehlsdetails an einen S3-Bucket und eine CloudWatch Logs-Protokollgruppe ausgibt.

```
aws ssm send-command \
    --instance-ids "i-1234567890abcdef0" \
    --document-name "AWS-RunShellScript" \
    --comment "IP config" \
    --parameters "commands=ifconfig" \
    --output-s3-bucket-name "s3-bucket-name" \
    --output-s3-key-prefix "runcommand" \
    --cloud-watch-output-
config "CloudWatchOutputEnabled=true,CloudWatchLogGroupName=CWLGroupName"
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter Run-Befehl mithilfe von Systems Manager in der AWS - Systems-Manager-Benutzerhandbuch.

Beispiel 6: So führen Sie Befehle auf mehreren Instances mit unterschiedlichen Tags aus

Im folgenden Beispiel `send-command` wird ein Befehl für Instances mit zwei verschiedenen Tag-Schlüsseln und -Werten ausgeführt.

```
aws ssm send-command \
    --document-name "AWS-RunPowerShellScript" \
    --parameters commands=["echo helloWorld"] \
    --targets Key=tag:Env,Values=Dev Key=tag:Role,Values=WebServers
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter Run-Befehl mithilfe von Systems Manager in der AWS - Systems-Manager-Benutzerhandbuch.

Beispiel 7: So zielen Sie mehrere Instances mit demselben Tag-Schlüssel als Ziel ab

Im folgenden Beispiel `send-command` wird ein Befehl für Instances ausgeführt, die denselben Tag-Schlüssel, aber unterschiedliche Werte haben.

```
aws ssm send-command \
    --document-name "AWS-RunPowerShellScript" \
    --parameters commands=["echo helloWorld"] \
    --targets Key=tag:Env,Values=Dev,Test
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter Run-Befehl mithilfe von Systems Manager in der AWS - Systems-Manager-Benutzerhandbuch.

Beispiel 8: So führen Sie einen Befehl aus, der ein freigegebenes Dokument verwendet

Im folgenden Beispiel `send-command` wird ein gemeinsam verwendetes Dokument auf einer Ziel-Instance ausgeführt.

```
aws ssm send-command \
    --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument"
 \
    --targets "Key=instanceids,Values=i-1234567890abcdef0"
```

Eine Beispielausgabe finden Sie in Beispiel 1.

Weitere Informationen finden Sie unter Verwenden von geteilten SSM-Dokumenten im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie SendCommandunter AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    /**
     * Sends a SSM command to a managed node asynchronously.
     *
     * @param documentName The name of the document to use.
     * @param instanceId The ID of the instance to send the command to.
     * @return The command ID.
     * <p>
     * This method initiates asynchronous requests to send a SSM command to a
managed node.
     * It waits until the document is active, sends the command, and checks the
command execution status.
     */
    public String sendSSMCommand(String documentName, String instanceId) throws
InterruptedException, SsmException {
        // Before we use Document to send a command - make sure it is active.
        CompletableFuture<Void> documentActiveFuture =
CompletableFuture.runAsync(() -> {
            boolean isDocumentActive = false;
            DescribeDocumentRequest request = DescribeDocumentRequest.builder()
                    .name(documentName)
                    .build();

            while (!isDocumentActive) {
                CompletableFuture<DescribeDocumentResponse> response =
getAsyncClient().describeDocument(request);
```

```
                String documentStatus =
response.join().document().statusAsString();
                if (documentStatus.equals("Active")) {
                    System.out.println("The SSM document is active and ready to
use.");
                    isDocumentActive = true;
                } else {
                    System.out.println("The SSM document is not active. Status: "
+ documentStatus);
                    try {
                        Thread.sleep(5000);
                    } catch (InterruptedException e) {
                        throw new RuntimeException(e);
                    }
                }
            }
        });

        documentActiveFuture.join();

        // Create the SendCommandRequest.
        SendCommandRequest commandRequest = SendCommandRequest.builder()
                .documentName(documentName)
                .instanceIds(instanceId)
                .build();

        // Send the command.
        CompletableFuture<SendCommandResponse> commandFuture =
getAsyncClient().sendCommand(commandRequest);
        final String[] commandId = {null};

        commandFuture.whenComplete((commandResponse, ex) -> {
            if (commandResponse != null) {
                commandId[0] = commandResponse.command().commandId();
                System.out.println("Command ID: " + commandId[0]);

                // Wait for the command execution to complete.
                GetCommandInvocationRequest invocationRequest =
GetCommandInvocationRequest.builder()
                        .commandId(commandId[0])
                        .instanceId(instanceId)
                        .build();

                try {
```

```
                    System.out.println("Wait 5 secs");
                    TimeUnit.SECONDS.sleep(5);

                    // Retrieve the command execution details.
                    CompletableFuture<GetCommandInvocationResponse>
invocationFuture = getAsyncClient().getCommandInvocation(invocationRequest);
                    invocationFuture.whenComplete((commandInvocationResponse,
invocationEx) -> {
                        if (commandInvocationResponse != null) {
                            // Check the status of the command execution.
                            CommandInvocationStatus status =
commandInvocationResponse.status();
                            if (status == CommandInvocationStatus.SUCCESS) {
                                System.out.println("Command execution
successful");
                            } else {
                                System.out.println("Command execution failed.
Status: " + status);
                            }
                        } else {
                            Throwable invocationCause = (invocationEx instanceof
CompletionException) ? invocationEx.getCause() : invocationEx;
                            throw new CompletionException(invocationCause);
                        }
                    }).join();
                } catch (InterruptedException e) {
                    throw new RuntimeException(e);
                }
            } else {
                Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
                if (cause instanceof SsmException) {
                    throw (SsmException) cause;
                } else {
                    throw new RuntimeException(cause);
                }
            }
        }).join();

        return commandId[0];
    }
```

- Einzelheiten zur API finden Sie SendCommandin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import { SendCommandCommand, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * Send an SSM command to a managed node.
 * @param {{ documentName: string }}
 */
export const main = async ({ documentName }) => {
  const client = new SSMClient({});
  try {
    await client.send(
      new SendCommandCommand({
        DocumentName: documentName,
      }),
    );
    console.log("Command sent successfully.");
    return { Success: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ValidationError") {
      console.warn(`${caught.message}. Did you provide a valid document name?`);
    } else {
      throw caught;
    }
  }
};
```

- Einzelheiten zur API finden Sie SendCommandin der AWS SDK für JavaScript API-Referenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird ein Echo-Befehl auf einer Ziel-Instance ausgeführt.

```
Send-SSMCommand -DocumentName "AWS-RunPowerShellScript" -Parameter @{commands =
 "echo helloWorld"} -Target @{Key="instanceids";Values=@("i-0cb2b964d3e14fd9f")}
```

Ausgabe:

```
CommandId          : d8d190fc-32c1-4d65-a0df-ff5ff3965524
Comment            :
CompletedCount     : 0
DocumentName       : AWS-RunPowerShellScript
ErrorCount         : 0
ExpiresAfter       : 3/7/2017 10:48:37 PM
InstanceIds        : {}
MaxConcurrency     : 50
MaxErrors          : 0
NotificationConfig : Amazon.SimpleSystemsManagement.Model.NotificationConfig
OutputS3BucketName :
OutputS3KeyPrefix  :
OutputS3Region     :
Parameters         : {[commands,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
RequestedDateTime  : 3/7/2017 9:48:37 PM
ServiceRole        :
Status             : Pending
StatusDetails      : Pending
TargetCount        : 0
Targets            : {instanceids}
```

Beispiel 2: Dieses Beispiel zeigt, wie ein Befehl ausgeführt wird, der verschachtelte Parameter akzeptiert.

```
Send-SSMCommand -DocumentName "AWS-RunRemoteScript" -Parameter
 @{ sourceType="GitHub";sourceInfo='{"owner": "me","repository": "amazon-
ssm","path": "Examples/Install-Win32OpenSSH"}'; "commandLine"=".\Install-
Win32OpenSSH.ps1"} -InstanceId i-0cb2b964d3e14fd9f
```

- Einzelheiten zur API finden Sie unter SendCommand AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird ein Echo-Befehl auf einer Ziel-Instance ausgeführt.

```
Send-SSMCommand -DocumentName "AWS-RunPowerShellScript" -Parameter @{commands =
 "echo helloWorld"} -Target @{Key="instanceids";Values=@("i-0cb2b964d3e14fd9f")}
```

Ausgabe:

```
CommandId          : d8d190fc-32c1-4d65-a0df-ff5ff3965524
Comment            :
CompletedCount     : 0
DocumentName       : AWS-RunPowerShellScript
ErrorCount         : 0
ExpiresAfter       : 3/7/2017 10:48:37 PM
InstanceIds        : {}
MaxConcurrency     : 50
MaxErrors          : 0
NotificationConfig : Amazon.SimpleSystemsManagement.Model.NotificationConfig
OutputS3BucketName :
OutputS3KeyPrefix  :
OutputS3Region     :
Parameters         : {[commands,
 Amazon.Runtime.Internal.Util.AlwaysSendList`1[System.String]]}
RequestedDateTime  : 3/7/2017 9:48:37 PM
ServiceRole        :
Status             : Pending
StatusDetails      : Pending
TargetCount        : 0
Targets            : {instanceids}
```

Beispiel 2: Dieses Beispiel zeigt, wie ein Befehl ausgeführt wird, der verschachtelte Parameter akzeptiert.

```
Send-SSMCommand -DocumentName "AWS-RunRemoteScript" -Parameter
 @{ sourceType="GitHub";sourceInfo='{"owner": "me","repository": "amazon-
ssm","path": "Examples/Install-Win32OpenSSH"}'; "commandLine"=".\Install-
Win32OpenSSH.ps1"} -InstanceId i-0cb2b964d3e14fd9f
```

- Einzelheiten zur API finden Sie unter SendCommand AWS -Tools für PowerShellCmdlet-Referenz (V5).

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class DocumentWrapper:
    """Encapsulates AWS Systems Manager Document actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.name = None

    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def send_command(self, instance_ids):
        """
        Sends a command to one or more instances.

        :param instance_ids: The IDs of the instances to send the command to.
        :return: The ID of the command.
        """
        try:
            response = self.ssm_client.send_command(
                InstanceIds=instance_ids, DocumentName=self.name,
 TimeoutSeconds=3600
            )
```

```
            return response["Command"]["CommandId"]
        except ClientError as err:
            logger.error(
                "Couldn't send command to %s. Here's why: %s: %s",
                self.name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Einzelheiten zur API finden Sie SendCommandin AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`StartAutomationExecution`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `StartAutomationExecution` verwendet wird.

CLI

AWS CLI

Beispiel 1: So führen Sie ein Automation-Dokument aus

Im folgenden Beispiel `start-automation-execution` wird ein Automation-Dokument ausgeführt.

```
aws ssm start-automation-execution \
    --document-name "AWS-UpdateLinuxAmi" \
    --parameters "AutomationAssumeRole=arn:aws:iam::123456789012:role/
SSMAutomationRole,SourceAmiId=ami-EXAMPLE,IamInstanceProfileName=EC2InstanceRole"
```

Ausgabe:

```
{
```

```
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
}
```

Weitere Informationen finden Sie unter Manuelles Ausführen eines Automation-Workflows im AWS -Systems-Manager-Benutzerhandbuch.

Beispiel 2: So führen Sie ein gemeinsam genutztes Automatisierungsdokument aus

Im folgenden Beispiel `start-automation-execution` wird ein gemeinsam genutztes Automation-Dokument ausgeführt.

```
aws ssm start-automation-execution \
    --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument"
```

Ausgabe:

```
{
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
}
```

Weitere Informationen finden Sie unter Verwenden von geteilten SSM-Dokumenten im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie StartAutomationExecutionin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird ein Dokument ausgeführt, in dem eine Automatisierungsrolle, eine AMI-Quell-ID und eine EC2 Amazon-Instance-Rolle angegeben sind.

```
Start-SSMAutomationExecution -DocumentName AWS-UpdateLinuxAmi -
Parameter @{'AutomationAssumeRole'='arn:aws:iam::123456789012:role/
SSMAutomationRole';'SourceAmiId'='ami-
f173cc91';'InstanceIamRole'='EC2InstanceRole'}
```

Ausgabe:

```
3a532a4f-0382-11e7-9df7-6f11185f6dd1
```

- Einzelheiten zur API finden Sie unter StartAutomationExecution AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird ein Dokument ausgeführt, in dem eine Automatisierungsrolle, eine AMI-Quell-ID und eine EC2 Amazon-Instance-Rolle angegeben sind.

```
Start-SSMAutomationExecution -DocumentName AWS-UpdateLinuxAmi -
Parameter @{'AutomationAssumeRole'='arn:aws:iam::123456789012:role/
SSMAutomationRole';'SourceAmiId'='ami-
f173cc91';'InstanceIamRole'='EC2InstanceRole'}
```

Ausgabe:

```
3a532a4f-0382-11e7-9df7-6f11185f6dd1
```

- Einzelheiten zur API finden Sie unter StartAutomationExecution AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`StartSession`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `StartSession` verwendet wird.

CLI

AWS CLI

Beispiel 1: So starten Sie eine Session-Manager-Sitzung

Dieses `start-session`-Beispiel initiiert eine Verbindung zu einer Instance für eine Session-Manager-Sitzung. Beachten Sie, dass für diesen interaktiven Befehl das Session-Manager-Plugin auf dem Client-Computer installiert sein muss, der den Anruf durchführt.

```
aws ssm start-session \
    --target "i-1234567890abcdef0"
```

Ausgabe:

```
Starting session with SessionId: Jane-Roe-07a16060613c408b5
```

Beispiel 2: So starten Sie eine Session-Manager-Sitzung mit SSH

Dieses `start-session`-Beispiel initiiert eine Verbindung zu einer Instance für eine Session-Manager-Sitzung mit SSH. Beachten Sie, dass für diesen interaktiven Befehl das Session Manager-Plug-In auf dem Client-Computer installiert sein muss, der den Aufruf durchführt, und dass der Befehl den Standardbenutzer auf der Instanz verwendet, z. B. `ec2-user` für EC2 Linux-Instanzen.

```
ssh -i /path/my-key-pair.pem ec2-user@i-02573cafcfEXAMPLE
```

Ausgabe:

```
Starting session with SessionId: ec2-user-07a16060613c408b5
```

Weitere Informationen finden Sie unter Starten einer Sitzung und Installieren des Session Manager-Plug-ins für die AWS CLI im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie StartSessionin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird eine Verbindung zu einem Ziel für eine Session-Manager-Sitzung initiiert, wodurch die Portweiterleitung aktiviert wird.

```
Start-SSMSession -Target 'i-064578e5e7454488f' -DocumentName 'AWS-
StartPortForwardingSession' -Parameter @{ localPortNumber = '8080'; portNumber =
 '80' }
```

Ausgabe:

```
SessionId    StreamUrl
----------    ----------
random-id0    wss://ssmmessages.amazonaws.com/v1/data-channel/random-id
```

- Einzelheiten zur API finden Sie unter StartSession AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird eine Verbindung zu einem Ziel für eine Session-Manager-Sitzung initiiert, wodurch die Portweiterleitung aktiviert wird. Hinweis: Sie müssen den Region-Parameter hinzufügen, falls er nicht bereits mit AWSCredentials festgelegt wurde.

```
Start-SSMSession -Target 'i-064578e5e7454488f' -DocumentName 'AWS-
StartPortForwardingSession' -Parameter @{ localPortNumber = '8080'; portNumber =
 '80' } -Region 'us-west-1'
```

Ausgabe:

```
Starting session with SessionId: testUser-xi4glew849asyeryde34u4dfsdfy
Port 8080 opened for sessionId testUser-xi4glew849asyeryde34u4dfsdfy.
Waiting for connections...
```

Beispiel 2: Hier wird eine interaktive Sitzung mit einer angegebenen Instance für eine Session-Manager-Sitzung erstellt.

```
Start-SSMSession -Target 'i-1234567890abcdef0' -Region 'us-west-1'
```

Ausgabe:

```
Starting session with SessionId : testUser-xi4glew849asyeryde34u4dfsdfy
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.



Install the latest PowerShell for new features and improvements!


PS C:\Windows\system32> whoami
```

```
ec2amaz-fnsdrwv\ec2-test-user
PS C:\Windows\system32>
```

Beispiel 3: Dieses Beispiel erstellt eine Sitzung, ohne eine Verbindung herzustellen, und gibt die TokenValue Eigenschaften SessionId, und zurück StreamUrl, die für die Verbindung mit der Sitzung erforderlich sind.

```
Start-SSMSession -Target 'i-1234567890abcdef0' -Region 'us-west-1' -
DisablePluginInvocation
```

Ausgabe:

```
SessionId        : testUser-xi4glew849asyeryde34u4dfsdfy
StreamUrl        : {StreamUrl value redacted}
TokenValue       : {Token value redacted}
ContentLength    : 1207
HttpStatusCode   : OK
```

- Einzelheiten zur API finden Sie unter StartSession AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`StopAutomationExecution`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `StopAutomationExecution` verwendet wird.

CLI

AWS CLI

So können Sie eine Automatisierungsausführung beenden

Im folgenden Beispiel `stop-automation-execution` wird ein Automation-Dokument gestoppt.

```
aws ssm stop-automation-execution
```

```
    --automation-execution-id "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Manuelles Ausführen eines Automation-Workflows im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie StopAutomationExecutionin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird eine Automation-Ausführung gestoppt. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Stop-SSMAutomationExecution -AutomationExecutionId "4105a4fc-
f944-11e6-9d32-8fb2db27a909"
```

- Einzelheiten zur API finden Sie unter StopAutomationExecution AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird eine Automation-Ausführung gestoppt. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Stop-SSMAutomationExecution -AutomationExecutionId "4105a4fc-
f944-11e6-9d32-8fb2db27a909"
```

- Einzelheiten zur API finden Sie unter StopAutomationExecution AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`UpdateAssociation`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `UpdateAssociation` verwendet wird.

CLI

AWS CLI

Beispiel 1: Um eine Dokumentverknüpfung zu aktualisieren

Das folgende `update-association`-Beispiel aktualisiert eine Zuordnung mit einer neuen Dokumentversion.

```
aws ssm update-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --document-version "\$LATEST"
```

Ausgabe:

```
{
    "AssociationDescription": {
        "Name": "AWS-UpdateSSMAgent",
        "AssociationVersion": "2",
        "Date": 1550508093.293,
        "LastUpdateAssociationDate": 1550508106.596,
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "DocumentVersion": "$LATEST",
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "tag:Name",
                "Values": [
                    "Linux"
                ]
            }
        ],
        "LastExecutionDate": 1550508094.879,
        "LastSuccessfulExecutionDate": 1550508094.879
    }
}
```

Weitere Informationen finden Sie unter Bearbeiten und Erstellen einer neuen Version einer Zuordnung im AWS -Systems-Manager-Benutzerhandbuch.

Beispiel 2: So aktualisieren Sie den Zeitplanausdruck einer Zuordnung

Im folgenden `update-association`-Beispiel wird der Zeitplanausdruck für die angegebene Zuordnung aktualisiert.

```
aws ssm update-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --schedule-expression "cron(0 0 0/4 1/1 * ? *)"
```

Ausgabe:

```
{
    "AssociationDescription": {
        "Name": "AWS-HelloWorld",
        "AssociationVersion": "2",
        "Date": "2021-02-08T13:54:19.203000-08:00",
        "LastUpdateAssociationDate": "2021-06-29T11:51:07.933000-07:00",
        "Overview": {
            "Status": "Pending",
            "DetailedStatus": "Creating"
        },
        "DocumentVersion": "$DEFAULT",
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "aws:NoOpAutomationTag",
                "Values": [
                    "AWS-NoOpAutomationTarget-Value"
                ]
            }
        ],
        "ScheduleExpression": "cron(0 0 0/4 1/1 * ? *)",
        "LastExecutionDate": "2021-06-26T19:00:48.110000-07:00",
        "ApplyOnlyAtCronInterval": false
    }
}
```

Weitere Informationen finden Sie unter Bearbeiten und Erstellen einer neuen Version einer Zuordnung im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie UpdateAssociationin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird eine Verknüpfung mit einer neuen Dokumentversion aktualisiert.

```
Update-SSMAssociation -AssociationId "93285663-92df-44cb-9f26-2292d4ecc439" -
DocumentVersion "1"
```

Ausgabe:

```
Name                 : AWS-UpdateSSMAgent
InstanceId           :
Date                 : 3/1/2017 6:22:21 PM
Status.Name          :
Status.Date          :
Status.Message       :
Status.AdditionalInfo :
```

- Einzelheiten zur API finden Sie unter UpdateAssociation AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird eine Verknüpfung mit einer neuen Dokumentversion aktualisiert.

```
Update-SSMAssociation -AssociationId "93285663-92df-44cb-9f26-2292d4ecc439" -
DocumentVersion "1"
```

Ausgabe:

```
Name                 : AWS-UpdateSSMAgent
InstanceId           :
Date                 : 3/1/2017 6:22:21 PM
Status.Name          :
Status.Date          :
Status.Message       :
Status.AdditionalInfo :
```

- Einzelheiten zur API finden Sie unter UpdateAssociation AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`UpdateAssociationStatus`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `UpdateAssociationStatus` verwendet wird.

CLI

AWS CLI

So aktualisieren Sie den Zuordnungsstatus

Im folgenden Beispiel `update-association-status` wird der Zuordnungsstatus der Verknüpfung zwischen einer Instance und einem Dokument aktualisiert.

```
aws ssm update-association-status \
    --name "AWS-UpdateSSMAgent" \
    --instance-id "i-1234567890abcdef0" \
    --association-
status "Date=1424421071.939,Name=Pending,Message=temp_status_change,AdditionalInfo=Additi
Config-Needed"
```

Ausgabe:

```
{
    "AssociationDescription": {
        "Name": "AWS-UpdateSSMAgent",
        "InstanceId": "i-1234567890abcdef0",
        "AssociationVersion": "1",
        "Date": 1550507529.604,
        "LastUpdateAssociationDate": 1550507806.974,
        "Status": {
            "Date": 1424421071.0,
            "Name": "Pending",
            "Message": "temp_status_change",
            "AdditionalInfo": "Additional-Config-Needed"
        },
```

```
        "Overview": {
            "Status": "Success",
            "AssociationStatusAggregatedCount": {
                "Success": 1
            }
        },
        "DocumentVersion": "$DEFAULT",
        "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
        "Targets": [
            {
                "Key": "InstanceIds",
                "Values": [
                    "i-1234567890abcdef0"
                ]
            }
        ],
        "LastExecutionDate": 1550507808.0,
        "LastSuccessfulExecutionDate": 1550507808.0
    }
}
```

Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie UpdateAssociationStatusin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird der Zuordnungsstatus der Zuordnung zwischen einer Instance und einem Konfigurationsdokument aktualisiert.

```
Update-SSMAssociationStatus -Name "AWS-UpdateSSMAgent" -InstanceId
 "i-0000293ffd8c57862" -AssociationStatus_Date "2015-02-20T08:31:11Z"
 -AssociationStatus_Name "Pending" -AssociationStatus_Message
 "temporary_status_change" -AssociationStatus_AdditionalInfo "Additional-Config-
Needed"
```

Ausgabe:

```
Name                   : AWS-UpdateSSMAgent
```

```
InstanceId            : i-0000293ffd8c57862
Date                  : 2/23/2017 6:55:22 PM
Status.Name           : Pending
Status.Date           : 2/20/2015 8:31:11 AM
Status.Message        : temporary_status_change
Status.AdditionalInfo : Additional-Config-Needed
```

- Einzelheiten zur API finden Sie unter UpdateAssociationStatus AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird der Zuordnungsstatus der Zuordnung zwischen einer Instance und einem Konfigurationsdokument aktualisiert.

```
Update-SSMAssociationStatus -Name "AWS-UpdateSSMAgent" -InstanceId
 "i-0000293ffd8c57862" -AssociationStatus_Date "2015-02-20T08:31:11Z"
 -AssociationStatus_Name "Pending" -AssociationStatus_Message
 "temporary_status_change" -AssociationStatus_AdditionalInfo "Additional-Config-
Needed"
```

Ausgabe:

```
Name                  : AWS-UpdateSSMAgent
InstanceId            : i-0000293ffd8c57862
Date                  : 2/23/2017 6:55:22 PM
Status.Name           : Pending
Status.Date           : 2/20/2015 8:31:11 AM
Status.Message        : temporary_status_change
Status.AdditionalInfo : Additional-Config-Needed
```

- Einzelheiten zur API finden Sie unter UpdateAssociationStatus AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`UpdateDocument`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `UpdateDocument` verwendet wird.

CLI

AWS CLI

Um eine neue Version eines Dokuments zu erstellen

Das folgende `update-document`-Beispiel erstellt eine neue Version eines Dokuments, wenn es auf einem Windows-Computer ausgeführt wird. Das von `--document` angegebene Dokument muss im JSON-Format vorliegen. Beachten Sie, dass `file://` darauf verwiesen werden muss, gefolgt vom Pfad der Inhaltsdatei. Wegen des $ am Anfang des `--document-version`-Parameters müssen Sie den Wert unter Windows in Anführungszeichen setzen. Unter Linux, macOS oder an einer PowerShell Eingabeaufforderung müssen Sie den Wert in einfache Anführungszeichen setzen.

Windows-Version:

```
aws ssm update-document \
    --name "RunShellScript" \
    --content "file://RunShellScript.json" \
    --document-version "$LATEST"
```

Linux/Mac-Version:

```
aws ssm update-document \
    --name "RunShellScript" \
    --content "file://RunShellScript.json" \
    --document-version '$LATEST'
```

Ausgabe:

```
{
  "DocumentDescription": {
      "Status": "Updating",
      "Hash": "f775e5df4904c6fa46686c4722fae9de1950dace25cd9608ff8d622046b68d9b",
      "Name": "RunShellScript",
      "Parameters": [
          {
              "Type": "StringList",
              "Name": "commands",
              "Description": "(Required) Specify a shell script or a command to
 run."
```

```
            }
        ],
        "DocumentType": "Command",
        "PlatformTypes": [
            "Linux"
        ],
        "DocumentVersion": "2",
        "HashType": "Sha256",
        "CreatedDate": 1487899655.152,
        "Owner": "809632081692",
        "SchemaVersion": "2.0",
        "DefaultVersion": "1",
        "LatestVersion": "2",
        "Description": "Run an updated script"
    }
}
```

- Einzelheiten zur API finden Sie UpdateDocumentin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dadurch wird eine neue Version eines Dokuments mit dem aktualisierten Inhalt der von Ihnen angegebenen JSON-Datei erstellt. Das Dokument muss im JSON-Format vorliegen. Sie können die Dokumentversion mit dem Cmdlet „Get- SSMDocumentVersionList“ abrufen.

```
Update-SSMDocument -Name RunShellScript -DocumentVersion "1" -Content (Get-
Content -Raw "c:\temp\RunShellScript.json")
```

Ausgabe:

```
CreatedDate     : 3/1/2017 2:59:17 AM
DefaultVersion  : 1
Description     : Run an updated script
DocumentType    : Command
DocumentVersion : 2
Hash            :
 1d5ce820e999ff051eb4841ed887593daf77120fd76cae0d18a53cc42e4e22c1
HashType        : Sha256
LatestVersion   : 2
Name            : RunShellScript
```

```
Owner           : 809632081692
Parameters      : {commands}
PlatformTypes   : {Linux}
SchemaVersion   : 2.0
Sha1            :
Status          : Updating
```

- Einzelheiten zur API finden Sie unter UpdateDocument AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dadurch wird eine neue Version eines Dokuments mit dem aktualisierten Inhalt der von Ihnen angegebenen JSON-Datei erstellt. Das Dokument muss im JSON-Format vorliegen. Sie können die Dokumentversion mit dem Cmdlet „Get- SSMDocumentVersionList“ abrufen.

```
Update-SSMDocument -Name RunShellScript -DocumentVersion "1" -Content (Get-
Content -Raw "c:\temp\RunShellScript.json")
```

Ausgabe:

```
CreatedDate     : 3/1/2017 2:59:17 AM
DefaultVersion  : 1
Description     : Run an updated script
DocumentType    : Command
DocumentVersion : 2
Hash            :
 1d5ce820e999ff051eb4841ed887593daf77120fd76cae0d18a53cc42e4e22c1
HashType        : Sha256
LatestVersion   : 2
Name            : RunShellScript
Owner           : 809632081692
Parameters      : {commands}
PlatformTypes   : {Linux}
SchemaVersion   : 2.0
Sha1            :
Status          : Updating
```

- Einzelheiten zur API finden Sie unter UpdateDocument AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`UpdateDocumentDefaultVersion`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `UpdateDocumentDefaultVersion` verwendet wird.

CLI

AWS CLI

So aktualisieren Sie die Standardversion eines Dokuments

Im folgenden `update-document-default-version`-Beispiel wird die Standardversion eines Systems-Manager-Dokuments aktualisiert.

```
aws ssm update-document-default-version \
    --name "Example" \
    --document-version "2"
```

Ausgabe:

```
{
    "Description": {
        "Name": "Example",
        "DefaultVersion": "2"
    }
}
```

Weitere Informationen finden Sie unter Schreiben von SSM-Dokumentinhalten im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie UpdateDocumentDefaultVersionin der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: Dadurch wird die Standardversion eines Dokuments aktualisiert. Sie können die verfügbaren Dokumentversionen mit dem Cmdlet „Get- SSMDocumentVersionList“ abrufen.

```
Update-SSMDocumentDefaultVersion -Name "RunShellScript" -DocumentVersion "2"
```

Ausgabe:

```
DefaultVersion Name
-------------- ----
2              RunShellScript
```

- Einzelheiten zur API finden Sie unter UpdateDocumentDefaultVersion AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: Dadurch wird die Standardversion eines Dokuments aktualisiert. Sie können die verfügbaren Dokumentversionen mit dem Cmdlet „Get- SSMDocumentVersionList“ abrufen.

```
Update-SSMDocumentDefaultVersion -Name "RunShellScript" -DocumentVersion "2"
```

Ausgabe:

```
DefaultVersion Name
-------------- ----
2              RunShellScript
```

- Einzelheiten zur API finden Sie unter UpdateDocumentDefaultVersion AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`UpdateMaintenanceWindow`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `UpdateMaintenanceWindow` verwendet wird.

CLI

AWS CLI

Beispiel 1: So aktualisieren Sie ein Wartungsfenster

Im folgenden Beispiel `update-maintenance-window` wird der Name eines Wartungsfensters aktualisiert.

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --name "My-Renamed-MW"
```

Ausgabe:

```
{
    "Cutoff": 1,
    "Name": "My-Renamed-MW",
    "Schedule": "cron(0 16 ? * TUE *)",
    "Enabled": true,
    "AllowUnassociatedTargets": true,
    "WindowId": "mw-1a2b3c4d5e6f7g8h9",
    "Duration": 4
}
```

Beispiel 2: So deaktivieren Sie ein Wartungsfenster

Das folgende `update-maintenance-window`-Beispiel deaktiviert ein Wartungsfenster.

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --no-enabled
```

Beispiel 3: So aktivieren Sie ein Wartungsfenster

Das folgende `update-maintenance-window`-Beispiel aktiviert ein Wartungsfenster.

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --enabled
```

Weitere Informationen finden Sie unter Aktualisieren eines Wartungsfensters (AWS CLI) im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie UpdateMaintenanceWindowunter AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
   /**
    * Updates an SSM maintenance window asynchronously.
    *
    * @param id The ID of the maintenance window to update.
    * @param name The new name for the maintenance window.
    * <p>
    * This method initiates an asynchronous request to update an SSM maintenance
window.
    * If the request is successful, it prints a success message.
    * If an exception occurs, it handles the error appropriately.
    */
   public void updateSSMMaintenanceWindow(String id, String name) throws
SsmException {
       UpdateMaintenanceWindowRequest updateRequest =
UpdateMaintenanceWindowRequest.builder()
               .windowId(id)
               .allowUnassociatedTargets(true)
               .duration(24)
               .enabled(true)
               .name(name)
               .schedule("cron(0 0 ? * MON *)")
               .build();

       CompletableFuture<UpdateMaintenanceWindowResponse> future =
getAsyncClient().updateMaintenanceWindow(updateRequest);
       future.whenComplete((response, ex) -> {
           if (response != null) {
               System.out.println("The SSM maintenance window was successfully
updated");
           } else {
               Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
```

```
                if (cause instanceof SsmException) {
                    throw new CompletionException(cause);
                } else {
                    throw new RuntimeException(cause);
                }
            }
        }).join();
    }
```

- Einzelheiten zur API finden Sie UpdateMaintenanceWindowin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import { UpdateMaintenanceWindowCommand, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * Update an SSM maintenance window.
 * @param {{ windowId: string, allowUnassociatedTargets?: boolean, duration?:
 number, enabled?: boolean, name?: string, schedule?: string }}
 */
export const main = async ({
  windowId,
  allowUnassociatedTargets = undefined, //Allow the maintenance window to run on
 managed nodes, even if you haven't registered those nodes as targets.
  duration = undefined, //The duration of the maintenance window in hours.
  enabled = undefined,
  name = undefined,
  schedule = undefined, //The schedule of the maintenance window in the form of a
 cron or rate expression.
}) => {
```

```
  const client = new SSMClient({});
  try {
    const { opsItemArn, opsItemId } = await client.send(
      new UpdateMaintenanceWindowCommand({
        WindowId: windowId,
        AllowUnassociatedTargets: allowUnassociatedTargets,
        Duration: duration,
        Enabled: enabled,
        Name: name,
        Schedule: schedule,
      }),
    );
    console.log("Maintenance window updated.");
    return { OpsItemArn: opsItemArn, OpsItemId: opsItemId };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ValidationError") {
      console.warn(`${caught.message}. Are these values correct?`);
    } else {
      throw caught;
    }
  }
};
```

- Einzelheiten zur API finden Sie UpdateMaintenanceWindowin der AWS SDK für JavaScript API-Referenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird der Name eines Wartungsfensters aktualisiert.

```
Update-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d" -Name "My-Renamed-
MW"
```

Ausgabe:

```
AllowUnassociatedTargets : False
Cutoff                   : 1
Duration                 : 2
Enabled                  : True
```

```
Name                        : My-Renamed-MW
Schedule                    : cron(0 */30 * * * ? *)
WindowId                    : mw-03eb9db42890fb82d
```

Beispiel 2: In diesem Beispiel wird ein Wartungsfenster aktiviert.

```
Update-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d" -Enabled $true
```

Ausgabe:

```
AllowUnassociatedTargets : False
Cutoff                   : 1
Duration                 : 2
Enabled                  : True
Name                     : My-Renamed-MW
Schedule                 : cron(0 */30 * * * ? *)
WindowId                 : mw-03eb9db42890fb82d
```

Beispiel 3: In diesem Beispiel wird ein Wartungsfenster deaktiviert.

```
Update-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d" -Enabled $false
```

Ausgabe:

```
AllowUnassociatedTargets : False
Cutoff                   : 1
Duration                 : 2
Enabled                  : False
Name                     : My-Renamed-MW
Schedule                 : cron(0 */30 * * * ? *)
WindowId                 : mw-03eb9db42890fb82d
```

- Einzelheiten zur API finden Sie unter UpdateMaintenanceWindow AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird der Name eines Wartungsfensters aktualisiert.

```
Update-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d" -Name "My-Renamed-
MW"
```

Ausgabe:

```
AllowUnassociatedTargets : False
Cutoff                   : 1
Duration                 : 2
Enabled                  : True
Name                     : My-Renamed-MW
Schedule                 : cron(0 */30 * * * ? *)
WindowId                 : mw-03eb9db42890fb82d
```

Beispiel 2: In diesem Beispiel wird ein Wartungsfenster aktiviert.

```
Update-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d" -Enabled $true
```

Ausgabe:

```
AllowUnassociatedTargets : False
Cutoff                   : 1
Duration                 : 2
Enabled                  : True
Name                     : My-Renamed-MW
Schedule                 : cron(0 */30 * * * ? *)
WindowId                 : mw-03eb9db42890fb82d
```

Beispiel 3: In diesem Beispiel wird ein Wartungsfenster deaktiviert.

```
Update-SSMMaintenanceWindow -WindowId "mw-03eb9db42890fb82d" -Enabled $false
```

Ausgabe:

```
AllowUnassociatedTargets : False
Cutoff                   : 1
Duration                 : 2
Enabled                  : False
Name                     : My-Renamed-MW
Schedule                 : cron(0 */30 * * * ? *)
WindowId                 : mw-03eb9db42890fb82d
```

- Einzelheiten zur API finden Sie unter UpdateMaintenanceWindow AWS -Tools für PowerShellCmdlet-Referenz (V5).

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class MaintenanceWindowWrapper:
    """Encapsulates AWS Systems Manager maintenance window actions."""

    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.window_id = None
        self.name = None

    @classmethod
    def from_client(cls):
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def update(
        self, name, enabled, schedule, duration, cutoff,
 allow_unassociated_targets
    ):
        """
        Update an AWS Systems Manager maintenance window.

        :param name: The name of the maintenance window.
        :param enabled: Whether the maintenance window is enabled to run on
 managed nodes.
        :param schedule: The schedule of the maintenance window.
        :param duration: The duration of the maintenance window.
        :param cutoff: The cutoff time of the maintenance window.
        :param allow_unassociated_targets: Allow the maintenance window to run on
 managed nodes, even
```

```
                                        if you haven't registered those nodes
as targets.
        """
        try:
            self.ssm_client.update_maintenance_window(
                WindowId=self.window_id,
                Name=name,
                Enabled=enabled,
                Schedule=schedule,
                Duration=duration,
                Cutoff=cutoff,
                AllowUnassociatedTargets=allow_unassociated_targets,
            )
            self.name = name
            logger.info("Updated maintenance window %s.", self.window_id)
        except ParamValidationError as error:
            logger.error(
                "Parameter validation error when trying to update maintenance
window %s. Here's why: %s",
                self.window_id,
                error,
            )
            raise
        except ClientError as err:
            logger.error(
                "Couldn't update maintenance window %s. Here's why: %s: %s",
                self.name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Einzelheiten zur API finden Sie UpdateMaintenanceWindowin AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **UpdateManagedInstanceRole** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `UpdateManagedInstanceRole` verwendet wird.

CLI

AWS CLI

Um die IAM-Rolle einer verwalteten Instance zu aktualisieren

Im folgenden Beispiel `update-managed-instance-role` wird das IAM-Instance-Profil einer verwalteten Instance aktualisiert.

```
aws ssm update-managed-instance-role \
    --instance-id "mi-08ab247cdfEXAMPLE" \
    --iam-role "ExampleRole"
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Schritt 4: Erstellen eines Instance-Profils für Systems Manager im AWS -Systems-Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie UpdateManagedInstanceRolein der AWS CLI Befehlsreferenz.

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel wird die Rolle einer verwalteten Instance aktualisiert. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Update-SSMManagedInstanceRole -InstanceId "mi-08ab247cdf1046573" -IamRole
 "AutomationRole"
```

- Einzelheiten zur API finden Sie unter UpdateManagedInstanceRole AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel wird die Rolle einer verwalteten Instance aktualisiert. Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

```
Update-SSMManagedInstanceRole -InstanceId "mi-08ab247cdf1046573" -IamRole
 "AutomationRole"
```

- Einzelheiten zur API finden Sie unter UpdateManagedInstanceRole AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`UpdateOpsItem`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `UpdateOpsItem` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Kennenlernen der Grundlagen

CLI

AWS CLI

Um ein zu aktualisieren OpsItem

Im folgenden `update-ops-item` Beispiel werden die Beschreibung, Priorität und Kategorie für ein aktualisiert OpsItem. Darüber hinaus gibt der Befehl ein SNS-Thema an, an das die Benachrichtigungen gesendet werden, wenn dieses bearbeitet oder geändert OpsItem wird.

```
aws ssm update-ops-item \
    --ops-item-id "oi-287b5EXAMPLE" \
    --description "Primary OpsItem for failover event 2020-01-01-fh398yf" \
    --priority 2 \
    --category "Security" \
    --notifications "Arn=arn:aws:sns:us-east-2:111222333444:my-us-east-2-topic"
```

Ausgabe:

```
This command produces no output.
```

Weitere Informationen finden Sie unter Arbeiten mit OpsItems im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie UpdateOpsItemunter AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    /**
     * Resolves an AWS SSM OpsItem asynchronously.
     *
     * @param opsID The ID of the OpsItem to resolve.
     * <p>
     * This method initiates an asynchronous request to resolve an SSM OpsItem.
     * If an exception occurs, it handles the error appropriately.
     */
    public void resolveOpsItem(String opsID) {
        UpdateOpsItemRequest opsItemRequest = UpdateOpsItemRequest.builder()
                .opsItemId(opsID)
                .status(OpsItemStatus.RESOLVED)
                .build();

        CompletableFuture<Void> future = CompletableFuture.runAsync(() -> {
            getAsyncClient().updateOpsItem(opsItemRequest)
                    .thenAccept(response -> {
                        System.out.println("OpsItem resolved successfully.");
                    })
                    .exceptionally(ex -> {
                        throw new CompletionException(ex);
                    }).join();
        }).exceptionally(ex -> {
            Throwable cause = (ex instanceof CompletionException) ?
ex.getCause() : ex;
            if (cause instanceof SsmException) {
```

```
                throw new RuntimeException("SSM error: " + cause.getMessage(),
 cause);
            } else {
                throw new RuntimeException("Unexpected error: " +
 cause.getMessage(), cause);
            }
        });

        try {
            future.join();
        } catch (CompletionException ex) {
            throw ex.getCause() instanceof RuntimeException ? (RuntimeException)
 ex.getCause() : ex;
        }
    }
```

- Einzelheiten zur API finden Sie UpdateOpsItemin der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import { UpdateOpsItemCommand, SSMClient } from "@aws-sdk/client-ssm";
import { parseArgs } from "node:util";

/**
 * Update an SSM OpsItem.
 * @param {{ opsItemId: string, status?: OpsItemStatus }}
 */
export const main = async ({
  opsItemId,
  status = undefined, // The OpsItem status. Status can be Open, In Progress, or
 Resolved
}) => {
  const client = new SSMClient({});
```

```
  try {
    await client.send(
      new UpdateOpsItemCommand({
        OpsItemId: opsItemId,
        Status: status,
      }),
    );
    console.log("Ops item updated.");
    return { Success: true };
  } catch (caught) {
    if (
      caught instanceof Error &&
      caught.name === "OpsItemLimitExceededException"
    ) {
      console.warn(
        `Couldn't create ops item because you have exceeded your open OpsItem
 limit. ${caught.message}.`,
      );
    } else {
      throw caught;
    }
  }
};
```

- Einzelheiten zur API finden Sie UpdateOpsItemin der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class OpsItemWrapper:
    """Encapsulates AWS Systems Manager OpsItem actions."""
```

```
    def __init__(self, ssm_client):
        """
        :param ssm_client: A Boto3 Systems Manager client.
        """
        self.ssm_client = ssm_client
        self.id = None

    @classmethod
    def from_client(cls):
        """
        :return: A OpsItemWrapper instance.
        """
        ssm_client = boto3.client("ssm")
        return cls(ssm_client)


    def update(self, title=None, description=None, status=None):
        """
        Update an OpsItem.

        :param title: The new OpsItem title.
        :param description: The new OpsItem description.
        :param status: The new OpsItem status.
        :return:
        """
        args = dict(OpsItemId=self.id)
        if title is not None:
            args["Title"] = title
        if description is not None:
            args["Description"] = description
        if status is not None:
            args["Status"] = status
        try:
            self.ssm_client.update_ops_item(**args)
        except ClientError as err:
            logger.error(
                "Couldn't update ops item %s. Here's why: %s: %s",
                self.id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Einzelheiten zur API finden Sie UpdateOpsItemin AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung von **`UpdatePatchBaseline`** mit einer CLI

Die folgenden Code-Beispiele zeigen, wie `UpdatePatchBaseline` verwendet wird.

CLI

AWS CLI

Beispiel 1: So können Sie eine Patch-Baseline aktualisieren

Im folgenden `update-patch-baseline`-Beispiel werden der angegebenen Patch-Baseline die beiden angegebenen Patches als abgelehnt und ein Patch als genehmigt hinzugefügt.

```
aws ssm update-patch-baseline \
        --baseline-id "pb-0123456789abcdef0" \
        --rejected-patches "KB2032276" "MS10-048" \
        --approved-patches "KB2124261"
```

Ausgabe:

```
{
    "BaselineId": "pb-0123456789abcdef0",
    "Name": "WindowsPatching",
    "OperatingSystem": "WINDOWS",
    "GlobalFilters": {
        "PatchFilters": []
    },
    "ApprovalRules": {
        "PatchRules": [
            {
                "PatchFilterGroup": {
                    "PatchFilters": [
```

```
                        {
                            "Key": "PRODUCT",
                            "Values": [
                                "WindowsServer2016"
                            ]
                        }
                    ]
                },
                "ComplianceLevel": "CRITICAL",
                "ApproveAfterDays": 0,
                "EnableNonSecurity": false
            }
        ]
    },
    "ApprovedPatches": [
        "KB2124261"
    ],
    "ApprovedPatchesComplianceLevel": "UNSPECIFIED",
    "ApprovedPatchesEnableNonSecurity": false,
    "RejectedPatches": [
        "KB2032276",
        "MS10-048"
    ],
    "RejectedPatchesAction": "ALLOW_AS_DEPENDENCY",
    "CreatedDate": 1550244180.465,
    "ModifiedDate": 1550244180.465,
    "Description": "Patches for Windows Servers",
    "Sources": []
}
```

Beispiel 2: So benennen Sie eine Patch-Baseline um

Im folgenden Beispiel `update-patch-baseline` wird die angegebene Patch-Baseline umbenannt.

```
aws ssm update-patch-baseline \
    --baseline-id "pb-0713accee01234567" \
    --name "Windows-Server-2012-R2-Important-and-Critical-Security-Updates"
```

Weitere Informationen finden Sie unter Aktualisieren oder Löschen einer Patch-Baseline < https://docs.aws.amazon.com/systems-manager/ latest/userguide/patch - baseline-update-or-delete .html>`__ im AWS Systems Manager Manager-Benutzerhandbuch.

- Einzelheiten zur API finden Sie in der Befehlsreferenz UpdatePatchBaseline.AWS CLI

PowerShell

Tools für PowerShell V4

Beispiel 1: In diesem Beispiel werden einer vorhandenen Patch-Baseline zwei Patches als abgelehnt und ein Patch als genehmigt hinzugefügt.

```
Update-SSMPatchBaseline -BaselineId "pb-03da896ca3b68b639" -RejectedPatch
 "KB2032276","MS10-048" -ApprovedPatch "KB2124261"
```

Ausgabe:

```
ApprovalRules   : Amazon.SimpleSystemsManagement.Model.PatchRuleGroup
ApprovedPatches : {KB2124261}
BaselineId      : pb-03da896ca3b68b639
CreatedDate     : 3/3/2017 5:02:19 PM
Description     : Baseline containing all updates approved for production systems
GlobalFilters   : Amazon.SimpleSystemsManagement.Model.PatchFilterGroup
ModifiedDate    : 3/3/2017 5:22:10 PM
Name            : Production-Baseline
RejectedPatches : {KB2032276, MS10-048}
```

- Einzelheiten zur API finden Sie unter UpdatePatchBaseline AWS -Tools für PowerShellCmdlet-Referenz (V4).

Tools für V5 PowerShell

Beispiel 1: In diesem Beispiel werden einer vorhandenen Patch-Baseline zwei Patches als abgelehnt und ein Patch als genehmigt hinzugefügt.

```
Update-SSMPatchBaseline -BaselineId "pb-03da896ca3b68b639" -RejectedPatch
 "KB2032276","MS10-048" -ApprovedPatch "KB2124261"
```

Ausgabe:

```
ApprovalRules   : Amazon.SimpleSystemsManagement.Model.PatchRuleGroup
ApprovedPatches : {KB2124261}
BaselineId      : pb-03da896ca3b68b639
CreatedDate     : 3/3/2017 5:02:19 PM
```

```
Description     : Baseline containing all updates approved for production systems
GlobalFilters   : Amazon.SimpleSystemsManagement.Model.PatchFilterGroup
ModifiedDate    : 3/3/2017 5:22:10 PM
Name            : Production-Baseline
RejectedPatches : {KB2032276, MS10-048}
```

- Einzelheiten zur API finden Sie unter UpdatePatchBaseline AWS -Tools für PowerShellCmdlet-Referenz (V5).

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. Verwenden dieses Dienstes mit einem AWS SDK Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Einloggen und Überwachen AWS Systems Manager

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit AWS Systems Manager und Leistung Ihrer AWS Lösungen. Sie sollten Überwachungsdaten aus allen Teilen Ihrer AWS Lösung sammeln, damit Sie einen etappenübergreifenden Ausfall debuggen können. Bevor Sie mit der Überwachung von Systems Manager beginnen, sollten Sie einen Überwachungsplan mit Antworten auf die folgenden Fragen erstellen:

- Was sind Ihre Ziele bei der Überwachung?
- Welche Ressourcen werden überwacht?
- Wie oft werden diese Ressourcen überwacht?
- Welche Überwachungstools werden verwendet?
- Wer führt die Überwachungsaufgaben aus?
- Wer soll benachrichtigt werden, wenn Fehler auftreten?

Nachdem Sie Ihre Überwachungsziele festgelegt und Ihren Überwachungsplan erstellt haben, legen Sie im nächsten Schritt einen Ausgangswert für normale Systems Manager-Leistung in Ihrer Umgebung fest. Sie sollten die Systems Manager-Leistung zu verschiedenen Zeiten und unter verschiedenen Belastungsbedingungen messen. Wenn Sie Systems Manager überwachen, sollten Sie einen Verlauf der von Ihnen gesammelten Überwachungsdaten speichern. Sie können die aktuelle Systems Manager-Leistung mit diesen historischen Daten zur Identifikation normaler Leistungsmuster und Leistungsanomalien sowie zur Erstellung von Verfahren für deren Handhabung vergleichen.

Beispielsweise können Sie die Erfolge oder Fehlschläge von Vorgängen wie Automation-Workflows, die Anwendung von Patch-Baselines, Wartungsfensterereignisse und die Konfigurations-Compliance überwachen. Automation ist ein Tool in AWS Systems Manager.

Sie können auch die CPU-Auslastung und die Festplatte I/O, and network utilization of your managed nodes. When performance falls outside your established baseline, you might need to reconfigure or optimize the node to reduce CPU utilization, improve disk I/O überwachen oder den Netzwerkverkehr reduzieren. Weitere Informationen zur Überwachung von EC2 Instances finden Sie unter Monitor Amazon EC2 im EC2 Amazon-Benutzerhandbuch.

Themen

- Tools zur Überwachung
- Senden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent)
- SSM AgentProtokolle an CloudWatch Logs senden
- Überwachung der Ereignisse Ihrer Änderungsanfragen
- Überwachung Ihrer Automatisierungen
- Überwachen von Run Command-Metriken mit Amazon CloudWatch
- Protokollierung von AWS Systems Manager-API-Aufrufen mit AWS CloudTrail
- Protokollierung der Automation-Aktionsausgabe mit CloudWatch Logs
- Konfigurieren von Amazon CloudWatch Logs für Run Command
- Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge
- Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen

Tools zur Überwachung

Der Inhalt dieses Kapitels enthält Informationen zur Verwendung der Tools, die zur Überwachung Ihres Systems Manager und anderer AWS Ressourcen zur Verfügung stehen. Eine vollständigere Liste der Tools finden Sie unter Protokollieren und Überwachen in AWS Systems Manager.

Senden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent)

Sie können den CloudWatch Amazon-Agenten konfigurieren und verwenden, um Metriken und Protokolle von Ihren Knoten zu sammeln, anstatt AWS Systems Manager Agent (SSM Agent) für diese Aufgaben zu verwenden. Der CloudWatch Agent ermöglicht es Ihnen, mehr Metriken zu EC2 Instances zu sammeln, als mit ihm verfügbar sindSSM Agent. Darüber hinaus können Sie mit dem CloudWatch Agenten Metriken von lokalen Servern sammeln.

Sie können die Agentenkonfigurationseinstellungen auch im Systems Manager speichern, Parameter Store um sie mit dem CloudWatch Agenten zu verwenden. Parameter Storeist ein Tool in AWS Systems Manager.

> **Note**
>
> AWS Systems Manager unterstützt nur unter 64-Bit-Versionen von Windows die Migration vom SSM Agent Unified CloudWatch Agent zum Sammeln von Protokollen und Messdaten.

Informationen zur Einrichtung des Unified CloudWatch Agents auf anderen Betriebssystemen und vollständige Informationen zur Verwendung des CloudWatch Agenten finden Sie unter Erfassung von Metriken und Protokollen von EC2 Amazon-Instances und lokalen Servern mit dem CloudWatch Agenten im CloudWatch Amazon-Benutzerhandbuch.
Sie können den CloudWatch Agenten auf anderen unterstützten Betriebssystemen verwenden, aber Sie können Systems Manager nicht verwenden, um eine Tool-Migration durchzuführen.

SSM Agent schreibt Informationen zu Ausführungen, geplanten Aktionen, Fehlern und dem Zustandsstatus in Protokolldateien auf jedem Knoten. Die manuelle Verbindung mit einem Knoten, um Protokolldateien anzuzeigen und zum Beheben eines Problems mit dem SSM Agent ist zeitaufwendig. Für eine effizientere Knotenüberwachung können Sie entweder SSM Agent sich selbst oder den CloudWatch Agenten so konfigurieren, dass er diese Protokolldaten an Amazon CloudWatch Logs sendet.

Important

Der Unified CloudWatch Agent wurde SSM Agent als Tool zum Senden von Protokolldaten an Amazon CloudWatch Logs ersetzt. Das SSM Agent-aws:cloudWatch-Plugin wird nicht unterstützt. Wir empfehlen, nur den Unified CloudWatch Agent für Ihre Protokollerfassungsprozesse zu verwenden. Weitere Informationen finden Sie unter den folgenden Themen:

- Senden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent)
- Migrieren Sie die Erfassung von Windows Server-Knotenprotokollen auf den CloudWatch Agenten
- Erfassung von Metriken, Protokollen und Traces mit dem CloudWatch Agenten im CloudWatch Amazon-Benutzerhandbuch.

Mithilfe von CloudWatch Logs können Sie Protokolldaten in Echtzeit überwachen, Protokolldaten suchen und filtern, indem Sie einen oder mehrere Metrikfilter erstellen, und historische Daten archivieren und abrufen, wenn Sie sie benötigen. Weitere Informationen zu CloudWatch Logs finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.

Die Konfiguration eines Agenten zum Senden von Protokolldaten an Amazon CloudWatch Logs bietet die folgenden Vorteile:

- Zentrale Speicherung von Protokolldateien alle SSM Agent-Protokolldateien.
- Schnellerer Zugriff auf Dateien zur Fehleranalyse.
- Unbegrenzte Protokolldatei-Aufbewahrung (konfigurierbar).
- Die Verwaltung und der Zugriff auf Protokolle ist unabhängig vom Status des Knotens möglich.
- Zugriff auf andere CloudWatch Funktionen wie Metriken und Alarme.

Informationen zur Überwachung von Session Manager-Aktivitäten finden Sie unter Protokollieren von Sitzungsaktivitäten und Protokollierung von Sitzungen aktivieren und deaktivieren.

Migrieren Sie die Erfassung von Windows Server-Knotenprotokollen auf den CloudWatch Agenten

Wenn Sie SSM Agent auf unterstützten Windows Server Knoten SSM Agent Protokolldateien an Amazon CloudWatch Logs senden, können Sie Systems Manager verwenden, SSM Agent um vom CloudWatch Agenten als Protokollerfassungstool zu migrieren und Ihre Konfigurationseinstellungen zu migrieren.

Der CloudWatch Agent wird auf 32-Bit-Versionen von nicht unterstütztWindows Server.

Bei EC2 64-Bit-Instances für Windows Server können Sie die Migration zum CloudWatch Agenten automatisch oder manuell durchführen. Bei On-Premises-Servern und virtuellen Maschinen muss der Prozess manuell ausgeführt werden.

> **Note**
>
> Während des Migrationsprozesses werden die an gesendeten Daten CloudWatch möglicherweise unterbrochen oder dupliziert. Ihre Metriken und Protokolldaten werden nach Abschluss der Migration wieder richtig in CloudWatch gespeichert.

Wir empfehlen, die Migration auf einer begrenzten Anzahl von Knoten zu testen, bevor eine gesamte Flotte auf den CloudWatch Agenten migriert wird. Nach der Migration können Sie die Protokollerfassung wieder mit SSM Agent ausführen, wenn Sie dies bevorzugen.

> **Important**
>
> In den folgenden Fällen können Sie mit den in diesem Thema beschriebenen Schritten nicht zum CloudWatch Agenten migrieren:

- Die bestehende Konfiguration für SSM Agent gibt mehrere Regionen an.
- Die bestehende Konfiguration für SSM Agent spezifiziert mehrere Gruppen von access/secret Schlüsselanmeldedaten.

In diesen Fällen ist es erforderlich, die Protokollerfassung zu deaktivieren SSM Agent und den CloudWatch Agenten ohne Migrationsprozess zu installieren. Weitere Informationen finden Sie in den folgenden Themen im CloudWatch Amazon-Benutzerhandbuch:

- Den CloudWatch Agenten installieren
- Installation des CloudWatch Agenten auf lokalen Servern

Bevor Sie beginnen

Bevor Sie mit der Migration zum CloudWatch Agenten für die Protokollerfassung beginnen, stellen Sie sicher, dass die Knoten, auf denen Sie die Migration durchführen werden, die folgenden Anforderungen erfüllen:

- Das Betriebssystem ist eine 64-Bit-Version von Windows Server.
- SSM Agent 2.2.93.0 oder höher ist auf dem Knoten installiert.
- SSM Agent ist für die Überwachung auf dem Knoten konfiguriert.

Topics

- Automatische Migration zum Agenten CloudWatch
- CloudWatch Manuelles Migrieren zum Agenten

Automatische Migration zum Agenten CloudWatch

Windows ServerNur für EC2 Instanzen können Sie die AWS Systems Manager Konsole oder die AWS Command Line Interface (AWS CLI) verwenden, um automatisch zum CloudWatch Agenten als Tool zur Protokollerfassung zu migrieren.

> **Note**
>
> AWS Systems Manager unterstützt die Migration vom Unified CloudWatch Agent SSM Agent zum Sammeln von Protokollen und Messdaten nur in 64-Bit-Versionen von Windows.

Informationen zur Einrichtung des Unified CloudWatch Agents auf anderen Betriebssystemen und vollständige Informationen zur Verwendung des CloudWatch Agenten finden Sie unter Erfassung von Metriken und Protokollen von EC2 Amazon-Instances und lokalen Servern mit dem CloudWatch Agenten im CloudWatch Amazon-Benutzerhandbuch.
Sie können den CloudWatch Agenten auf anderen unterstützten Betriebssystemen verwenden, aber Sie können Systems Manager nicht verwenden, um eine Tool-Migration durchzuführen.

Überprüfen Sie nach erfolgreicher Migration Ihre Ergebnisse, CloudWatch um sicherzustellen, dass Sie die erwarteten Metriken, Protokolle oder Windows-Ereignisprotokolle erhalten. Wenn Sie mit den Ergebnissen zufrieden sind, können Sie optional folgende Aktion durchführen: Speichern Sie die CloudWatch Agentenkonfigurationseinstellungen in Parameter Store. Wenn die Migration nicht erfolgreich verlaufen ist oder die Ergebnisse nicht den Erwartungen entsprechen, können Sie Rollback zur Protokollerfassung mit SSM Agent ausprobieren.

Note

Wenn Sie eine Quellkonfigurationsdatei migrieren möchten, die einen `{hostname}`-Eintrag enthält, sollten Sie daran denken, dass der `{hostname}`-Eintrag den Wert des Felds ändern kann, falls die Migration abgeschlossen ist. Nehmen wir beispielsweise an, dass der folgende `"LogStream": "{hostname}"` Eintrag einem Server mit dem Namen MyLogServer001 zugeordnet ist.

```
{
"Id": "CloudWatchIISLogs",
"FullName":
 "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
"Parameters": {
"AccessKey": "",
"SecretKey": "",
"Region": "us-east-1",
"LogGroup": "Production-Windows-IIS",
"LogStream": "{hostname}"
     }
}
```

Nach der Migration wird dieser Eintrag einer Domäne wie ip-11-1-1-11.production zugeordnet. ExampleCompany.com. Um den lokalen hostname-Wert beizubehalten, geben Sie `{local_hostname}` anstelle von `{hostname}` an.

Um automatisch zum CloudWatch Agenten (Konsole) zu migrieren

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command und anschließend Befehl ausführen aus.
3. Wählen Sie in der Liste Befehlsdokument die Option `AmazonCloudWatch-MigrateCloudWatchAgent` aus.
4. Wählen Sie für Status die Option Enabled.
5. Identifizieren Sie für den Abschnitt Ziele die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

6. Für Ratenregelung:

   - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten

Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

7. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, sind diejenigen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle (hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, nicht die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

8. Aktivieren Sie das Kontrollkästchen SNS-Benachrichtigungen aktivieren im Abschnitt SNS-Benachrichtigungen, wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

   Weitere Informationen zum Konfigurieren von Amazon-SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

9. Klicken Sie auf Ausführen.

Um automatisch zum Agenten zu migrieren () CloudWatch AWS CLI

- Führen Sie den folgenden Befehl aus.

```
aws ssm send-command --document-name AmazonCloudWatch-MigrateCloudWatchAgent --
targets Key=instanceids,Values=ID1,ID2,ID3
```

ID1ID2, und *ID3* stellen die Knoten dar, IDs die Sie aktualisieren möchten, z. B. i-02573cafcfExample.

CloudWatch Manuelles Migrieren zum Agenten

Gehen Sie für lokale Windows Server Knoten oder EC2 Instances für wie folgt vorWindows Server, um die Protokollerfassung manuell auf den CloudWatch Amazon-Agenten zu migrieren.

> **Note**
>
> Wenn Sie eine Quellkonfigurationsdatei migrieren möchten, die einen `{hostname}`-Eintrag enthält, sollten Sie daran denken, dass der `{hostname}`-Eintrag den Wert des Felds ändern kann, falls die Migration abgeschlossen ist. Nehmen wir zum Beispiel an, dass der folgende `"LogStream": "{hostname}"` Eintrag einem Server mit dem Namen MyLogServer001 zugeordnet ist.
>
> ```
> {
> "Id": "CloudWatchIISLogs",
> "FullName":
>  "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
> "Parameters": {
> "AccessKey": "",
> "SecretKey": "",
> "Region": "us-east-1",
> "LogGroup": "Production-Windows-IIS",
> "LogStream": "{hostname}"
>     }
> }
> ```
>
> Nach der Migration wird dieser Eintrag einer Domäne wie ip-11-1-1-11.production zugeordnet. ExampleCompany.com. Um den lokalen hostname-Wert beizubehalten, geben Sie `{local_hostname}` anstelle von `{hostname}` an.

Erstens: Um den CloudWatch Agenten (Konsole) zu installieren

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Run Command und anschließend Befehl ausführen aus.
3. Wählen Sie in der Liste Befehlsdokument die Option `AWS-ConfigureAWSPackage` aus.
4. Für Action (Aktion), wählen Sie `Install` aus.
5. Geben Sie unter Name **`AmazonCloudWatchAgent`** ein.
6. Geben Sie unter Version **`latest`** ein, wenn dies nicht bereits standardmäßig bereitgestellt wird.
7. Identifizieren Sie für den Abschnitt Ziele die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

8. Für Ratenregelung:

   - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, sind diejenigen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle (hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, nicht die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

10. Aktivieren Sie das Kontrollkästchen SNS-Benachrichtigungen aktivieren im Abschnitt SNS-Benachrichtigungen, wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon-SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

11. Klicken Sie auf Ausführen.

2) Aktualisieren des JSON-Formats der Konfigurationsdaten

- Um die JSON-Formatierung der vorhandenen Konfigurationseinstellungen für den CloudWatch Agenten zu aktualisieren Run Command, verwenden Sie ein Tool oder melden Sie sich direkt mit einer RDP-Verbindung beim Knoten an, um die folgenden PowerShell Windows-Befehle nacheinander auf dem Knoten auszuführen. AWS Systems Manager

```
cd ${Env:ProgramFiles}\\Amazon\\AmazonCloudWatchAgent
```

```
.\\amazon-cloudwatch-agent-config-wizard.exe --isNonInteractiveWindowsMigration
```

  *{Env:ProgramFiles}*steht normalerweise für den Speicherort, an dem sich das Amazon-Verzeichnis befindet, das den CloudWatch Agenten enthält`C:\Program Files`.

Drei: Um den CloudWatch Agenten (Konsole) zu konfigurieren und zu starten

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command und anschließend Befehl ausführen aus.
3. Wählen Sie in der Liste Befehlsdokument die Option `AWS-RunPowerShellScript` aus.
4. Geben Sie unter Commands (Befehle) die beiden folgenden Befehle ein.

```
cd ${Env:ProgramFiles}\Amazon\AmazonCloudWatchAgent
```

```
.\amazon-cloudwatch-agent-ctl.ps1 -a fetch-config -m ec2 -c file:config.json -s
```

*{Env:ProgramFiles}*steht normalerweise für den Speicherort, an dem sich das Amazon-Verzeichnis befindet, das den CloudWatch Agenten enthält`C:\Program Files`.

5. Identifizieren Sie für den Abschnitt Ziele die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

> **Tip**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

6. Für Ratenregelung:

   - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

> **Note**
>
> Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

7. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, sind diejenigen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle (hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, nicht die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

8. Aktivieren Sie das Kontrollkästchen SNS-Benachrichtigungen aktivieren im Abschnitt SNS-Benachrichtigungen, wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

   Weitere Informationen zum Konfigurieren von Amazon-SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

9. Klicken Sie auf Ausführen.

4) Deaktivieren der Protokollerfassung im SSM Agent (Konsole)

1. Öffnen Sie die Konsole unter. AWS Systems Manager https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Run Command und anschließend Befehl ausführen aus.

3. Wählen Sie in der Liste Befehlsdokument die Option `AWS-ConfigureCloudWatch` aus.
4. Wählen Sie unter Status die Option Disabled (Deaktiviert) aus.
5. Identifizieren Sie für den Abschnitt Ziele die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

6. Wählen Sie unter Status die Option `Disabled` aus.
7. Für Ratenregelung:

   - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

8. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, sind diejenigen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle (hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, nicht die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

9. Aktivieren Sie das Kontrollkästchen SNS-Benachrichtigungen aktivieren im Abschnitt SNS-Benachrichtigungen, wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

   Weitere Informationen zum Konfigurieren von Amazon-SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

10. Klicken Sie auf Ausführen.

    Nachdem Sie diese Schritte abgeschlossen haben, überprüfen Sie Ihre Logs, CloudWatch um sicherzustellen, dass Sie die erwarteten Metriken, Protokolle oder Windows-Ereignisprotokolle erhalten. Wenn die Ergebnisse zufriedenstellend sind, können Sie optional Speichern Sie die CloudWatch Agentenkonfigurationseinstellungen in Parameter Store. Wenn die Migration nicht erfolgreich verlaufen ist oder die Ergebnisse nicht den Erwartungen entsprechen, können Sie Rollback zur Protokollerfassung mit SSM Agent ausprobieren.

Speichern Sie die CloudWatch Agentenkonfigurationseinstellungen in Parameter Store

Sie können den Inhalt einer CloudWatch Agent-Konfigurationsdatei in speichernParameter Store. Wenn diese Konfigurationsdaten in einem Parameter angegeben werden, können mehrere Knoten ihre Konfigurations-Einstellungen daraus ableiten und Sie müssen keine Konfigurationsdateien auf Ihren Knoten erstellen oder manuell aktualisieren. Sie können beispielsweise den Inhalt des Parameters in Konfigurationsdateien auf mehreren Knoten schreiben oder ein Tool verwenden

State Manager AWS Systems Manager, um Konfigurationsabweichungen in den CloudWatch Agentenkonfigurationseinstellungen über eine ganze Flotte von Knoten hinweg zu vermeiden. Run Command

Wenn Sie den Assistenten zur CloudWatch Agentenkonfiguration ausführen, können Sie festlegen, dass der Assistent Ihre Konfigurationseinstellungen als neuen Parameter in speichertParameter Store. Informationen zur Ausführung des Assistenten für die CloudWatch Agentenkonfiguration finden Sie unter Erstellen der CloudWatch Agentenkonfigurationsdatei mit dem Assistenten im CloudWatch Amazon-Benutzerhandbuch.

Wenn Sie den Assistenten ausgeführt, aber nicht die Option zum Speichern der Einstellungen als Parameter ausgewählt haben, oder wenn Sie die CloudWatch Agenten-Konfigurationsdatei manuell erstellt haben, können Sie die Daten, die als Parameter auf Ihrem Knoten gespeichert werden sollen, in der folgenden Datei abrufen.

```
${Env:ProgramFiles}\Amazon\AmazonCloudWatchAgent\config.json
```

*{Env:ProgramFiles}*steht normalerweise für den Speicherort, an dem sich das Amazon-Verzeichnis befindet, das den CloudWatch Agenten enthält`C:\Program Files`.

Wir empfehlen, ein Backup des JSON-Formats in dieser Datei an einem anderen Speicherort als den Knoten selbst zu speichern.

Weitere Informationen zum Erstellen eines Parameters finden Sie unter Parameter Store-Parameter im Systems Manager erstellen.

Weitere Informationen über den CloudWatch Agenten finden Sie im CloudWatch Amazon-Benutzerhandbuch unter Erfassung von Metriken und Protokollen von EC2 Amazon-Instances und lokalen Servern mit dem CloudWatch Agenten.

Rollback zur Protokollerfassung mit SSM Agent

Wenn Sie wieder SSM Agent für die Protokollerfassung verwenden möchten, führen Sie die folgenden Schritte aus.

1) Abrufen von Konfigurationsdaten aus SSM Agent

1. Suchen Sie die Inhalte der SSM Agent-Config-Datei auf dem Knoten, auf dem Sie erneut Protokolle mit dem SSM Agent erfassen möchten. Diese JSON-Datei befindet sich in der Regel am folgenden Speicherort:

```
${Env:ProgramFiles}\\Amazon\\SSM\\Plugins\\awsCloudWatch\
\AWS.EC2.Windows.CloudWatch.json
```

*{Env:ProgramFiles}*steht in der Regel `C:\Program Files` für den Speicherort, an dem sich das `Amazon` Verzeichnis befindet.

2. Kopieren Sie diese Daten für die Verwendung in einem späteren Schritt in eine Textdatei.

   Wir empfehlen, ein Backup der JSON-Datei an einem anderen Speicherort als den Knoten selbst zu speichern.

Zweitens: Um den CloudWatch Agenten (Konsole) zu deinstallieren

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command und anschließend Befehl ausführen aus.
3. Wählen Sie in der Liste Befehlsdokument die Option `AWS-ConfigureAWSPackage` aus.
4. Wählen Sie für Action (Aktion) die Option Uninstall (Deinstallieren) aus.
5. Geben Sie unter Name **`AmazonCloudWatchAgent`** ein.
6. Identifizieren Sie für den Abschnitt Ziele die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Ratenregelung:

   - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

> **Note**
>
> Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

- Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

8. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, sind diejenigen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle (hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, nicht die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

9. Aktivieren Sie das Kontrollkästchen SNS-Benachrichtigungen aktivieren im Abschnitt SNS-Benachrichtigungen, wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

   Weitere Informationen zum Konfigurieren von Amazon-SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

10. Klicken Sie auf Ausführen.

Drei: Aktivieren der Protokollerfassung in SSM Agent(Konsole)

1. Öffnen Sie die Konsole unter. AWS Systems Manager https://console.aws.amazon.com/systems-manager/
2. Wählen Sie im Navigationsbereich Run Command und anschließend Befehl ausführen aus.
3. Wählen Sie in der Liste Befehlsdokument die Option `AWS-ConfigureCloudWatch` aus.
4. Wählen Sie unter Status die Option `Enabled` aus.
5. Fügen Sie unter Properties (Eigenschaften) den Inhalt der alten Konfigurationsdaten ein, die Sie in der Textdatei gespeichert haben.
6. Identifizieren Sie für den Abschnitt Ziele die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Ratenregelung:

   - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler

empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

8. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, sind diejenigen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle (hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, nicht die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

9. Aktivieren Sie das Kontrollkästchen SNS-Benachrichtigungen aktivieren im Abschnitt SNS-Benachrichtigungen, wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

   Weitere Informationen zum Konfigurieren von Amazon-SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

10. Klicken Sie auf Ausführen.

SSM AgentProtokolle an CloudWatch Logs senden

AWS Systems Manager Agent (SSM Agent) ist Amazon-Software, die auf Ihren EC2 Instances, Edge-Geräten, lokalen Servern und virtuellen Maschinen (VMs) ausgeführt wird, die für Systems Manager konfiguriert sind. SSM Agentverarbeitet Anfragen vom Systems Manager Manager-Dienst in der Cloud und konfiguriert Ihren Computer wie in der Anfrage angegeben. Mehr über SSM Agent erfahren Sie unter Arbeiten mit SSM Agent.

Darüber hinaus können Sie mithilfe der folgenden Schritte konfigurieren, SSM Agent dass Protokolldaten an Amazon CloudWatch Logs gesendet werden.

Bevor Sie beginnen

Erstellen Sie eine Protokollgruppe in CloudWatch Logs. Weitere Informationen finden Sie unter Erste Schritte mit CloudWatch Logs im Amazon CloudWatch Logs-Benutzerhandbuch.

So konfigurieren Sie SSM Agent das Senden von Protokollen an CloudWatch

1. Melden Sie sich bei einem Knoten an und suchen Sie die folgende Datei:

   Linux

   Bei den meisten Linux-Knotentypen: `/etc/amazon/ssm/seelog.xml.template`.

   Unter Ubuntu Server 20.04, 18.04 und 16.04 LTS: `/snap/amazon-ssm-agent/current/seelog.xml.template`

   macOS

   `/opt/aws/ssm/seelog.xml.template`

   Windows

   `%ProgramFiles%\Amazon\SSM\seelog.xml.template`
2. Ändern des Dateinamens von `seelog.xml.template` in `seelog.xml`

   > **Note**
   >
   > Unter Ubuntu Server 20.04, 18.04 und 16.04 LTS muss die Datei `seelog.xml` im Verzeichnis `/etc/amazon/ssm/` erstellt werden. Sie können dieses Verzeichnis und diese Datei mit den folgenden Befehlen erstellen.
   >
   > ```
   > sudo mkdir -p /etc/amazon/ssm
   > ```
   >
   > ```
   > sudo cp -pr /snap/amazon-ssm-agent/current/* /etc/amazon/ssm
   > ```
   >
   > ```
   > sudo cp -p /etc/amazon/ssm/seelog.xml.template /etc/amazon/ssm/seelog.xml
   > ```
3. Öffnen Sie die Datei `seelog.xml` in einem Texteditor und suchen Sie nach folgendem Abschnitt.

Linux and macOS

```
<outputs formatid="fmtinfo">
   <console formatid="fmtinfo"/>
   <rollingfile type="size" filename="/var/log/amazon/ssm/amazon-ssm-agent.log"
 maxsize="30000000" maxrolls="5"/>
   <filter levels="error,critical" formatid="fmterror">
      <rollingfile type="size" filename="/var/log/amazon/ssm/errors.log"
 maxsize="10000000" maxrolls="5"/>
   </filter>
</outputs>
```

Windows

```
<outputs formatid="fmtinfo">
   <console formatid="fmtinfo"/>
   <rollingfile type="size" maxrolls="5" maxsize="30000000"
 filename="{{LOCALAPPDATA}}\Amazon\SSM\Logs\amazon-ssm-agent.log"/>
   <filter formatid="fmterror" levels="error,critical">
      <rollingfile type="size" maxrolls="5" maxsize="10000000"
 filename="{{LOCALAPPDATA}}\Amazon\SSM\Logs\errors.log"/>
   </filter>
</outputs>
```

4. Bearbeiten Sie die Datei und fügen Sie nach dem schließenden </filter>-Tag ein benutzerdefiniertes Namenselement hinzu. Im folgenden Beispiel wird der benutzerdefinierte Name als `cloudwatch_receiver` angegeben.

   Linux and macOS

```
<outputs formatid="fmtinfo">
   <console formatid="fmtinfo"/>
   <rollingfile type="size" filename="/var/log/amazon/ssm/amazon-ssm-agent.log"
 maxsize="30000000" maxrolls="5"/>
   <filter levels="error,critical" formatid="fmterror">
      <rollingfile type="size" filename="/var/log/amazon/ssm/errors.log"
 maxsize="10000000" maxrolls="5"/>
   </filter>
   <custom name="cloudwatch_receiver" formatid="fmtdebug" data-log-group="your-
CloudWatch-log-group-name"/>
</outputs>
```

Windows

```
<outputs formatid="fmtinfo">
   <console formatid="fmtinfo"/>
   <rollingfile type="size" maxrolls="5" maxsize="30000000"
 filename="{{LOCALAPPDATA}}\Amazon\SSM\Logs\amazon-ssm-agent.log"/>
   <filter formatid="fmterror" levels="error,critical">
      <rollingfile type="size" maxrolls="5" maxsize="10000000"
 filename="{{LOCALAPPDATA}}\Amazon\SSM\Logs\errors.log"/>
   </filter>
   <custom name="cloudwatch_receiver" formatid="fmtdebug" data-log-group="your-
CloudWatch-log-group-name"/>
</outputs>
```

5. Speichern Sie Ihre Änderungen und starten Sie dann SSM Agent oder den Knoten neu.
6. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
7. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus und wählen Sie dann den Namen der Protokollgruppe aus.

> **Tip**
>
> Der Protokoll-Stream für SSM Agent-Protokolldatei-Daten wird nach Knoten-ID organisiert.

Überwachung der Ereignisse Ihrer Änderungsanfragen

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

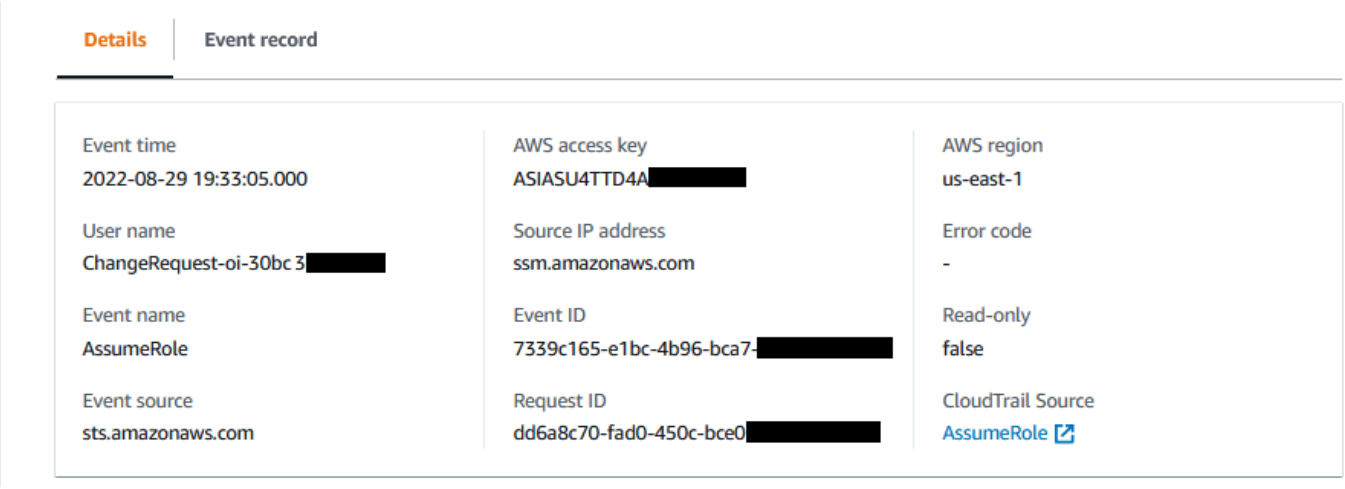
Nachdem Sie die Integration mit AWS CloudTrail Lake aktiviert und einen Ereignisdatenspeicher erstellt haben, können Sie überprüfbare Details zu den Änderungsanforderungen anzeigen, die in Ihrem Konto oder Ihrer Organisation ausgeführt werden. Dazu gehören Details wie die folgenden:

- Die Identität des Benutzers, der die Änderungsanfrage initiiert hat
- Der AWS-Regionen Ort, an dem die Änderungen vorgenommen wurden
- Die Quell-IP-Adresse für die Anfrage
- Der für die Anfrage verwendete AWS Zugriffsschlüssel
- Die Ausführung der API-Aktionen für die Änderungsanfrage
- Die für diese Aktionen enthaltenen Anfrageparameter
- Die während des Vorgangs aktualisierten Ressourcen

Im Folgenden finden Sie Beispiele für Ereignisdetails, die Sie für eine Änderungsanforderung anzeigen können, nachdem Sie den Ereignisdatenspeicher in AWS CloudTrail Lake erstellt haben.

Details

Das folgende Image zeigt die allgemeinen Informationen zu einer Änderungsanfrage, die auf der Registerkarte Details verfügbar sind. Zu diesen Details gehören Informationen wie der Zeitpunkt des Beginns der Änderungsanfrage, die ID des Benutzers, der die Änderungsanfrage initiiert hat, die betroffenen AWS-Region sowie die mit der Anfrage verknüpften Ereignis-ID und Anfrage-ID.



Event record

Die folgende Abbildung zeigt die Struktur des JSON-Inhalts, der von CloudTrail Lake für ein Änderungsanforderungsereignis bereitgestellt wird. Diese Daten werden in einem Änderungsauftrag auf der Registerkarte Event record (Ereignisdatensatz) bereitgestellt.

Important

Wenn Sie Change Manager für eine Organisation verwenden, können Sie das folgende Verfahren ausführen, während Sie entweder beim Verwaltungskonto oder beim delegierten Administratorkonto für Change Manager angemeldet sind.
Um jedoch das delegierte Administratorkonto für diese Schritte zu verwenden, muss dasselbe delegierte Administratorkonto sowohl für als auch CloudTrail angegeben werden. Change Manager
Wenn Sie sich beim Verwaltungskonto für anmeldenChange Manager, können Sie das delegierte Administratorkonto für CloudTrail auf der CloudTrail Seite Einstellungen hinzufügen oder ändern. Dies muss erfolgen, bevor das delegierte Administratorkonto einen Ereignisdatenspeicher zur Verwendung durch die gesamte Organisation erstellen kann.

Um die CloudTrail Lake-Ereignisverfolgung zu aktivieren, gehen Sie von Change Manager

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Change Manager aus.

3. Wählen Sie die Registerkarte Requests (Anforderungen).
4. Wählen Sie eine bereits vorhandene Anfrage aus und wählen Sie dann die Registerkarte Associated events (Zugeordnete Ereignisse).
5. Wählen Sie Enable CloudTrail Lake.
6. Folgen Sie den Schritten unter Erstellen eines Ereignisdatenspeichers für CloudTrail Ereignisse im AWS CloudTrail Benutzerhandbuch.

   Um sicherzustellen, dass die Ereignisdaten für Ihre Änderungsanfragen gespeichert werden, treffen Sie die folgenden Auswahlen, während Sie das Verfahren abschließen:

   - Behalten Sie für Ereignistyp die AWS Standardereignisse und CloudTrailEreignisse bei.
   - Wenn Sie Change Manager mit einer Organisation verwenden, wählen Sie die Option Für alle Konten in meiner Organisation aktivieren aus.
   - Deaktivieren Sie bei Verwaltungsereignissen das Kontrollkästchen Schreiben nicht.

   Andere Optionen, die Sie beim Erstellen Ihres Ereignisdatenspeichers auswählen, wirken sich nicht auf die Speicherung von Ereignisdaten für Ihre Änderungsanfragen aus.

Überwachung Ihrer Automatisierungen

Metriken sind das grundlegende Konzept in Amazon CloudWatch. Eine Metrik stellt eine zeitlich angeordnete Gruppe von Datenpunkten, die an CloudWatch veröffentlicht werden, dar. Eine Metrik können Sie sich als eine zu überwachende Variable und die Datenpunkte als die Werte dieser Variablen im Laufe der Zeit vorstellen.

Automation ist ein Tool in AWS Systems Manager. Systems Manager veröffentlicht Metriken zur Automation-Nutzung in CloudWatch. So können Sie Alarme basierend auf diesen Metriken festlegen.

So zeigen Sie Metriken in der CloudWatch-Konsole an

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie SSM aus.
4. Wählen Sie auf der Registerkarte Metrics (Metriken) erst Usage (Nutzung) und dann By AWS Resource (Nach Ressource) aus.
5. Geben Sie in das Suchfeld neben der Liste der Metriken SSM ein.

So zeigen Sie Automation-Metriken mit der AWS CLI an

Öffnen Sie einen Eingabe-Prompt und verwenden Sie den folgenden Befehl.

```
aws cloudwatch list-metrics \
    --namespace "AWS/Usage"
```

Automation-Metriken

Systems Manager sendet die folgenden Automation-Metriken an CloudWatch.

Metrik	Beschreibung
`ConcurrentAutomationUsage`	Die Anzahl der Automatisierungen, die gleichzeitig in AWS-Konto und AWS-Region ausgeführt werden können.
`QueuedAutomationUsage`	Die Anzahl der derzeit in der Warteschlange befindlichen Automatisierungen, die noch nicht gestartet wurden und den Status `Pending` aufweisen.

Weitere Informationen zum Arbeiten mit CloudWatch-Metriken finden Sie in den folgenden Themen im Amazon CloudWatch-Benutzerhandbuch:

- Metrics (Metriken)
- Verwenden von Amazon-CloudWatch-Metriken
- Verwenden von Amazon CloudWatch-Alarmen

Überwachen von Run Command-Metriken mit Amazon CloudWatch

Metriken sind das grundlegende Konzept in Amazon CloudWatch. Eine Metrik stellt eine zeitlich angeordnete Gruppe von Datenpunkten, die an CloudWatch veröffentlicht werden, dar. Sie können sich eine Metrik als eine zu überwachende Variable und die Datenpunkte als die Werte dieser Variablen im Laufe der Zeit vorstellen.

AWS Systems Manager veröffentlicht Metriken über den Status von Run Command-Befehlen an CloudWatch, wodurch Sie Alarme auf Basis dieser Metriken einstellen können. Run Command ist ein Tool in AWS Systems Manager. Diese Statistiken werden für einen längeren Zeitraum aufgezeichnet, damit Sie auf historische Informationen zugreifen können und eine bessere Übersicht über die Erfolgsrate der in Ihrem AWS-Kontoausgeführten Befehle erhalten.

Zu den Terminalstatuswerten für Befehle, für die Sie Metriken verfolgen können, gehören `Success`, `Failed` und `Delivery Timed Out`. Für ein SSM-Befehlsdokument, das stündlich ausgeführt werden soll, können Sie beispielsweise einen Alarm konfigurieren, der Sie benachrichtigt, wenn der Status `Success` für eine dieser Stunden nicht gemeldet wird. Weitere Informationen zu Befehlsstatuswerten finden Sie unter Grundlegendes zu Befehlsstatus.

Anzeigen von Metriken in der CloudWatch-Konsole

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie im Bereich Alarms by AWS service (Alarme nach Service) für Services (Services) SSM-Run Command aus.

So zeigen Sie Metriken mit der AWS CLI

Öffnen Sie eine Eingabeaufforderung und verwenden Sie den folgenden Befehl.

```
aws cloudwatch list-metrics --namespace "AWS/SSM-RunCommand"
```

Verwenden Sie den folgenden Befehl, um alle verfügbaren Metriken aufzulisten.

```
aws cloudwatch list-metrics
```

Systems Manager Run Command-Metriken und -Dimensionen

Systems Manager sendet einmal pro Minute Run Command-Befehlsmetriken an CloudWatch.

Systems Manager sendet die folgenden Befehlsmetriken an CloudWatch.

> **Note**
>
> Diese Metriken verwenden `Count` als Einheit, daher sind `Sum` und `SampleCount` die nützlichsten Statistiken.

Metrik	Beschreibung
`CommandsDeliveryTimedOut`	Die Anzahl der Befehle, die den Terminalstatus `Delivery Timed Out` haben.
`CommandsFailed`	Die Anzahl der Befehle, die den Terminalstatus `Failed` haben.
`CommandsSucceeded`	Die Anzahl der Befehle, die den Terminalstatus `Success` haben.

Weitere Informationen zum Arbeiten mit CloudWatch-Metriken finden Sie in den folgenden Themen im Amazon CloudWatch-Benutzerhandbuch:

- Metrics (Metriken)
- Verwenden von Amazon-CloudWatch-Metriken
- Verwenden von Amazon CloudWatch-Alarmen

Protokollierung von AWS Systems Manager-API-Aufrufen mit AWS CloudTrail

AWS Systems Manager ist in AWS CloudTrail integriert, einem Service, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS-Service ausgeführt werden. CloudTrail erfasst alle API-Aufrufe für Systems Manager als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der Systems Manager-Konsole und Code-Aufrufe der Systems Manager-API-Operationen. Mit den von CloudTrail erfassten Informationen können Sie die an Systems Manager gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und zusätzliche Details bestimmen.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Anmeldeinformationen des Root-Benutzers oder des Benutzers gestellt wurde.
- Die Anforderung wurde im Namen eines IAM-Identity-Center-Benutzers erstellt.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anforderung aus einem anderen AWS-Service gesendet wurde.

Bei Kontoerstellung ist CloudTrail standardmäßig in AWS-Konto aktiviert und Sie haben automatisch Zugriff auf den CloudTrail-Ereignisverlauf. Der CloudTrail-Ereignisverlauf stellt eine anzeigbare, durchsuchbare, herunterladbare und unveränderliche Aufzeichnung der Verwaltungsereignisse der letzten 90 Tage in einer AWS-Region bereit. Weitere Informationen finden Sie unter Arbeiten mit dem CloudTrail-Ereignisverlauf im AWS CloudTrail-Benutzerhandbuch. Für die Anzeige des Eventverlaufs fallen keine CloudTrail-Gebühren an.

Erstellen Sie einen Trail oder einen Ereignisdatenspeicher in CloudTrail Lake, um Aktivitäten und Ereignisse in Ihrer AWS-Konto über 90 Tage hinaus fortlaufend aufzuzeichnen.

CloudTrail-Trails

Ein Trail ermöglicht es CloudTrail, Protokolldateien in einem Amazon-S3-Bucket bereitzustellen. Trails, die mit der AWS-Managementkonsole erstellt wurden, sind multiregional. Sie können mithilfe von AWS CLI einen Einzel-Region- oder einen Multi-Region-Trail erstellen. Das Erstellen eines Multi-Region-Trails wird empfohlen, da Sie so die Aktivitäten in allen AWS-Regionen in Ihrem Konto erfassen. Wenn Sie einen Einzel-Region-Trail erstellen, können Sie nur die Ereignisse anzeigen, die im AWS-Region des Trails protokolliert wurden. Weitere Informationen zu Trails finden Sie unter Erstellen eines Trails für Ihr AWS-Konto und Erstellen eines Trails für eine Organisation im AWS CloudTrail-Benutzerhandbuch.

Sie können eine Kopie Ihrer laufenden Verwaltungsereignisse kostenlos von CloudTrail an Ihren Amazon-S3-Bucket senden, indem Sie einen Trail erstellen. Es fallen jedoch Speichergebühren für Amazon S3 an. Weitere Informationen zu CloudTrail-Preisen finden Sie unter AWS CloudTrail – Preise. Informationen zu Amazon-S3-Preisen finden Sie unter Amazon S3 – Preise.

CloudTrail-Lake-Ereignisdatenspeicher

Mit CloudTrail Lake können Sie SQL-basierte Abfragen zu Ihren Ereignissen ausführen. CloudTrail Lake konvertiert vorhandene Ereignisse im zeilenbasierten JSON-Format in das

Apache ORC-Format. ORC ist ein spaltenförmiges Speicherformat, das für den schnellen Abruf von Daten optimiert ist. Die Ereignisse werden in Ereignisdatenspeichern zusammengefasst, bei denen es sich um unveränderliche Sammlungen von Ereignissen handelt, die auf Kriterien basieren, die Sie mit Hilfe von erweiterten Ereignisselektoren auswählen. Die Selektoren, die Sie auf einen Ereignisdatenspeicher anwenden, legen fest, welche Ereignisse beibehalten werden und für die Abfrage verfügbar sind. Weitere Informationen zu CloudTrail Lake finden Sie unter Arbeiten mit AWS CloudTrail-Lake im AWS CloudTrail-Benutzerhandbuch.

Für CloudTrail-Lake-Ereignisdatenspeicher und -abfragen fallen Gebühren an. Sie können die Preisoption auswählen, die Sie für einen Ereignisdatenspeicher verwenden möchten, während Sie ihn erstellen. Die Preisoption bestimmt die Kosten für die Erfassung und Speicherung von Ereignissen sowie die standardmäßige und maximale Aufbewahrungsdauer für den Ereignisdatenspeicher. Weitere Informationen zu CloudTrail-Preisen finden Sie unter AWS CloudTrail – Preise.

Systems-Manager-Datenereignisse in CloudTrail

Datenereignisse liefern Informationen zu den Ressourcenvorgängen, die an oder in einer Ressource ausgeführt werden (beispielsweise das Erstellen oder Öffnen eines Steuerkanals). Sie werden auch als Operationen auf Datenebene bezeichnet. Datenereignisse sind oft Aktivitäten mit hohem Volume. Standardmäßig protokolliert CloudTrail keine Datenereignisse. Der CloudTrail-Ereignisverlauf zeichnet keine Datenereignisse auf.

Für Datenereignisse werden zusätzliche Gebühren fällig. Weitere Informationen zu CloudTrail-Preisen finden Sie unter AWS CloudTrail – Preise.

Sie können Datenereignisse für die Systems-Manager-Ressourcentypen mithilfe der CloudTrail-Konsole, AWS CLI, oder CloudTrail-API-Vorgänge protokollieren. Weitere Informationen zum Protokollieren von Datenereignissen finden Sie unter Protokollieren von Datenereignissen mit dem AWS-Managementkonsole und Protokollieren von Datenereignissen mit dem AWS Command Line Interface im AWS CloudTrail-Benutzerhandbuch.

In der folgenden Tabelle sind die Systems-Manager-Ressourcentypen aufgeführt, für die Sie Datenereignisse protokollieren können. In der Spalte Datenereignistyp (Konsole) wird der Wert angezeigt, der in der CloudTrail-Konsole aus der Liste Datenereignistyp ausgewählt werden kann. In der Wertspalte resources.type wird der `resources.type` Wert angezeigt, den Sie angeben würden, wenn Sie erweiterte Event-Selektoren mithilfe der APIs AWS CLI oder CloudTrail konfigurieren

würden. In der Spalte Daten-APIs, die in CloudTrail protokolliert wurden, werden die API-Aufrufe angezeigt, die für den Ressourcentyp in CloudTrail protokolliert wurden.

Typ des Datenereignisses (Konsole)	resources.type-Wert	Bei CloudTrail protokollierte Daten-APIs
Systems Manager	`AWS::SSMMessages::ControlChannel`	• `CreateControlChannel` • `OpenControlChannel` Weitere Informationen zu diesen Vorgängen finden Sie unter Vom Amazon Message Gateway Service definierte Aktionen in der Service-Authorisierungs-Referenz.
Von Systems Manager verwalteter Knoten	`AWS::SSM::ManagedNode`	• `RequestManagedInstanceRoleToken` – Dieses Ereignis wird generiert, wenn der AWS Systems Manager-Agent (SSM Agent), der auf einem von Systems Manager verwalteten Knoten ausgeführt wird, Anmeldeinformationen vom Systems-Manager-Anmeldeinformationsservice anfordert. Weitere Informationen über den Vorgang `RequestManagedInstanceRoleToken` finden Sie im Abschnitt Validierung von hybrid-ak

Typ des Datenereignisses (Konsole)	resources.type-Wert	Bei CloudTrail protokollierte Daten-APIs
		tivierten Maschinen mit einem Hardware-Fingerabdruck

Sie können erweiterte Event-Selektoren so konfigurieren, dass sie nach den Feldern `eventName`, `readOnly` und `resources.ARN` filtern, sodass nur die Ereignisse protokolliert werden, die für Sie wichtig sind. Weitere Informationen zu diesen Kontingenten finden Sie unter AdvancedFieldSelector in derAWS CloudTrail-API-Referenz.

Systems-Manager-Verwaltungsereignisse in CloudTrail

Verwaltungsereignisse liefern Informationen zu Verwaltungsvorgängen, die für Ressourcen im AWS-Konto ausgeführt wurden. Sie werden auch als Operationen auf Steuerebene bezeichnet. CloudTrail protokolliert standardmäßig Verwaltungsereignisse.

Systems Manager protokolliert alle Vorgänge der Steuerebene als Verwaltungsereignisse in CloudTrail. Systems-Manager-API-Vorgänge sind in der AWS Systems Manager-API-Referenz dokumentiert. Zum Beispiel generieren Aufrufe der Aktionen `CreateMaintenanceWindows`, `PutInventory`, `SendCommand` und `StartSession` Einträge in den CloudTrail-Protokolldateien. (Ein Beispiel für die Einrichtung von CloudTrail zur Überwachung eines Systems-Manager-API-Aufrufs finden Sie unter Überwachen der Sitzungsaktivität mithilfe von Amazon EventBridge (Konsole).

Systems Manager Beispiele für Ereignisse

Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit des API-Vorgangs, Anforderungsparameter usw. CloudTrail-Protokolleinträge sind kein geordnetes Stack-Trace der öffentlichen API-Aufrufe und Ereignisse erscheinen daher in keiner bestimmten Reihenfolge.

Beispiele:

- Beispiele für Verwaltungsereignisse
- Beispiele für Datenereignisse

Beispiele für Verwaltungsereignisse

Beispiel 1: **`DeleteDocument`**

Das folgende Beispiel zeigt ein CloudTrail-Ereignis, das den `DeleteDocument`-Vorgang zum Löschen von Dokumenten auf ein Dokument mit dem Namen `example-Document` in der Region USA Ost (Ohio) (us-east-2) demonstriert.


```
{
    "eventVersion": "1.04",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE:203.0.113.11",
        "arn": "arn:aws:sts::123456789012:assumed-role/example-role/203.0.113.11",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-03-06T20:19:16Z"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/example-role",
                "accountId": "123456789012",
                "userName": "example-role"
            }
        }
    },
    "eventTime": "2018-03-06T20:30:12Z",
    "eventSource": "ssm.amazonaws.com",
    "eventName": "DeleteDocument",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "203.0.113.11",
    "userAgent": "example-user-agent-string",
    "requestParameters": {
        "name": "example-Document"
    },
    "responseElements": null,
    "requestID": "86168559-75e9-11e4-8cf8-75d18EXAMPLE",
    "eventID": "832b82d5-d474-44e8-a51d-093ccEXAMPLE",
    "resources": [
```

```
        {
            "ARN": "arn:aws:ssm:us-east-2:123456789012:document/example-Document",
            "accountId": "123456789012"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

Beispiel 2: **`StartConnection`**

Das folgende Beispiel zeigt das CloudTrail-Ereignis für den Benutzer, das eine RDP-Verbindung mit Fleet Manager in der Region USA Ost (Ohio) (us-east-2) startet. Die zugrunde liegende API-Aktion ist `StartConnection`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::123456789012:assumed-role/exampleRole",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:sts::123456789012:assumed-role/exampleRole",
                "accountId": "123456789012",
                "userName": "exampleRole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2021-12-13T14:57:05Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2021-12-13T16:50:41Z",
    "eventSource": "ssm-guiconnect.amazonaws.com",
    "eventName": "StartConnection",
    "awsRegion": "us-east-2",
```

```
    "sourceIPAddress": "34.230.45.60",
    "userAgent": "example-user-agent-string",
    "requestParameters": {
        "AuthType": "Credentials",
        "Protocol": "RDP",
        "ConnectionType": "SessionManager",
        "InstanceId": "i-02573cafcfEXAMPLE"
    },
    "responseElements": {
        "ConnectionArn": "arn:aws:ssm-guiconnect:us-east-2:123456789012:connection/
fcb810cd-241f-4aae-9ee4-02d59EXAMPLE",
        "ConnectionKey": "71f9629f-0f9a-4b35-92f2-2d253EXAMPLE",
        "ClientToken": "49af0f92-d637-4d47-9c54-ea51aEXAMPLE",
        "requestId": "d466710f-2adf-4e87-9464-055b2EXAMPLE"
    },
    "requestID": "d466710f-2adf-4e87-9464-055b2EXAMPLE",
    "eventID": "fc514f57-ba19-4e8b-9079-c2913EXAMPLE",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

Beispiele für Datenereignisse

Beispiel 1: **`CreateControlChannel`**

Das folgende Beispiel zeigt ein CloudTrail-Ereignis, das die `CreateControlChannel`-Operation demonstriert.

```
{
  "eventVersion":"1.08",
  "userIdentity":{
    "type":"AssumedRole",
    "principalId":"AKIAI44QH8DHBEXAMPLE",
    "arn":"arn:aws:sts::123456789012:assumed-role/exampleRole",
    "accountId":"123456789012",
    "accessKeyId":"AKIAI44QH8DHBEXAMPLE",
    "sessionContext":{
      "sessionIssuer":{
        "type":"Role",
        "principalId":"AKIAI44QH8DHBEXAMPLE",
```

```
        "arn":"arn:aws:iam::123456789012:role/exampleRole",
        "accountId":"123456789012",
        "userName":"exampleRole"
      },
      "attributes":{
        "creationDate":"2023-05-04T23:14:50Z",
        "mfaAuthenticated":"false"
      }
    }
  },
  "eventTime":"2023-05-04T23:53:55Z",
  "eventSource":"ssm.amazonaws.com",
  "eventName":"CreateControlChannel",
  "awsRegion":"us-east-1",
  "sourceIPAddress":"192.0.2.0",
  "userAgent":"example-agent",
  "requestParameters":{
    "channelId":"44295c1f-49d2-48b6-b218-96823EXAMPLE",
    "messageSchemaVersion":"1.0",
    "requestId":"54993150-0e8f-4142-aa54-3438EXAMPLE",
    "userAgent":"example-agent"
  },
  "responseElements":{
    "messageSchemaVersion":"1.0",
    "tokenValue":"Value hidden due to security reasons.",
    "url":"example-url"
  },
  "requestID":"54993150-0e8f-4142-aa54-3438EXAMPLE",
  "eventID":"a48a28de-7996-4ca1-a3a0-a51fEXAMPLE",
  "readOnly":false,
  "resources":[
    {
      "accountId":"123456789012",
      "type":"AWS::SSMMessages::ControlChannel",
      "ARN":"arn:aws:ssmmessages:us-east-1:123456789012:control-
channel/44295c1f-49d2-48b6-b218-96823EXAMPLE"
    }
  ],
  "eventType":"AwsApiCall",
  "managementEvent":false,
  "recipientAccountId":"123456789012",
  "eventCategory":"Data"
}
```

Beispiel 2: **RequestManagedInstanceRoleToken**

Das folgende Beispiel zeigt ein CloudTrail-Ereignis, das die `RequestManagedInstanceRoleToken`-Operation demonstriert.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "123456789012:aws:ec2-instance:i-02854e4bEXAMPLE",
        "arn": "arn:aws:sts::123456789012:assumed-role/aws:ec2-instance/
i-02854e4bEXAMPLE",
        "accountId": "123456789012",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "123456789012:aws:ec2-instance",
                "arn": "arn:aws:iam::123456789012:role/aws:ec2-instance",
                "accountId": "123456789012",
                "userName": "aws:ec2-instance"
            },
            "attributes": {
                "creationDate": "2023-08-27T03:34:46Z",
                "mfaAuthenticated": "false"
            },
            "ec2RoleDelivery": "2.0"
        }
    },
    "eventTime": "2023-08-27T03:37:15Z",
    "eventSource": "ssm.amazonaws.com",
    "eventName": "RequestManagedInstanceRoleToken",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Apache-HttpClient/UNAVAILABLE (Java/1.8.0_362)",
    "requestParameters": {
        "fingerprint": "i-02854e4bf85EXAMPLE"
    },
    "responseElements": null,
    "requestID": "2582cced-455b-4189-9b82-7b48EXAMPLE",
    "eventID": "7f200508-e547-4c27-982d-4da0EXAMLE",
    "readOnly": true,
    "resources": [
        {
```

```
            "accountId": "123456789012",
            "type": "AWS::SSM::ManagedNode",
            "ARN": "arn:aws:ec2:us-east-1:123456789012:instance/i-02854e4bEXAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data"
}
```

Informationen zu den Inhalten von CloudTrail-Datensätzen finden Sie unter CloudTrail-Datensatzinhalte im AWS CloudTrail-Benutzerhandbuch.

Protokollierung der Automation-Aktionsausgabe mit CloudWatch Logs

Automation, ein Tool in AWS Systems Manager, ist in Amazon CloudWatch Logs integriert. Sie können die Ausgabe von `aws:executeScript`-Aktionen in Ihren Runbooks an die von Ihnen angegebene Protokollgruppe senden. Systems Manager erstellt keine Protokollgruppe oder Protokoll-Streams für Dokumente, die `aws:executeScript`-Aktionen nicht verwenden. Wenn das Dokument `aws:executeScript` verwendet, bezieht sich die an CloudWatch Logs gesendete Ausgabe nur auf diese Aktionen. Sie können die `aws:executeScript`-Aktionsausgabe, die in der Protokollgruppe „CloudWatch Logs“ für Debug- und Fehlerbehebungszwecke gespeichert ist, verwenden. Wenn Sie eine Protokollgruppe auswählen, die verschlüsselt ist, wird die `aws:executeScript`-Aktionsausgabe ebenfalls verschlüsselt. Protokollierungsausgabe von `aws:executeScript`-Aktionen ist eine Einstellung auf Kontoebene.

Um Aktionsausgaben an CloudWatch Logs für Amazon-eigene Runbooks zu senden, muss der Benutzer oder die Rolle, der bzw. die die Automatisierung ausführt, über Berechtigungen für die folgenden Vorgänge verfügen:

- `logs:CreateLogGroup`
- `logs:CreateLogStream`
- `logs:DescribeLogGroups`
- `logs:DescribeLogStreams`
- `logs:PutLogEvents`

Für Runbooks, die Sie besitzen, müssen der IAM-Servicerolle (oder AssumeRole), die Sie zum Ausführen des Runbooks verwenden, dieselben Berechtigungen hinzugefügt werden.

Senden einer Aktionsausgabe an CloudWatch Logs (Konsole)

1. Öffnen Sie die AWS Systems Manager-Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Klicken Sie im Navigationsbereich auf Automation.
3. Wählen Sie die Registerkarte Präferenzen und anschließend Bearbeiten aus.
4. Aktivieren Sie das Kontrollkästchen neben Send output to CloudWatch Logs (Senden von Ausgaben an CloudWatch Logs).
5. (Empfohlen) Aktivieren Sie das Kontrollkästchen neben Encrypt log data (Verschlüsseln von Protokolldaten). Wenn diese Funktion aktiviert ist, werden die Protokolldaten mithilfe des serverseitigen Verschlüsselungsschlüssels, der für die Protokollgruppe angegeben wurde, verschlüsselt. Wenn Sie die an CloudWatch Logs gesendeten Protokolldaten nicht verschlüsseln möchten, deaktivieren Sie das Kontrollkästchen. Deaktivieren Sie das Kontrollkästchen, wenn die Verschlüsselung für die Protokollgruppe nicht zulässig ist.
6. Wählen Sie für CloudWatch Logs log group (CloudWatch Logs-Protokollgruppe) einen der folgenden Schritte aus, um die vorhandene CloudWatch Logs-Protokollgruppe in Ihrem AWS-Konto anzugeben, an die Sie Aktionsausgaben senden wollen:

   - Send output to the default log group (Ausgabe an die Standardprotokollgruppe senden) - Wenn die Standard-Protokollgruppe nicht vorhanden ist (`/aws/ssm/automation/executeScript`), erstellt Automation diese für Sie.
   - Choose from a list of log groups (Aus einer Liste von Protokollgruppen auswählen): Wählen Sie eine Protokollgruppe, die bereits in Ihrem Konto erstellt wurde, um die Aktionsausgabe zu speichern.
   - Enter a log group name (Eingabe eines Protokollgruppennamens): Geben Sie in das Textfeld den Namen einer Protokollgruppe ein, die bereits in Ihrem Konto angelegt wurde, um die Aktionsausgabe zu speichern.
7. Wählen Sie Speichern.

Senden einer Aktionsausgabe an CloudWatch Logs (Befehlszeile)

1. Öffnen Sie das bevorzugte Befehlszeilen-Tool und führen Sie den folgenden Befehl aus, um das Aktionsausgabeziel zu aktualisieren.

Linux & macOS

```
aws ssm update-service-setting \
    --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
customer-script-log-destination \
    --setting-value CloudWatch
```

Windows

```
aws ssm update-service-setting ^
    --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
customer-script-log-destination ^
    --setting-value CloudWatch
```

PowerShell

```
Update-SSMServiceSetting `
    -SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
customer-script-log-destination" `
    -SettingValue "CloudWatch"
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

2. Führen Sie den folgenden Befehl aus, um die Protokollgruppe anzugeben, an die die Aktionsausgabe gesendet werden soll.

Linux & macOS

```
aws ssm update-service-setting \
    --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
customer-script-log-group-name \
    --setting-value my-log-group
```

Windows

```
aws ssm update-service-setting ^
    --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
customer-script-log-group-name ^
    --setting-value my-log-group
```

PowerShell

```
Update-SSMServiceSetting `
    -SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
customer-script-log-group-name" `
    -SettingValue "my-log-group"
```

Wenn der Befehl erfolgreich ausgeführt wurde, gibt es keine Ausgabe.

3. Führen Sie den folgenden Befehl aus, um den aktuellen Durchsatz für Serviceeinstellungen für Einstellungen für die Aktionsprotokollierung für Automation im aktuellen AWS-Konto und in der AWS-Region anzuzeigen.

   Linux & macOS

   ```
   aws ssm get-service-setting \
       --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
   customer-script-log-destination
   ```

   Windows

   ```
   aws ssm get-service-setting ^
       --setting-id arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
   customer-script-log-destination
   ```

   PowerShell

   ```
   Get-SSMServiceSetting `
       -SettingId "arn:aws:ssm:region:account-id:servicesetting/ssm/automation/
   customer-script-log-destination"
   ```

   Der Befehl gibt Informationen wie die folgenden zurück.

   ```
   {
       "ServiceSetting": {
           "Status": "Customized",
           "LastModifiedDate": 1613758617.036,
           "SettingId": "/ssm/automation/customer-script-log-destination",
   ```

```
        "LastModifiedUser": "arn:aws:sts::123456789012:assumed-role/Administrator/
User_1",
        "SettingValue": "CloudWatch",
        "ARN": "arn:aws:ssm:us-east-2:123456789012:servicesetting/ssm/automation/
customer-script-log-destination"
    }
}
```

Konfigurieren von Amazon CloudWatch Logs für Run Command

Wenn Sie einen Befehl mit Run Command, einem Tool in AWS Systems Manager, senden, können Sie angeben, wohin Sie die Befehlsausgabe senden möchten. Standardmäßig gibt Systems Manager nur die ersten 24 000 Zeichen der Befehlsausgabe zurück. Wenn Sie alle Details der Befehlsausgabe anzeigen möchten, können Sie einen Amazon Simple Storage Service (Amazon S3)-Bucket angeben. Alternativ können Sie Amazon CloudWatch Logs angeben. Wenn Sie CloudWatch Logs angeben, sendet Run Command in regelmäßigen Abständen alle Befehlsausgabe- und Fehlerprotokolle an CloudWatch Logs. Sie können Ausgabeprotokolle nahezu in Echtzeit überwachen, nach bestimmten Ausdrücken, Werten oder Mustern suchen und Alarme basierend auf der Suche erstellen.

Wenn Sie Ihren verwalteten Knoten so konfiguriert haben, dass er die AWS Identity and Access Management (IAM)-verwalteten Richtlinien `AmazonSSMManagedInstanceCore` und `CloudWatchAgentServerPolicy` verwendet, dann benötigt Ihr Knoten keine zusätzliche Konfiguration, um Ausgaben an CloudWatch Logs zu senden. Wählen Sie diese Option, wenn Sie Befehle von der Konsole senden, oder den Abschnitt `cloud-watch-output-config` und den Parameter `CloudWatchOutputEnabled` hinzufügen, wenn Sie die AWS Command Line Interface (AWS CLI), AWS Tools for Windows PowerShell, oder einen API-Vorgang verwenden. Der `cloud-watch-output-config`-Abschnitt und der `CloudWatchOutputEnabled`-Parameter sind später in diesem Thema noch ausführlicher beschrieben.

Informationen zum Hinzufügen von Richtlinien zu einem Instance-Profil für EC2-Instances finden Sie unter Konfigurieren von erforderlichen Instance-Berechtigungen für Systems Manager. Informationen zum Hinzufügen von Richtlinien zu einer Servicerolle für On-Premises-Server und virtuelle Maschinen, die Sie als verwaltete Knoten verwenden möchten, finden Sie unter Erstellen der für Systems Manager in Hybrid- und Multi-Cloud-Umgebungen erforderlichen IAM-Servicerolle.

Wenn Sie eine benutzerdefinierte Richtlinie auf Ihren Knoten verwenden, aktualisieren Sie die Richtlinie auf jedem Knoten, damit Systems Manager Ausgaben und Protokolle an CloudWatch Logs

sendet. Fügen Sie Ihrer benutzerdefinierten Richtlinie die folgenden Richtlinienobjekte hinzu. Weitere Informationen zum Aktualisieren einer IAM-Richtlinie finden Sie unter Editing IAM policies (Bearbeiten von IAM-Richtlinien) im IAM-Benutzerhandbuch.

```
{
    "Effect": "Allow",
    "Action": "logs:DescribeLogGroups",
    "Resource": "*"
},
{
   "Effect":"Allow",
   "Action":[
      "logs:CreateLogGroup",
      "logs:CreateLogStream",
      "logs:DescribeLogStreams",
      "logs:PutLogEvents"
   ],
   "Resource":"arn:aws:logs:*:*:log-group:/aws/ssm/*"
},
```

Angeben von CloudWatch Logs beim Senden von Befehlen

Um CloudWatch Logs als Ausgabe festzulegen, wenn Sie einen Befehl von der AWS-Managementkonsole senden, wählen Sie im Abschnitt Output Options (Ausgabeoptionen) die Option CloudWatch Output (CloudWatch-Ausgabe) aus. Optional können Sie den Namen der CloudWatch Logs-Gruppe angeben, an die Sie die Befehlsausgabe senden möchten. Wenn Sie keinen Gruppennamen angeben, erstellt Systems Manager automatisch eine Protokollgruppe für Sie. Die Protokollgruppe verwendet das folgende Bezeichnungsformat: `/aws/ssm/`*`SystemsManagerDocumentName`*

Wenn Sie Befehle über die AWS CLI ausführen, geben Sie den Abschnitt `cloud-watch-output-config` in Ihrem Befehl an. Dieser Abschnitt ermöglicht Ihnen, den `CloudWatchOutputEnabled`-Parameter und optional den `CloudWatchLogGroupName`-Parameter anzugeben. Ein Beispiel.

Linux & macOS

```
aws ssm send-command \
    --instance-ids "instance ID" \
    --document-name "AWS-RunShellScript" \
    --parameters "commands=echo helloWorld" \
```

```
    --cloud-watch-output-config
 "CloudWatchOutputEnabled=true,CloudWatchLogGroupName=log group name"
```

Windows

```
aws ssm send-command ^
    --document-name "AWS-RunPowerShellScript" ^
    --parameters commands=["echo helloWorld"] ^
    --targets "Key=instanceids,Values=an instance ID" ^
    --cloud-watch-output-config '{"CloudWatchLogGroupName":"log group
 name","CloudWatchOutputEnabled":true}'
```

Anzeigen von Befehlsausgaben in CloudWatch Logs

Sobald der Befehl ausgeführt wird, sendet Systems Manager beinahe in Echtzeit Ausgaben an CloudWatch Logs. Die Ausgabe in CloudWatch Logs verwendet das folgende Format:

CommandID/*InstanceID*/*PluginID*/stdout

CommandID/*InstanceID*/*PluginID*/stderr

Die Ausgabe der Ausführung wird alle 30 Sekunden hochgeladen, oder wenn der Puffer mehr als 200 KB umfasst (je nachdem, was eher eintritt).

> **Note**
>
> Log Streams werden nur erstellt, wenn Ausgabedaten verfügbar sind. Wenn es beispielsweise keine Fehlerdaten für eine Ausführung gibt, wird der stderr-Stream nicht erstellt.

Hier sehen Sie ein Beispiel für die Befehlsausgabe, wie sie in CloudWatch Logs angezeigt wird.

```
Group - /aws/ssm/AWS-RunShellScript
Streams –
1234-567-8910/i-abcd-efg-hijk/AWS-RunPowerShellScript/stdout
24/1234-567-8910/i-abcd-efg-hijk/AWS-RunPowerShellScript/stderr
```

Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge

Amazon EventBridge ist ein Serverless-Event-Bus-Service, mit dem Sie Ihre Anwendungen mit Daten aus verschiedenen Quellen verbinden können. EventBridge stellt einen Stream von Echtzeitdaten aus Ihren eigenen Anwendungen, Software-as-a-Service-(SaaS)-Anwendungen und AWS-Services und leitet diese Daten dann an Ziele wie AWS Lambda weiter. Sie können Routing-Regeln einrichten, um festzulegen, wohin Ihre Daten gesendet werden. Auf diese Weise können Sie Anwendungsarchitekturen erstellen, die in Echtzeit auf alle Ihre Datenquellen reagieren. EventBridge ermöglicht es Ihnen, ereignisgesteuerte Architekturen zu erstellen, die lose gekoppelt und verteilt sind.

EventBridge wurde früher als Amazon CloudWatch Events bezeichnet. EventBridge enthält neue Funktionen, mit denen Sie Ereignisse aus SaaS-Partnern und Ihren eigenen Anwendungen empfangen können. Vorhandene Benutzer von CloudWatch Events können auf ihren vorhandenen Standardbus, ihre Regeln und ihre Ereignisse in der neuen EventBridge-Konsole und in der CloudWatch Events-Konsole zugreifen. EventBridge verwendet dieselbe CloudWatch Ereignis-API, sodass die gesamte vorhandene CloudWatch Events-API-Nutzung unverändert bleibt.

EventBridge kann Ereignisse aus Dutzenden von AWS-Services zu Ihren Regeln und Zielen von über 20 AWS-Services hinzufügen.

EventBridge bietet Unterstützung für AWS Systems Manager-Ereignisse und Systems Manager-Ziele.

Unterstützte Systems Manager Ereignistypen

Unter den vielen Typen von Systems Manager Ereignissen, die EventBridge erkennen kann, sind:

- Eine Statusaktualisierung für Just-in-Time-Knotenzugriffsanforderungen für manuelle Genehmigungen.
- Eine Just-in-Time-Knotenzugriffsanforderung ist fehlgeschlagen.
- Ein Wartungsfenster, das ausgeschaltet wird.
- Ein erfolgreiches Abschließen eines Automation-Workflows Automation ist ein Tool in AWS Systems Manager.
- Ein verwalteter Knoten, der außerhalb der Patch-Compliance liegt.
- Ein Parameterwert, der aktualisiert wird.

EventBridge unterstützt Ereignisse aus den folgenden AWS Systems Manager-Tools:

- Just-in-Time-Knotenzugriff (Ereignisse werden nach bestem Bemühen ausgegeben.)
- Automatisierung (Ereignisse werden auf bestmögliche Weise ausgegeben.)
- Change Calendar (Ereignisse werden auf bestmögliche Weise ausgegeben.)
- Compliance
- Inventory (Ereignisse werden auf bestmögliche Weise ausgegeben.)
- Maintenance Windows (Ereignisse werden auf bestmögliche Weise ausgegeben.)
- Parameter Store (Ereignisse werden auf bestmögliche Weise ausgegeben.)
- Run Command (Ereignisse werden auf bestmögliche Weise ausgegeben.)
- State Manager (Ereignisse werden auf bestmögliche Weise ausgegeben.)

Ausführliche Details zu unterstützten Systems Manager Ereignistypen finden Sie unter Referenz: EventBridge Amazon-Ereignistypen und -muster für Systems Manager und EventBridge Amazon-Veranstaltungsbeispiele für Systems Manager.

Unterstützte Systems Manager Zieltypen

EventBridge unterstützt die folgenden drei Systems-Manager-Tools als Ziele einer Ereignisregel:

- Ausführen eines Automation-Workflows
- Ausführen eines Run Command-Befehlsdokuments Ereignisse werden auf bestmögliche Weise ausgegeben.)
- Erstellen eines OpsCenter OpsItem

Die vorgeschlagenen Möglichkeiten, wie Sie diese Ziele verwenden können, finden Sie unter Beispielszenarien: Systems Manager Manager-Ziele in EventBridge Amazon-Regeln.

Weitere Informationen zu den ersten Schritten mit EventBridge und Einrichtungsregeln finden Sie unter Erste Schritte mit Amazon EventBridge im Benutzerhandbuch zu Amazon EventBridge. Weitere Informationen zum Arbeiten mit EventBridge finden Sie unter Amazon EventBridge-Benutzerhandbuch .

Themen

- Konfigurieren von EventBridge für Systems-Manager-Ereignisse

- EventBridge Amazon-Veranstaltungsbeispiele für Systems Manager
- Beispielszenarien: Systems Manager Manager-Ziele in EventBridge Amazon-Regeln

Konfigurieren von EventBridge für Systems-Manager-Ereignisse

Sie können Amazon EventBridge verwenden, um ein Zielereignis durchzuführen, wenn unterstützte AWS Systems Manager-Statusänderungen, Zustandsänderungen oder andere Bedingungen auftreten. Sie können eine Regel erstellen, die ausgeführt wird, sobald ein Statusübergang oder ein Übergang zu einem oder mehreren Status stattfindet, die für sie von Interesse sind.

Das folgende Verfahren enthält allgemeine Schritte zum Erstellen einer EventBridge-Regel, die aktiviert wird, wenn ein bestimmtes Ereignis von Systems Manager ausgelöst wird. Eine Liste der Verfahren in diesem Benutzerhandbuch, die bestimmte Szenarien behandelt, finden Sie unter Weiter Informationen am Ende dieses Themas.

> **Note**
>
> Wenn ein Service in Ihrem AWS-Konto ein Ereignis ausgibt, wird es stets an den Standard-Event-Bus Ihres Kontos weitergeleitet. Um eine Regel zu schreiben, die bei Ereignissen aus AWS-Services in Ihrem Konto reagiert, ordnen Sie die Regel dem Standard-Event-Bus zu. Sie können eine Regel für einen benutzerdefinierten Event Bus erstellen, der nach Ereignissen aus AWS-Services sucht. Diese Regel wird jedoch nur ausgelöst, wenn Sie ein solches Ereignis über die kontoübergreifende Ereignisbereitstellung aus einem anderen Konto erhalten. Weitere Informationen finden Sie unter Senden und Empfangen von Amazon-EventBridge-Ereignissen zwischen AWS-Konten im Benutzerhandbuch zu Amazon EventBridge.

Konfigurieren von EventBridge für Systems Manager-Ereignisse

1. Öffnen Sie die Amazon EventBridge-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Regeln aus.
3. Wählen Sie Regel erstellen aus.
4. Geben Sie einen Namen und eine Beschreibung für die Regel ein.

   Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben AWS-Region und auf demselben Event Bus haben.

5. Wählen Sie für Event Bus den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel auf übereinstimmende Ereignisse reagiert, die von Ihrem eigenen AWS-Konto stammen, wählen Sie Standard aus. Wenn ein AWS-Service in Ihrem Konto ein Ereignis ausgibt, wird es stets an den Standard-Event-Bus Ihres Kontos weitergeleitet.
6. Bei Regeltyp wählen Sie Regel mit einem Ereignismuster aus.
7. Wählen Sie Weiter aus.
8. Wählen Sie für Ereignisquelle AWS-Ereignisse oder EventBridge-Partnerereignisse.
9. Wählen Sie im Abschnitt Ereignismuster die Option Ereignismusterformular aus.
10. Als Event source (Ereignisquelle) wählen Sie AWS-Services aus.
11. Wählen Sie für AWS-Service, die Option Systems Manager aus.
12. Führen Sie für Type (Typ) eine der folgenden Aktionen aus:

    - Wählen Sie Add Events (Ereignisse hinzufügen) aus.

      Wenn Sie All Events (Alle Ereignisse) auswählen, stimmen alle von diesem Systems Manager-Service ausgegebenen Ereignisse mit der Regel überein. Beachten Sie, dass diese Option zu vielen Ereigniszielaktionen führen kann.
    - Wählen Sie den Typ des Systems Manager-Ereignisses aus, der für diese Regel verwendet werden soll. EventBridge unterstützt Ereignisse aus den folgenden AWS Systems Manager-Tools:
      - Automatisierung
      - Change Calendar
      - Compliance
      - Bestand
      - Maintenance Windows
      - Parameter Store
      - Run Command
      - State Manager

      > **Note**
      >
      > Für Systems Manager-Aktionen, die von EventBridge nicht unterstützt werden, können Sie einen AWS-API-Aufruf über CloudTrail auswählen, um eine Ereignisregel zu erstellen, die auf einem API-Aufruf basiert, der von CloudTrail aufgezeichnet wird.

Ein Beispiel finden Sie unter Überwachen der Sitzungsaktivität mithilfe von Amazon EventBridge (Konsole).

13. (Optional) Fügen Sie Filterwerte hinzu, um die Regel spezifischer zu gestalten. Wenn Sie beispielsweise State Manager ausgewählt haben und die Regel auf den Zustand einer einzelnen verwalteten Instanz beschränken möchten, die von einer Zuordnung anvisiert wird, wählen Sie für Specific type(s) (Spezifische(r) Typ(en)) EC2 State Manager Instance Association State Change aus.

    Ausführliche Informationen zu unterstützten Detailtypen finden Sie unter Referenz: EventBridge Amazon-Ereignistypen und -muster für Systems Manager.

    Einige Detailtypen verfügen über andere unterstützte Optionen wie den Status. Die verfügbaren Optionen hängen vom ausgewählten Tool ab.

14. Wählen Sie Weiter aus.
15. Bei Zieltypen wählen Sie AWS-Service aus.
16. Wählen Sie für Ziel auswählen ein Ziel aus, z. B. ein Amazon-SNS-Thema oder eine AWS Lambda-Funktion. Das Ziel wird ausgelöst, wenn ein Ereignis empfangen wird, das dem in der Regel definierten Ereignismuster entspricht.
17. Für viele Zieltypen benötigt EventBridge Berechtigungen zum Senden von Ereignissen an das Ziel. In diesen Fällen kann EventBridge die AWS Identity and Access Management-(IAM)-Rolle erstellen, die zum Ausführen Ihrer Regel erforderlich ist:

    - Um automatisch eine IAM-Rolle zu erstellen, wählen Sie Eine neue Rolle für diese spezifische Ressource erstellen.
    - Wenn Sie eine zuvor erstellte IAM-Rolle verwenden möchten, wählen Sie Use existing role (Vorhandene Rolle verwenden)

18. (Optional) Wählen Sie Add another target (Weiteres Ziel hinzufügen) aus, um ein weiteres Ziel für diese Regel hinzuzufügen.
19. Wählen Sie Weiter aus.
20. (Optional) Geben Sie ein oder mehrere Tags für die Regel ein. Weitere Informationen finden Sie unter Amazon-EventBridge-Tags im Benutzerhandbuch zu Amazon EventBridge.
21. Wählen Sie Weiter aus.
22. Überprüfen Sie die Details der Regel und wählen Sie dann Regel erstellen aus.

Weitere Informationen

- Erstellen eines EventBridge Ereignisses, das ein Runbook (Konsole) verwendet
- Übergabe von Daten an Automation mithilfe von Eingangstransformatoren
- Beheben von Compliance-Problemen mithilfe von EventBridge
- Aktionen zum Löschen von Inventar anzeigen in EventBridge
- Zu erstellende EventBridge Regeln konfigurieren OpsItems
- Konfigurieren von EventBridge-Regeln für Parameter und Parameter-Richtlinien

EventBridge Amazon-Veranstaltungsbeispiele für Systems Manager

Im Folgenden finden Sie Beispiele im JSON-Format für unterstützte EventBridge Ereignisse für AWS Systems Manager.

Systems Manager Ereignistypen

- AWS Systems Manager Automatisierungsereignisse
- AWS Systems Manager-Change Calendar-Ereignisse
- AWS Systems Manager-Change Manager-Ereignisse
- AWS Systems Manager Veranstaltungen zur Einhaltung von Vorschriften
- AWS Systems Manager-Maintenance Windows-Ereignisse
- AWS Systems Manager-Parameter Store-Ereignisse
- AWS Systems Manager-OpsCenter-Ereignisse
- AWS Systems Manager-Run Command-Ereignisse
- AWS Systems Manager-State Manager-Ereignisse

AWS Systems Manager Automatisierungsereignisse

Benachrichtigung über die Änderung des Automatisierungsschrittstatus

```
{
  "version": "0",
  "id": "eeca120b-a321-433e-9635-dab369006a6b",
  "detail-type": "EC2 Automation Step Status-change Notification",
  "source": "aws.ssm",
```

```
  "account": "123456789012",
  "time": "2024-11-29T19:43:35Z",
  "region": "us-east-1",
  "resources": ["arn:aws:ssm:us-east-2:123456789012:automation-
execution/333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "arn:aws:ssm:us-east-2:123456789012:automation-definition/runcommand1:1"],
  "detail": {
    "ExecutionId": "333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "Definition": "runcommand1",
    "DefinitionVersion": 1.0,
    "Status": "Success",
    "EndTime": "Nov 29, 2024 7:43:25 PM",
    "StartTime": "Nov 29, 2024 7:43:23 PM",
    "Time": 2630.0,
    "StepName": "runFixedCmds",
    "Action": "aws:runCommand"
  }
}
```

Benachrichtigung über die Änderung des Ausführungsstatus

```
{
  "version": "0",
  "id": "d290ece9-1088-4383-9df6-cd5b4ac42b99",
  "detail-type": "EC2 Automation Execution Status-change Notification",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-11-29T19:43:35Z",
  "region": "us-east-2",
  "resources": ["arn:aws:ssm:us-east-2:123456789012:automation-
execution/333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "arn:aws:ssm:us-east-2:123456789012:automation-definition/runcommand1:1"],
  "detail": {
    "ExecutionId": "333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "Definition": "runcommand1",
    "DefinitionVersion": 1.0,
    "Status": "Success",
    "StartTime": "Nov 29, 2024 7:43:20 PM",
    "EndTime": "Nov 29, 2024 7:43:26 PM",
    "Time": 5753.0,
    "ExecutedBy": "arn:aws:iam::123456789012:user/userName"
  }
}
```

AWS Systems Manager-Change Calendar-Ereignisse

Verwenden Sie die Informationen in diesem Thema, um das Verhalten von EventBridge Ereignissen für zu planen und zu verstehen AWS Systems ManagerChange Calendar.

> **Note**
>
> Statusänderungen für Kalender, die von anderen gemeinsam genutzt wurden, AWS-Konten werden derzeit nicht unterstützt.

Change CalendarIntegration mit Amazon EventBridge

AWS Systems Manager Change Calendarintegriert sich in Amazon EventBridge , um Sie über Änderungen des Kalenderstatus zu informieren. Beachten Sie die folgenden Verhaltensweisen im Zusammenhang mit der zugrunde liegenden Planungsarchitektur:

Timing und Zuverlässigkeit von Ereignissen

- EventBridge liefert Benachrichtigungen nach bestem Wissen und mit einer Termintoleranz von bis zu 15 Minuten.
- Zustandsänderungsereignisse spiegeln allgemeine Statusübergänge im Kalender wider, nicht einzelne Kalendereereignisse.
- Wenn mehrere Kalendereereignisse gleichzeitig auftreten, EventBridge wird für jede tatsächliche Änderung des Kalenderstatus nur ein Ereignis generiert.
- EventBridge löst Ereignisse nur aus, wenn der Gesamtstatus des Kalenders wechselt (z. B. von GESCHLOSSEN zu OFFEN), nicht für einzelne Kalendereereignisse, die nicht zu einer Statusänderung führen.
- Hinweisereignisse, die den Kalenderstatus nicht ändern, lösen keine EventBridge Benachrichtigungen aus.

Änderungen von Ereignissen und Hinweise zum Timing

- Wenn Sie Kalendereereignisse innerhalb von 15 Minuten nach ihrer geplanten Start- oder Endzeit ändern, werden EventBridge möglicherweise doppelte Benachrichtigungen generiert oder Benachrichtigungen verpasst.
- Dieses Verhalten tritt auf, weil das Planungssystem möglicherweise nicht genügend Zeit hat, um zuvor geplante Benachrichtigungen ordnungsgemäß zu aktualisieren oder zu stornieren.

- Bei wiederkehrenden Ereignissen wirkt sich dieses Verhalten in der Regel nur auf das erste Ereignis nach der Änderung aus.

Aufeinanderfolgende und sich überschneidende Ereignisse

- Wenn Kalenderereignisse in Abständen von 5 Minuten geplant sind, kann es je nach Statusänderung zu Statusübergangsereignissen kommen.
- Wenn Sie sich überschneidende Ereignisse in bestimmter Reihenfolge erstellen, können zusätzliche EventBridge Ereignisse generiert werden, auch wenn keine tatsächliche Statusänderung erfolgt.
- Um ein vorhersehbares Verhalten zu gewährleisten, sollten Sie es vermeiden, Kalenderereignisse kurz vor ihrer Ausführungszeit zu erstellen oder zu ändern.

Bewährte Methoden

- Entwerfen Sie Ihre EventBridge Regeln und die nachgelagerte Automatisierung so, dass potenzielle doppelte Ereignisse behandelt werden.
- Implementieren Sie Idempotenz in Ihre Automatisierungsworkflows, um Probleme durch doppelte Benachrichtigungen zu vermeiden.
- Planen Sie ausreichend Vorlaufzeit (mindestens 15 Minuten) ein, wenn Sie Kalenderereignisse erstellen oder ändern.
- Testen Sie Ihre EventBridge Integrationen gründlich anhand Ihrer spezifischen Kalenderereignismuster.

Kalender OFFEN


```
{
    "version": "0",
    "id": "47a3f03a-f30d-1011-ac9a-du3bdEXAMPLE",
    "detail-type": "Calendar State Change",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2024-09-19T18:00:07Z",
    "region": "us-east-2",
    "resources": [
        "arn:aws:ssm:us-east-2:123456789012:document/MyCalendar"
    ],
    "detail": {
        "state": "OPEN",
        "atTime": "2024-09-19T18:00:07Z",
        "nextTransitionTime": "2024-10-11T18:00:07Z"
```

```
    }
}
```

Kalender GESCHLOSSEN

```
{
    "version": "0",
    "id": "f30df03a-1011-ac9a-47a3-f761eEXAMPLE",
    "detail-type": "Calendar State Change",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2024-09-17T21:40:02Z",
    "region": "us-east-2",
    "resources": [
        "arn:aws:ssm:us-east-2:123456789012:document/MyCalendar"
    ],
    "detail": {
        "state": "CLOSED",
        "atTime": "2024-08-17T21:40:00Z",
        "nextTransitionTime": "2024-09-19T18:00:07Z"
    }
}
```

AWS Systems Manager-Change Manager-Ereignisse

Benachrichtigung über Statusaktualisierung der Änderungsanfrage – Beispiel 1

```
{
  "version": "0",
  "id": "feab80c1-a8ff-c721-b8b1-96ce70939696",
  "detail-type": "Change Request Status Update",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-10-24T10:51:52Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ssm:us-west-2:123456789012:opsitem/oi-12345abcdef",
    "arn:aws:ssm:us-west-2:123456789012:document/MyRunbook1"
  ],
  "detail": {
    "change-request-id": "d0585556-80f6-4522-8dad-dada6d45b67d",
    "change-request-title": "A change request title",
    "ops-item-id": "oi-12345abcdef",
```

```
    "ops-item-created-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "ops-item-created-time": "2024-10-24T10:50:33.180334Z",
    "ops-item-modified-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "ops-item-modified-time": "2024-10-24T10:50:33.180340Z",
    "ops-item-status": "InProgress",
    "change-template-document-name": "MyChangeTemplate",
    "runbook-document-arn": "arn:aws:ssm:us-west-2:123456789012:document/MyRunbook1",
    "runbook-document-version": "1",
    "auto-approve": true,
    "approvers": [
      "arn:aws:iam::123456789012:user/JaneDoe"
    ]
  }
}
```

Benachrichtigung über Statusaktualisierung der Änderungsanfrage – Beispiel 2

```
{
  "version": "0",
  "id": "25ce6b03-2e4e-1a2b-2a8f-6c9de8d278d2",
  "detail-type": "Change Request Status Update",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-10-24T10:51:52Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ssm:us-west-2:123456789012:opsitem/oi-abcdef12345",
    "arn:aws:ssm:us-west-2:123456789012:document/MyRunbook1"
  ],
  "detail": {
    "change-request-id": "d0585556-80f6-4522-8dad-dada6d45b67d",
    "change-request-title": "A change request title",
    "ops-item-id": "oi-abcdef12345",
    "ops-item-created-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "ops-item-created-time": "2024-10-24T10:50:33.180334Z",
    "ops-item-modified-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "ops-item-modified-time": "2024-10-24T10:50:33.997163Z",
    "ops-item-status": "Rejected",
    "change-template-document-name": "MyChangeTemplate",
    "runbook-document-arn": "arn:aws:ssm:us-west-2:123456789012:document/MyRunbook1",
    "runbook-document-version": "1",
    "auto-approve": true,
    "approvers": [
```

```
      "arn:aws:iam::123456789012:user/JaneDoe"
    ]
  }
}
```

AWS Systems Manager Veranstaltungen zur Einhaltung von Vorschriften

Im Folgenden finden Sie Beispiele für Veranstaltungen zum Thema AWS Systems Manager Compliance.

Zuordnung regelkonform

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-07-17T19:03:26Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "last-runtime": "2024-01-01T10:10:10Z",
    "compliance-status": "compliant",
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-type": "Association"
  }
}
```

Zuordnung nicht regelkonform

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-07-17T19:02:31Z",
  "region": "us-east-2",
  "resources": [
```

```
    "arn:aws:ssm:us-east-2:123456789012:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "last-runtime": "2024-01-01T10:10:10Z",
    "compliance-status": "non_compliant",
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-type": "Association"
  }
}
```

Patch regelkonform

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.123456789012",
  "account": "123456789012",
  "time": "2024-07-17T19:03:26Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-status": "compliant",
    "compliance-type": "Patch",
    "patch-baseline-id": "PB789",
    "severity": "critical"
  }
}
```

Patch nicht regelkonform

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-07-17T19:02:31Z",
```

```
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-status": "non_compliant",
    "compliance-type": "Patch",
    "patch-baseline-id": "PB789",
    "severity": "critical"
  }
}
```

AWS Systems Manager-Maintenance Windows-Ereignisse

Es folgen Beispiele der Ereignisse für Systems Manager Maintenance Windows.

Registrieren eines Ziels

Gültige Statuswerte sind `REGISTERED` und `DEREGISTERED`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Target Registration Notification",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2024-11-16T00:58:37Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-0ed7251d3fcf6e0c2",
      "arn:aws:ssm:us-east-2:123456789012:windowtarget/
e7265f13-3cc5-4f2f-97a9-7d3ca86c32a6"
   ],
   "detail":{
      "window-target-id":"e7265f13-3cc5-4f2f-97a9-7d3ca86c32a6",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "status":"REGISTERED"
   }
}
```

Fensterausführungstyp

Gültige Statuswerte sind unter anderem:

- CANCELLED
- CANCELLING
- FAILED
- IN_PROGRESS
- PENDING
- SKIPPED_OVERLAPPING
- SUCCESS TIMED_OUT


```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Execution State-change Notification",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2025-06-02T14:52:18Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-0c50858d01EXAMPLE"
   ],
   "detail":{
      "start-time":"2025-06-02T14:48:28.039273Z",
      "end-time":"2025-06-02T14:52:18.083773Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
      "status":"SUCCESS"
   }
}
```


Typ der Aufgabenausführung

Gültige Statuswerte sind IN_PROGRESS, SUCCESS, FAILED und TIMED_OUT.


```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Task Execution State-change Notification",
   "source":"aws.ssm",
```

```
   "account":"123456789012",
   "time":"2025-06-02T14:52:18Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-0c50858d01EXAMPLE"
   ],
   "detail":{
      "start-time":"2025-06-02T14:48:28.039273Z",
      "task-execution-id":"6417e808-7f35-4d1a-843f-123456789012",
      "end-time":"2025-06-02T14:52:18.083773Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"14bea65d-5ccc-462d-a2f3-e99c8EXAMPLE",
      "status":"SUCCESS"
   }
}
```

Aufgabenziel verarbeitet

Gültige Statuswerte sind `IN_PROGRESS`, `SUCCESS`, `FAILED` und `TIMED_OUT`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Task Target Invocation State-change Notification",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2025-06-02T14:52:18Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-123456789012345678"
   ],
   "detail":{
      "start-time":"2025-06-02T14:48:28.039273Z",
      "end-time":"2025-06-02T14:52:18.083773Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"791b72e0-f0da-4021-8b35-f95dfEXAMPLE",
      "task-execution-id":"c9b05aba-197f-4d8d-be34-e73fbEXAMPLE",
      "window-target-id":"e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE",
      "status":"SUCCESS",
      "owner-information":"Owner"
   }
}
```

Fensterstatusänderung

Gültige Statuswerte sind `ENABLED` und `DISABLED`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window State-change Notification",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2024-11-16T00:58:37Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ssm:us-east-2:123456789012:maintenancewindow/mw-0c50858d01EXAMPLE"
   ],
   "detail":{
      "window-id":"mw-0c50858d01EXAMPLE",
      "status":"DISABLED"
   }
}
```

AWS Systems Manager-Parameter Store-Ereignisse

Es folgen Beispiele der Ereignisse für Systems Manager Parameter Store.

Create Parameter (Parameter erstellen)

```
{
  "version": "0",
  "id": "6a7e4feb-b491-4cf7-a9f1-bf3703497718",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-05-22T16:43:48Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:parameter/MyExampleParameter"
  ],
  "detail": {
    "operation": "Create",
    "name": "MyExampleParameter",
    "type": "String",
```

```
    "description": "Sample Parameter"
  }
}
```

Update Parameter (Parameter aktualisieren)

```
{
  "version": "0",
  "id": "9547ef2d-3b7e-4057-b6cb-5fdf09ee7c8f",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-05-22T16:44:48Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:parameter/MyExampleParameter"
  ],
  "detail": {
    "operation": "Update",
    "name": "MyExampleParameter",
    "type": "String",
    "description": "Sample Parameter"
  }
}
```

DeleteParameter (Parameter löschen)

```
{
  "version": "0",
  "id": "80e9b391-6a9b-413c-839a-453b528053af",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-05-22T16:45:48Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ssm:us-east-2:123456789012:parameter/MyExampleParameter"
  ],
  "detail": {
    "operation": "Delete",
    "name": "MyExampleParameter",
    "type": "String",
    "description": "Sample Parameter"
```

```
  }
}
```

AWS Systems Manager-OpsCenter-Ereignisse

OpsCenter OpsItem Benachrichtigung erstellen

```
{
  "version": "0",
  "id": "aae66adc-7aac-f0c0-7854-7691e8c079b8",
  "detail-type": "OpsItem Create",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-10-19T02:48:11Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ssm:us-west-2:123456789012:opsitem/oi-123456abcdef"
  ],
  "detail": {
    "created-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "created-time": "2024-10-19T02:46:53.629361Z",
    "source": "aws.ssm",
    "status": "Open",
    "ops-item-id": "oi-123456abcdef",
    "title": "An issue title",
    "ops-item-type": "/aws/issue",
    "description": "A long description may appear here"
  }
}
```

OpsCenter OpsItem Benachrichtigung aktualisieren

```
{
  "version": "0",
  "id": "2fb5b168-b725-41dd-a890-29311200089c",
  "detail-type": "OpsItem Update",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2024-10-19T02:48:11Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ssm:us-west-2:123456789012:opsitem/oi-123456abcdef"
  ],
```

```
  "detail": {
    "created-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "created-time": "2024-10-19T02:46:54.049271Z",
    "modified-by": "arn:aws:iam::123456789012:user/JohnDoe",
    "modified-time": "2024-10-19T02:46:54.337354Z",
    "source": "aws.ssm",
    "status": "Open",
    "ops-item-id": "oi-123456abcdef",
    "title": "An issue title",
    "ops-item-type": "/aws/issue",
    "description": "A long description may appear here"
  }
}
```

AWS Systems Manager-Run Command-Ereignisse

Run Command-Statusänderungsbenachrichtigung

```
{
    "version": "0",
    "id": "51c0891d-0e34-45b1-83d6-95db273d1602",
    "detail-type": "EC2 Command Status-change Notification",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2024-07-10T21:51:32Z",
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2:123456789012:instance/i-02573cafcfEXAMPLE"],
    "detail": {
        "command-id": "e8d3c0e4-71f7-4491-898f-c9b35bee5f3b",
        "document-name": "AWS-RunPowerShellScript",
        "expire-after": "2024-07-14T22:01:30.049Z",
        "parameters": {
            "executionTimeout": ["3600"],
            "commands": ["date"]
        },
        "requested-date-time": "2024-07-10T21:51:30.049Z",
        "status": "Success"
    }
}
```

Run CommandInvocation Status-Änderungsbenachrichtigung

```
{
```

```
    "version": "0",
    "id": "4780e1b8-f56b-4de5-95f2-95dbEXAMPLE",
    "detail-type": "EC2 Command Invocation Status-change Notification",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2024-07-10T21:51:32Z",
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2:123456789012:instance/i-02573cafcfEXAMPLE"],
    "detail": {
        "command-id": "e8d3c0e4-71f7-4491-898f-c9b35bee5f3b",
        "document-name": "AWS-RunPowerShellScript",
        "instance-id": "i-02573cafcfEXAMPLE",
        "requested-date-time": "2024-07-10T21:51:30.049Z",
        "status": "Success"
    }
}
```

AWS Systems Manager-State Manager-Ereignisse

State Manager-Association-Statusänderung

```
{
   "version":"0",
   "id":"db839caf-6f6c-40af-9a48-25b2ae2b7774",
   "detail-type":"EC2 State Manager Association State Change",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2024-05-16T23:01:10Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ssm:us-east-2::document/AWS-RunPowerShellScript"
   ],
   "detail":{
      "association-id":"6e37940a-23ba-4ab0-9b96-5d0a1a05464f",
      "document-name":"AWS-RunPowerShellScript",
      "association-version":"1",
      "document-version":"Optional.empty",
      "targets":"[{\"key\":\"InstanceIds\",\"values\":[\"i-12345678\"]}]",
      "creation-date":"2024-02-13T17:22:54.458Z",
      "last-successful-execution-date":"2024-05-16T23:00:01Z",
      "last-execution-date":"2024-05-16T23:00:01Z",
      "last-updated-date":"2024-02-13T17:22:54.458Z",
      "status":"Success",
```

```
      "association-status-aggregated-count":"{\"Success\":1}",
      "schedule-expression":"cron(0 */30 * * * ? *)",
      "association-cwe-version":"1.0"
   }
}
```

State Manager-Instance-Association-Statusänderung

```
{
   "version":"0",
   "id":"6a7e8feb-b491-4cf7-a9f1-bf3703467718",
   "detail-type":"EC2 State Manager Instance Association State Change",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2024-02-23T15:23:48Z",
   "region":"us-east-2",
   "resources":[
      "arn:aws:ec2:us-east-2:123456789012:instance/i-12345678",
      "arn:aws:ssm:us-east-2:123456789012:document/my-custom-document"
   ],
   "detail":{
      "association-id":"34fcb7e0-9a14-4984-9989-0e04e3f60bd8",
      "instance-id":"i-02573cafcfEXAMPLE",
      "document-name":"my-custom-document",
      "document-version":"1",
      "targets":"[{\"key\":\"instanceids\",\"values\":[\"i-02573cafcfEXAMPLE\"]}]",
      "creation-date":"2024-02-23T15:23:48Z",
      "last-successful-execution-date":"2024-02-23T16:23:48Z",
      "last-execution-date":"2024-02-23T16:23:48Z",
      "status":"Success",
      "detailed-status":"",
      "error-code":"testErrorCode",
      "execution-summary":"testExecutionSummary",
      "output-url":"sampleurl",
      "instance-association-cwe-version":"1"
   }
}
```

Beispielszenarien: Systems Manager Manager-Ziele in EventBridge Amazon-Regeln

Wenn Sie das aufzurufende Ziel in einer EventBridge Amazon-Regel angeben, können Sie aus über 20 Zieltypen wählen und jeder Regel bis zu fünf Ziele hinzufügen.

Unter den verschiedenen Zielen können Sie zwischen Automatisierung und OpsCenterRun Command, bei denen es sich um Tools handelt AWS Systems Manager, als Zielaktionen wählen, wenn ein EventBridge Ereignis eintritt.

Im Folgenden finden Sie einige Beispiele dafür, wie Sie diese Tools als Ziel einer EventBridge Regel verwenden können.

Beispiele zur Automation

Sie können eine EventBridge Regel so konfigurieren, dass Automatisierungsworkflows gestartet werden, wenn Ereignisse wie die folgenden eintreten:

- Wenn ein CloudWatch Amazon-Alarm meldet, dass ein verwalteter Knoten eine Statusprüfung (`StatusCheckFailed_Instance=1`) nicht bestanden hat, führen Sie das `AWSSupport-ExecuteEC2Rescue` Automation-Runbook auf dem Knoten aus.
- Wenn ein `EC2 Instance State-change Notification` Ereignis eintritt, weil eine neue Amazon Elastic Compute Cloud (Amazon EC2) -Instance läuft, führen Sie das `AWS-AttachEBSVolume` Automation-Runbook auf der Instance aus.
- Wenn ein Amazon Elastic Block Store (Amazon EBS)-Volume erstellt wurde und verfügbar ist, führen Sie das aus `AWS-CreateSnapshot`–Automation-Runbook auf dem Volume aus.

Beispiele für OpsCenter

Sie können eine EventBridge Regel konfigurieren, um eine neue zu erstellenOpsItem, wenn Vorfälle wie die folgenden auftreten:

- Ein Drosselungssereignis für Amazon DynamoDB tritt auf, oder die Leistung des Amazon EBS-Volumes hat sich verschlechtert.
- Eine Amazon EC2 Auto Scaling Scaling-Gruppe kann einen Knoten nicht starten, oder ein Systems Manager Automation-Workflow schlägt fehl.
- Eine EC2 Instance ändert ihren Status von `Running` zu`Stopped`.

Beispiele für Run Command

Sie können eine EventBridge Regel so konfigurieren, dass ein Systems Manager Manager-Befehlsdokument ausgeführt wird, Run Command wenn Ereignisse wie die folgenden eintreten:

- Wenn eine Auto-Scaling-Gruppe kurz vor dem Ende steht, könnte ein Run Command-Skript die Protokolldateien des Knotens erfassen, bevor er beendet wird.
- Wenn ein neuer Knoten in einer Auto-Scaling-Gruppe erstellt wird, könnte eine Run Command-Zielaktion die Webserver-Rolle aktivieren oder Software auf dem Knoten installieren.
- Wenn ein verwalteter Knoten als nicht konform befunden wird, könnte eine Run Command-Zielaktion Patches auf dem Knoten aktualisieren, indem das `AWS-RunPatchBaseline`-Dokument ausgeführt wird.

Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen

Sie können Amazon Simple Notification Service (Amazon SNS) zum Senden von Benachrichtigungen über den Status der Befehle konfigurieren, die Sie mithilfe von Run Command oder Maintenance Windows, Tools in AWS Systems Manager, senden. Amazon SNS koordiniert und verwaltet das Senden und Zustellen von Benachrichtigungen an Clients oder Endpunkte, die Amazon SNS-Themen abonniert haben. Sie können eine Benachrichtigung erhalten, wenn ein Befehl in einen neuen Status oder in einen bestimmten Status wechselt, z. B. Ausgefallen oder Timeout. In Fällen, in denen Sie einen Befehl an mehrere Knoten senden, können Sie eine Benachrichtigung für jede Kopie des Befehls abrufen, die an einen bestimmten Knoten gesendet wurde. Jede Kopie wird als Aufruf bezeichnet.

Amazon SNS kann Benachrichtigungen als HTTP oder HTTPS POST, E-Mail (SMTP, entweder im Klartext oder im JSON-Format) oder als Nachricht, die an eine Amazon Simple Queue Service (Amazon SQS)-Queue gesendet wird, verschicken. Weitere Informationen finden Sie unter Was ist Amazon SNS im Amazon Simple Notification Service-Entwicklerhandbuch. Beispiele für die Struktur der JSON-Daten in der Amazon SNS-Benachrichtigung, die von Run Command und Maintenance Windows bereitgestellt wird, finden Sie unter Beispiele für Amazon SNS-Benachrichtigungen für AWS Systems Manager.

> **Important**
>
> Notieren Sie die folgenden wichtigen Informationen.

- FIFO-Themen des Amazon Simple Notification Service werden nicht unterstützt.
- Amazon Q Developer in Chat-Anwendungen wird für die Überwachung von Systems Manager mit Amazon SNS nicht unterstützt. Wenn Sie Amazon Q Developer in Chat-Anwendungen zur Überwachung von Systems Manager verwenden möchten, müssen Sie es mit Amazon verwenden EventBridge. Hinweise zur Überwachung von Systems Manager mithilfe von EventBridge Systems Manager finden Sie unterÜberwachung von Systems Manager-Ereignissen mit Amazon EventBridge. Informationen zu Amazon EventBridge und Amazon Q Developer in Chat-Anwendungen finden Sie unter Tutorial: Erstellen einer EventBridge Regel, die Benachrichtigungen an Amazon Q Developer in Chat-Anwendungen sendet im Administratorhandbuch für Amazon Q Developer in Chat-Anwendungen.

Konfigurieren von Amazon SNS-Benachrichtigungen für AWS Systems Manager

Run Command und Maintenance Windows-Aufgaben, die für ein Wartungsfenster registriert sind, können Amazon SNS-Benachrichtigungen zu Befehlsaufgaben senden, die in folgende Status wechseln:

- In Bearbeitung
- Herzlichen Glückwunsch
- Fehlgeschlagen
- Timed Out (Zeitüberschreitung)
- Abgebrochen

Weitere Informationen über die Bedingungen, die dazu führen, dass ein Befehl in einen dieser Status wechselt, finden Sie unter Grundlegendes zu Befehlsstatus.

> **Note**
>
> Befehle, die mit Run Command gesendet wurden, melden auch die Status „Cancelling (Abbrechen)“ und „Pending (Ausstehend)“. Diese Status werden nicht von Amazon SNS-Benachrichtigungen erfasst.

Amazon SNS-Benachrichtigungen

Wenn Sie Run Command oder eine Run Command-Aufgabe in Ihrem Wartungsfenster für Amazon SNS-Benachrichtigungen konfigurieren, sendet Amazon SNS zusammenfassende Nachrichten mit folgenden Informationen.

Feld	Typ	Description
eventTime	Zeichenfolge	Der Zeitpunkt, an dem das Ereignis initiiert wurde. Der Zeitstempel ist wichtig, weil Amazon SNS keine Garantie für die Reihenfol ge der Nachrichtenzustell ung übernimmt. Beispiel: 2016-04-26T13:15:30Z
documentName	Zeichenfolge	Der Name des SSM-Dokum ents, das zur Ausführung dieses Befehls verwendet wurde.
commandId	Zeichenfolge	Die von Run Command erstellte ID, nachdem der Befehl gesendet wurde.
expiresAfter	Date	Wenn diese Zeit erreicht ist und der Befehl noch nicht gestartet wurde, wird er nicht ausgeführt.
Ausgänge: 3 BucketName	Zeichenfolge	Der Amazon Simple Storage Service (Amazon S3)-Bucket, in dem die Antworten auf die Befehlsausführung gespeiche rt werden sollten.

Feld	Typ	Description
Ausgänge 3 KeyPrefix	Zeichenfolge	Der Amazon S3-Verzeichnispfad innerhalb des Buckets, in dem die Antworten auf die Befehlsausführung gespeichert werden sollten.
requestedDateTime	Zeichenfolge	Die Uhrzeit und das Datum, an dem die Anforderung an diesen spezifischen Knoten gesendet wurde.
instanceIds	StringList	Die Knoten, auf die der Befehl abzielt hat. **Note** Instanzen IDs sind nur dann in der Übersichtsnachricht enthalten, wenn die Run Command Aufgabe IDs direkt auf die Instanz abzielte. Instanzen IDs sind nicht in der Übersichtsnachricht enthalten, wenn die Run Command Aufgabe mithilfe von Tag-basiertem Targeting ausgegeben wurde.
Status	Zeichenfolge	Befehlsstatus für den Befehl.

Aufrufbasierte Amazon SNS-Benachrichtigungen

Wenn Sie einen Befehl an mehrere Knoten senden, kann Amazon SNS Nachrichten über jede Kopie oder jeden Aufruf des Befehls senden. Die Nachrichten enthalten folgende Angaben.

Feld	Typ	Description
eventTime	Zeichenfolge	Der Zeitpunkt, an dem das Ereignis initiiert wurde. Der Zeitstempel ist wichtig, weil Amazon SNS keine Garantie für die Reihenfol ge der Nachrichtenzustell ung übernimmt. Beispiel: 2016-04-26T13:15:30Z
documentName	Zeichenfolge	Der Name des System Manager-Dokuments (SSM-Dokument), das zur Ausführun g dieses Befehls verwendet wurde.
requestedDateTime	Zeichenfolge	Die Uhrzeit und das Datum, an dem die Anforderung an diesen spezifischen Knoten gesendet wurde.
commandId	Zeichenfolge	Die von Run Command erstellte ID, nachdem der Befehl gesendet wurde.
instanceId	Zeichenfolge	Die Instance, auf die der Befehl abzielt.
Status	Zeichenfolge	Befehlsstatus für diesen Aufruf.

Um Amazon SNS-Benachrichtigungen einzurichten, wenn ein Befehl seinen Status ändert, führen Sie die folgenden Aufgaben aus.

> **Note**
>
> Wenn Sie keine Amazon SNS-Benachrichtigungen für Ihr Wartungsfenster konfigurieren, können Sie Aufgabe 5 weiter unten in diesem Thema überspringen.

Themen

- Aufgabe 1: Erstellen und Abonnieren eines Amazon SNS-Themas
- Aufgabe 2: Erstellen einer IAM-Richtlinie für Amazon SNS-Benachrichtigungen
- Aufgabe 3: Erstellen einer IAM-Role für Amazon SNS-Benachrichtigungen
- Aufgabe 4: Konfigurieren des Benutzerzugriffs
- Aufgabe 5: Hängen Sie die iam: PassRole -Richtlinie an Ihre Rolle im Wartungsfenster an

Aufgabe 1: Erstellen und Abonnieren eines Amazon SNS-Themas

Ein Amazon SNS-Thema ist ein Kommunikationskanal, über den Run Command und Run Command -Aufgaben, die in einem Wartungsfenster registriert sind, Benachrichtigungen über den Status Ihrer Befehle senden. Amazon SNS unterstützt verschiedene Kommunikationsprotokolle, darunter HTTP/S, E-Mail und andere Protokolle AWS-Services wie Amazon Simple Queue Service (Amazon SQS). Für den Anfang empfehlen wir, mit dem E-Mail-Protokoll anzufangen. Weitere Informationen zum Erstellen eines Themas finden Sie unter Erstellen eines Amazon-SNS-Themas im Entwicklerhandbuch zu Amazon Simple Notification Service.

> **Note**
>
> Nachdem Sie das Thema erstellt haben, kopieren oder notieren Sie sich die Thema-ARN. Sie legen diesen ARN fest, wenn Sie einen Befehl senden, der so konfiguriert wurde, dass er Statusbenachrichtigungen zurückgibt.

Nachdem Sie das Thema erstellt haben, abonnieren Sie es, indem Sie einen Endpunkt angeben. Wenn Sie das E-Mail-Protokoll gewählt haben, ist der Endpunkt die E-Mail-Adresse, an die Sie Benachrichtigungen erhalten möchten. Weitere Informationen zum Abonnieren eines Themas finden

Sie unter Abonnieren eines Amazon-SNS-Themas im Entwicklerhandbuch zu Amazon Simple Notification Service.

Amazon SNS sendet eine Bestätigungs-E-Mail von AWS Notifications an die von Ihnen angegebene E-Mail-Adresse. Öffnen Sie die E-Mail und wählen Sie den Link Abonnement bestätigen aus.

Sie erhalten eine Bestätigungsnachricht von. AWS Amazon SNS ist jetzt so konfiguriert, dass Benachrichtigungen empfangen und als E-Mail an die angegebene E-Mail-Adresse gesendet werden.

Aufgabe 2: Erstellen einer IAM-Richtlinie für Amazon SNS-Benachrichtigungen

Gehen Sie wie folgt vor, um eine benutzerdefinierte Richtlinie AWS Identity and Access Management (IAM) zu erstellen, die Berechtigungen für die Initiierung von Amazon SNS SNS-Benachrichtigungen bereitstellt.

So erstellen Sie eine benutzerdefinierte IAM-Richtlinie für Amazon SNS-Benachrichtigungen

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Richtlinien und dann Richtlinie erstellen. (Wenn die Schaltfläche Get Started (Erste Schritte) angezeigt wird, klicken Sie darauf und wählen Sie anschließend Create Policy (Richtlinie erstellen) aus.)
3. Wählen Sie den Tab JSON.
4. Ersetzen Sie den Standardinhalt durch einen der folgenden Inhalte, je nachdem, ob das Amazon SNS SNS-Thema AWS KMS Verschlüsselung verwendet:

   SNS topic not encrypted

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sns:Publish"
            ],
            "Resource": "arn:aws:sns:us-east-1:111122223333:sns-topic-
name"
        }
```

```
    ]
}
```

*region*steht für die Kennung einer Region, die von AWS-Region unterstützt wird AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

*account-id*steht für den 12-stelligen Bezeichner für Ihren AWS-Konto, im folgenden Format. `123456789012`

*sns-topic-name*steht für den Namen des Amazon SNS SNS-Themas, das Sie für die Veröffentlichung von Benachrichtigungen verwenden möchten.

SNS topic encrypted

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sns:Publish"
            ],
            "Resource": "arn:aws:sns:us-east-1:111122223333:sns-topic-
name"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/kms-key-
id"
        }
    ]
}
```

*region*steht für die Kennung einer Region, die von AWS-Region unterstützt wird AWS Systems Manager, z. B. `us-east-2` für die Region USA Ost (Ohio). Eine Liste der unterstützten *region* Werte finden Sie in der Spalte Region der Systems Manager Manager-Dienstendpunkte in der Allgemeine Amazon Web Services-Referenz.

*account-id*steht für den 12-stelligen Bezeichner für Ihren AWS-Konto, im folgenden Format. `123456789012`

*sns-topic-name*steht für den Namen des Amazon SNS SNS-Themas, das Sie für die Veröffentlichung von Benachrichtigungen verwenden möchten.

*kms-key-id*stellt die ID des KMS-Schlüssels für die symmetrische Verschlüsselung dar, der für AWS KMS die Verschlüsselung und Entschlüsselung des Themas verwendet werden soll, im folgenden Format. `1234abcd-12ab-34cd-56ef-12345EXAMPLE`

> **Note**
>
> Die Verwendung der Verschlüsselung ist kostenpflichtig. AWS KMS
> Weitere Informationen finden Sie unter Verwalten von Amazon-SNS-Verschlüsselungsschlüsseln und Kosten im AWS Key Management Service - Entwicklerhandbuch.

5. Wählen Sie Weiter: Tags aus.
6. (Optional) Fügen Sie ein oder mehrere Tag-Schlüssel-Wert-Paare hinzu, um den Zugriff für diese Richtlinie zu organisieren, zu verfolgen oder zu steuern.
7. Wählen Sie Weiter: Prüfen aus.
8. Geben Sie auf der Seite Richtlinie prüfen im Feld Name einen Namen für die Inline-Richtlinie ein. Beispiel: **`my-sns-publish-permissions`**.
9. (Optional) Geben Sie im Feld Beschreibung eine Beschreibung für die Richtlinie ein.
10. Wählen Sie Richtlinie erstellen aus.

Aufgabe 3: Erstellen einer IAM-Role für Amazon SNS-Benachrichtigungen

Verwenden Sie das folgende Verfahren zum Erstellen einer IAM-Rolle für Amazon SNS-Benachrichtigungen. Diese Service-Rolle wird von Systems Manager verwendet, um Amazon SNS-

Benachrichtigungen zu initiieren. In allen nachfolgenden Verfahren wird diese Rolle als Amazon SNS IAM-Rolle bezeichnet.

So erstellen Sie eine IAM-Service-Rolle für Amazon SNS-Benachrichtigungen

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Klicken Sie im Navigationsbereich der IAM-Konsole auf Rollen, und wählen Sie dann Rolle erstellen.
3. Wählen Sie den AWS-Service-Rollentyp und dann Systems Manager aus.
4. Wählen Sie den Systems-Manager-Anwendungsfall aus. Wählen Sie anschließend Weiter.
5. Markieren Sie auf der Seite Attach permissions policies (Richtlinien für Berechtigungen anfügen) das Kontrollkästchen links neben dem Namen der benutzerdefinierten Richtlinie aus, die Sie in Aufgabe 2 erstellt haben. Beispiel: **`my-sns-publish-permissions`**.
6. (Optional) Legen Sie eine Berechtigungsgrenze fest. Dies ist ein erweitertes Feature, das für Servicerollen verfügbar ist, aber nicht für servicegebundene Rollen.

   Öffnen Sie den Abschnitt Permissions boundary (Berechtigungsgrenze) und wählen Sie Use a permissions boundary to control the maximum role permissions (Eine Berechtigungsgrenze verwenden, um die maximalen Rollen-Berechtigungen zu steuern). IAM enthält eine Liste der AWS verwalteten und kundenverwalteten Richtlinien in Ihrem Konto. Wählen Sie die Richtlinie aus, die für die Berechtigungsgrenze verwendet werden soll, oder wählen Create policy (Richtlinie erstellen), um eine neue Registerkarte im Browser zu öffnen und eine vollständig neue Richtlinie zu erstellen. Weitere Informationen finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch. Nachdem Sie die Richtlinie erstellt haben, schließen Sie diese Registerkarte und kehren Sie zu Ihrer ursprünglichen Registerkarte zurück, um die Richtlinie auszuwählen, die für die Berechtigungsgrenze verwendet werden soll.
7. Wählen Sie Weiter aus.
8. Geben Sie möglichst einen Rollennamen oder ein Rollennamen-Suffix ein, mit dem der Zweck dieser Rolle einfach zu erkennen ist. Rollennamen müssen innerhalb Ihres AWS-Konto-Kontos eindeutig sein. Sie werden nicht nach Groß- und Kleinschreibung unterschieden. z. B. können Sie keine Rollen erstellen, die **`PRODROLE`** bzw. **`prodrole`** heißen. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung nicht bearbeitet werden.
9. (Optional) Geben Sie unter Role description (Rollenbeschreibung) eine Beschreibung für die neue Rolle ein.

10. Wählen Sie in den Abschnitten Step 1: Select trusted entities (Schritt 1: Vertrauenswürdige Entitäten auswählen) oder Step 2: Add permissions (Schritt 2: Berechtigungen hinzufügen) die Option Edit (Bearbeiten), um die Anwendungsfälle und Berechtigungen für die Rolle zu bearbeiten.
11. (Optional) Fügen Sie dem Benutzer Metadaten hinzu, indem Sie Markierungen als Schlüssel-Wert-Paare anfügen. Weitere Informationen zur Verwendung von Tags in IAM finden Sie unter Markieren von IAM-Ressourcen im IAM-Benutzerhandbuch.
12. Prüfen Sie die Rolle und klicken Sie dann auf Rolle erstellen.
13. Wählen Sie den Namen der Rolle und kopieren Sie dann oder notieren Sie sich den Wert der Role ARN (Rollen-ARN). Dieser Amazon-Ressourcenname (ARN) für die Rolle wird verwendet, wenn Sie einen Befehl senden, der so konfiguriert wurde, dass er Amazon-SNS-Benachrichtigungen zurückgibt.
14. Lassen Sie die Seite Summary (Übersicht) geöffnet.

Aufgabe 4: Konfigurieren des Benutzerzugriffs

Wenn einer IAM-Entität (Benutzer, Rolle oder Gruppe) Administratorberechtigungen zugewiesen wurden, hat der Benutzer oder die Rolle Zugriff auf Run Command und Maintenance Windows, Tools in AWS Systems Manager.

Für Entitäten ohne Administratorrechte muss ein Administrator der IAM-Entität die folgenden Berechtigungen gewähren:

- Die von `AmazonSSMFullAccess` verwaltete Richtlinie oder eine Richtlinie, die vergleichbare Berechtigungen bereitstellt.
- `iam:PassRole`-Berechtigungen für die Rolle, die in Aufgabe 3: Erstellen einer IAM-Role für Amazon SNS-Benachrichtigungen erstellt wurde. Beispiel:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
```

```
            "Resource": "arn:aws:iam::111122223333:role/sns-role-name",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ssm.amazonaws.com"
                }
            }
        }
    ]
}
```

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

  Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center -Benutzerhandbuch.
- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

  Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anleitung unter Eine Rolle für einen externen Identitätsanbieter (Verbund) erstellen im IAM-Benutzerhandbuch.
- IAM-Benutzer:
  - Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Befolgen Sie die Anleitung unter Eine Rolle für einen IAM-Benutzer erstellen im IAM-Benutzerhandbuch.
  - (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter Hinzufügen von Berechtigungen zu einem Benutzer (Konsole) im IAM-Benutzerhandbuch.

So konfigurieren Sie Benutzerzugriff und fügen die Richtlinie **`iam:PassRole`** an ein Benutzerkonto an

1. Wählen Sie im IAM-Navigationsbereich die Option Users (Benutzer) und anschließend das Benutzerkonto aus, das Sie konfigurieren möchten.
2. Prüfen Sie auf der Registerkarte Permissions (Berechtigungen) in der Richtlinienliste, ob entweder die **`AmazonSSMFullAccess`**-Richtlinie aufgeführt ist, oder ob es eine vergleichbare Richtlinie gibt, die dem Konto Berechtigungen für den Zugriff auf Systems Manager erteilt.
3. Wählen Sie Inline-Richtlinie hinzufügen.
4. Wählen Sie auf der Seite Richtlinie erstellen die Registerkarte Visueller Editor aus.

5. Wählen Sie Choose a service und dann IAM aus.
6. Geben Sie für Aktionen in das Textfeld Aktionen filtern den Text ein**`PassRole`**, und aktivieren Sie dann das Kontrollkästchen neben PassRole.
7. Vergewissern Sie sich bei Resources (Ressourcen), dass Specific (spezifisch) ausgewählt ist, und wählen Sie dann Add ARN (ARN hinzufügen).
8. Fügen Sie im Feld Specify ARN for role (ARN für Rolle angeben) den ARN der Amazon SNS IAM-Rolle ein, den Sie am Ende von Aufgabe 3 kopiert haben. Das System füllt die Felder Account (Konto) und Role name with path (Rollenname mit Pfad) automatisch aus.
9. Wählen Sie Hinzufügen aus.
10. Wählen Sie Richtlinie prüfen.
11. Geben Sie auf der Seite Review Policy (Richtlinie überprüfen) einen Namen ein und wählen Sie anschließend Create Policy (Richtlinie erstellen) aus.

Aufgabe 5: Hängen Sie die iam: PassRole -Richtlinie an Ihre Rolle im Wartungsfenster an

Wenn Sie eine Run Command-Aufgabe mit einem Wartungsfenster registrieren, geben Sie den Amazon-Ressourcennamen (ARN) einer Servicerolle an. Diese Servicerolle wird von Systems Manager zur Durchführung von Aufgaben verwendet, die beim Wartungsfenster registriert sind. Hängen Sie eine `iam:PassRole`-Richtlinie an die angegebene Wartungsfenster-Servicerolle an, um Amazon SNS-Benachrichtigungen für eine registrierte Run Command-Aufgabe zu konfigurieren Wenn Sie nicht beabsichtigen, die registrierte Aufgabe für Amazon SNS-Benachrichtigungen zu konfigurieren, können Sie diese Aufgabe überspringen.

Mithilfe der `iam:PassRole`-Richtlinie kann die Maintenance Windows-Servicerolle die in Aufgabe 3 erstellte IAM-Rolle an den Amazon SNS-Service übergeben. Das folgende Verfahren zeigt, wie die `iam:PassRole`-Richtlinie an die Maintenance Windows-Servicerolle angehängt wird.

> **Note**
>
> Verwenden Sie eine benutzerdefinierte Servicerolle für Wartungsfenster, um Benachrichtigungen mit Bezug zu den registrierten Run Command-Aufgaben zu senden. Weitere Informationen finden Sie unter Einrichten von Maintenance Windows.
> Wenn Sie eine benutzerdefinierte Servicerolle für Wartungsfenster-Aufgaben erstellen müssen, finden Sie weitere Informationen dazu unter Einrichten von Maintenance Windows.

Anhängen der **`iam:PassRole`**-Richtlinie an Ihre Maintenance Windows-Rolle

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich erst Roles (Rollen) und dann die in Aufgabe 3 erstellte Amazon SNS IAM-Rolle aus.
3. Kopieren oder notieren Sie den Role ARN (Rollen-ARN) und kehren Sie zum Abschnitt Roles (Rollen) der IAM-Konsole zurück.
4. Wählen Sie die benutzerdefinierte Maintenance Windows-Servicerolle aus, die Sie aus der Liste Role name (Rollennamen) erstellt haben.
5. Stellen Sie in der Registerkarte Permissions (Berechtigungen) sicher, dass entweder die `AmazonSSMMaintenanceWindowRole`-Richtlinie oder eine vergleichbare Richtlinie angegeben ist, durch die Wartungsfenster Berechtigungen für die Systems-Manager-API erhalten. Wenn dies nicht der Fall ist, wählen Sie Berechtigungen hinzufügen, Richtlinien anfügen, um sie anzufügen.
6. Wählen Sie Berechtigungen hinzufügen, eingebundene Richtlinie erstellen.
7. Wählen Sie die Registerkarte Visual Editor (Visueller Editor) aus.
8. Wählen Sie unter Service die Option IAM aus.
9. Geben Sie für Aktionen in das Textfeld Aktionen filtern den Text ein**`PassRole`**, und aktivieren Sie dann das Kontrollkästchen neben PassRole.
10. Wählen Sie unter Resources (Ressourcen), die Option Specific (Spezifisch) und dann Add ARN (ARN hinzufügen)aus.
11. Fügen Sie im Feld Specify ARN for role (ARN für Rolle angeben) den ARN der in Aufgabe 3 erstellten Amazon SNS IAM-Rolle ein und wählen Sie dann Add (Hinzufügen) aus.
12. Wählen Sie Richtlinie prüfen.
13. Geben Sie auf der Seite Richtlinie überprüfen einen Namen für die `PassRole`-Richtlinie an und wählen Sie dann Richtlinie erstellen aus.

Beispiele für Amazon SNS-Benachrichtigungen für AWS Systems Manager

Sie können Amazon Simple Notification Service (Amazon SNS) zum Senden von Benachrichtigungen über den Status der Befehle konfigurieren, die Sie mithilfe von Run Command oder Maintenance Windows, Tools in AWS Systems Manager, senden.

> **Note**
>
> Diese Anleitung enthält keine Beschreibungen darüber, wie Benachrichtigungen für Run Command oder Maintenance Windows konfiguriert werden. Weitere Informationen zum Konfigurieren von Run Command oder Maintenance Windows zum Senden von Amazon SNS-Benachrichtigungen über den Status von Befehlen finden Sie unter Konfigurieren von Amazon SNS-Benachrichtigungen für AWS Systems Manager.

Die folgenden Beispiele zeigen die Struktur der JSON-Ausgabe, die von Amazon SNS-Benachrichtigungen zurückgeben wird, wenn eine Konfiguration für Run Command oder Maintenance Windows durchgeführt wurde.

Beispiel einer JSON-Ausgabe für zusammenfassende Nachrichten zu Befehlen mithilfe von Instance-ID-Targeting

```
{
    "commandId": "a8c7e76f-15f1-4c33-9052-0123456789ab",
    "documentName": "AWS-RunPowerShellScript",
    "instanceIds": [
        "i-1234567890abcdef0",
        "i-9876543210abcdef0"
    ],
    "requestedDateTime": "2019-04-25T17:57:09.17Z",
    "expiresAfter": "2019-04-25T19:07:09.17Z",
    "outputS3BucketName": "amzn-s3-demo-bucket",
    "outputS3KeyPrefix": "runcommand",
    "status": "InProgress",
    "eventTime": "2019-04-25T17:57:09.236Z"
}
```

Beispiel einer JSON-Ausgabe für zusammenfassende Nachrichten zu Befehlen mithilfe von Tag-basiertem Targeting

```
{
    "commandId": "9e92c686-ddc7-4827-b040-0123456789ab",
    "documentName": "AWS-RunPowerShellScript",
    "instanceIds": [],
    "requestedDateTime": "2019-04-25T18:01:03.888Z",
    "expiresAfter": "2019-04-25T19:11:03.888Z",
```

```
    "outputS3BucketName": "",
    "outputS3KeyPrefix": "",
    "status": "InProgress",
    "eventTime": "2019-04-25T18:01:05.825Z"
}
```

Beispiel einer JSON-Ausgabe für Nachrichten zu Aufrufen

```
{
    "commandId": "ceb96b84-16aa-4540-91e3-925a9a278b8c",
    "documentName": "AWS-RunPowerShellScript",
    "instanceId": "i-1234567890abcdef0",
    "requestedDateTime": "2019-04-25T18:06:05.032Z",
    "status": "InProgress",
    "eventTime": "2019-04-25T18:06:05.099Z"
}
```

Verwenden von Run Command zum Senden eines Befehls, der Statusbenachrichtigungen zurückgibt

Die folgenden Verfahren zeigen, wie Sie mithilfe der AWS Command Line Interface (AWS CLI) oder AWS Systems Manager -Konsole einen Befehl über ein Tool sendenRun Command, das für die Rückgabe von Statusbenachrichtigungen konfiguriert ist. AWS Systems Manager

Senden eines Run Command, der Benachrichtigungen zurückgibt (Konsole)

Mit der folgenden Vorgehensweise können Sie mittels Run Command einen Befehl senden, der so konfiguriert ist, dass Statusbenachrichtigungen mithilfe der Systems Manager-Konsole zurückgeben werden.

So senden Sie einen Befehl, der Benachrichtigungen zurückgibt (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Befehl ausführen aus.
4. Wählen Sie in der Liste Befehlsdokument ein Systems Manager-Dokument.
5. Geben Sie im Abschnitt Befehlsparameter Werte für erforderliche Parameter an.

6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

   > **Tip**
   >
   > Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Weitere Parameter:

   - Geben Sie im Feld Kommentar Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (Sekunden) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

8. Für Ratenregelung:

   - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, sind diejenigen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle (hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, nicht die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

10. Wählen Sie im Abschnitt SNS Notifications (SNS-Benachrichtigungen)die Option Enable SNS notifications (SNS-Benachrichtigungen aktivieren) aus.
11. Wählen Sie für IAM role (IAM-Rolle) den ARN Amazon SNS IAM-Rolle aus, den Sie in Aufgabe 3 in Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen erstellt haben.
12. Geben Sie für SNS-Thema den ARN für das Amazon SNS-Thema an, welcher verwendet werden soll.
13. Wählen Sie für Event notifications (Ereignisbenachrichtigungen) die Ereignisse aus, für die Sie Benachrichtigungen erhalten möchten.
14. Wählen Sie für Change notifications (Änderungsbenachrichtigen), ob Sie Benachrichtigungen nur für die Befehlsübersicht erhalten möchten (Command status changes (Befehls-Statusänderungen)) oder für jede Kopie eines Befehls, der an mehrere Knoten (Command status on each instance changes (Befehls-Statusänderungen bei jeder Instance)) gesendet wurde.
15. Klicken Sie auf Ausführen.
16. Überprüfen Sie, ob Sie eine Nachricht von Amazon SNS erhalten haben, und öffnen Sie die E-Mail-Nachricht. Es kann einige Minuten dauern, bis Amazon SNS die E-Mail-Nachricht sendet.

Senden eines Run Command, der Benachrichtigungen zurückgibt (CLI)

Gehen Sie wie folgt vor, um mittels Run Command einen Befehl zu senden, der so konfiguriert ist, dass er mithilfe von AWS CLI Statusbenachrichtigungen zurückgibt.

So senden Sie einen Befehl, der Statusbenachrichtigungen zurückgibt (CLI)

1. Öffnen AWS CLI Sie die.
2. Geben Sie im folgenden Befehl die Parameter an, die auf dem verwalteten Knoten basieren sollen IDs.

```
aws ssm send-command --instance-ids "ID-1, ID-2" --document-name "Name"
 --parameters '{"commands":["input"]}' --service-role "SNSRoleARN" --
notification-config '{"NotificationArn":"SNSTopicName","NotificationEvents":
["All"],"NotificationType":"Command"}'
```

Im Folgenden sehen Sie ein Beispiel.

```
aws ssm send-command --instance-ids "i-02573cafcfEXAMPLE, i-0471e04240EXAMPLE"
 --document-name "AWS-RunPowerShellScript" --parameters '{"commands":
["Get-Process"]}' --service-role "arn:aws:iam::111122223333:role/
SNS_Role" --notification-config '{"NotificationArn":"arn:aws:sns:us-
east-1:111122223333:SNSTopic","NotificationEvents":
["All"],"NotificationType":"Command"}'
```

Alternative Befehle

Geben Sie im folgenden Befehl Parameter ab, um auf verwaltete Instances abzuzielen, die Tags verwenden.

```
aws ssm send-command --targets "Key=tag:TagName,Values=TagKey" --document-name
 "Name" --parameters '{"commands":["input"]}' --service-role "SNSRoleARN" --
notification-config '{"NotificationArn":"SNSTopicName","NotificationEvents":
["All"],"NotificationType":"Command"}'
```

Im Folgenden sehen Sie ein Beispiel.

```
aws ssm send-command --targets "Key=tag:Environment,Values=Dev" --
document-name "AWS-RunPowerShellScript" --parameters '{"commands":
["Get-Process"]}' --service-role "arn:aws:iam::111122223333:role/
SNS_Role" --notification-config '{"NotificationArn":"arn:aws:sns:us-
east-1:111122223333:SNSTopic","NotificationEvents":
["All"],"NotificationType":"Command"}'
```

3. Drücken Sie die Eingabetaste.

4. Überprüfen Sie, ob Sie eine Nachricht von Amazon SNS erhalten haben, und öffnen Sie die E-Mail-Nachricht. Es kann einige Minuten dauern, bis Amazon SNS die E-Mail-Nachricht sendet.

Weitere Informationen finden Sie unter send-command in der Referenz zum AWS CLI -Befehl.

Verwenden eines Wartungsfensters zum Senden eines Befehls, der Statusbenachrichtigungen zurückgibt

Die folgenden Verfahren zeigen, wie Sie mithilfe der AWS Systems Manager Konsole oder der AWS Command Line Interface (AWS CLI) eine Run Command Aufgabe in Ihrem Wartungsfenster registrieren. Run Commandist ein Tool in AWS Systems Manager. Zudem wird erläutert, wie die Run Command-Aufgabe so konfiguriert wird, dass Statusbenachrichtigungen zurückgegeben werden.

Bevor Sie beginnen

Wenn Sie kein Wartungsfenster erstellt oder Ziele registriert haben, informieren Sie sich unter Wartungsfenster mit der Konsole erstellen und verwalten, wie ein Wartungsfenster erstellt wird und Ziele registriert werden.

Zum Abrufen von Benachrichtigungen über den Amazon Simple Notification Service (Amazon SNS)-Service müssen Sie eine `iam:PassRole`-Richtlinie an die Maintenance Windows-Servicerolle anhängen, die in der registrierten Aufgabe angegeben ist. Wenn Sie Ihrer Maintenance Windows-Servicerolle keine `iam:PassRole`-Berechtigungen hinzugefügt haben, finden Sie weitere Informationen unter Aufgabe 5: Hängen Sie die iam: PassRole -Richtlinie an Ihre Rolle im Wartungsfenster an.

Registrieren einer Run Command-Aufgabe bei einem Wartungsfenster, die Benachrichtigungen zurückgibt (Konsole)

Gehen Sie wie folgt vor, um eine Run Command-Aufgabe zu registrieren, die so konfiguriert ist, dass mithilfe der Systems Manager-Konsole Statusbenachrichtigungen an Ihr Wartungsfenster zurückgegeben werden.

So registrieren Sie eine Run Command-Aufgabe mit Ihrem Wartungsfenster, die Benachrichtigungen zurückgibt (Konsole)

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.

2. Wählen Sie im Navigationsbereich Maintenance Windows aus.
3. Wählen Sie das Wartungsfenster aus, für das Sie eine Run Command-Aufgabe registrieren möchten, mit der Amazon Simple Notification Service (Amazon SNS)-Benachrichtigungen gesendet werden sollen.
4. Wählen Sie Actions (Aktionen) und anschließend Register Run command task (Run command-Aufgabe registrieren) aus.
5. (Optional) Geben Sie im Feld Name einen Namen für die Aufgabe ein.
6. (Optional) Geben Sie in das Feld Description eine Beschreibung ein.
7. Wählen Sie für Command document (Befehlsdokument) ein Befehlsdokument aus.
8. Geben Sie für Task priority (Aufgabenpriorität) eine Priorität für diese Aufgabe an. Null (`0`) ist die höchste Priorität. Aufgaben in einem Wartungsfenster werden in der Reihenfolge ihrer Priorität geplant. Aufgaben mit derselben Priorität werden parallel geplant.
9. Wählen Sie im Abschnitt Targets (Ziele) eine registrierte Zielgruppe aus, oder wählen Sie nicht registrierte Ziele aus.
10. Für Ratenregelung:

    - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

      > **Note**
      >
      > Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

    - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

11. Wählen Sie im Bereich IAM service role (Servicerolle) die Maintenance Windows-Servicerolle aus, die `iam:PassRole`-Berechtigungen für die SNS-Rolle hat.

> **Note**
>
> Fügen Sie der Maintenance Windows-Rolle `iam:PassRole`-Berechtigungen hinzu, damit Systems Manager die SNS-Rolle an Amazon SNS weitergeben kann. Wenn Sie keine `iam:PassRole`-Berechtigungen hinzugefügt haben, finden Sie weitere Informationen unter Aufgabe 5 im Thema Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

12. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben der Ausgabe in S3 aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

> **Note**
>
> Die S3-Berechtigungen zum Schreiben von Daten in einen S3-Bucket sind die Berechtigungen des dem verwalteten Knoten zugewiesenen Instance-Profils und nicht diejenigen des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

13. Führen Sie im Abschnitt SNS notifications (Benachrichtigungen) folgende Schritte aus:

    - Wählen Sie Enable SNS Notifications (Aktivieren von SNS-Benachrichtigungen).
    - Wählen Sie als IAM role (IAM-Rolle) den Amazon SNS IAM-Rolle Amazon-Ressourcennamen (ARN), den Sie in Aufgabe 3 in Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen erstellt haben, um Amazon SNS zu initiieren.
    - Geben Sie für SNS-Thema den ARN für das Amazon SNS-Thema an, welcher verwendet werden soll.
    - Wählen Sie unter Event type (Ereignistyp) die Ereignisse aus, für die Sie Benachrichtigungen erhalten möchten.
    - Wählen Sie für das Feld Notification type (Benachrichtigungstyp) aus, dass Sie Benachrichtigen für jede Kopie eines Befehls erhalten, der an mehrere Knoten (Aufrufe) gesendet wird, oder die Befehls-Zusammenfassung erhalten.

14. Geben Sie im Abschnitt Parameters (Parameter) die erforderlichen Parameter anhand des ausgewählten Befehlsdokuments ein.
15. Wählen Sie Register Run command task.
16. Sehen Sie nach der nächsten Ausführung Ihres Wartungsfensters in Ihrem Posteingang nach, ob Sie eine Nachricht von Amazon SNS erhalten haben und öffnen Sie die E-Mail-Nachricht. Es kann einige Minuten dauern, bis Amazon SNS die E-Mail-Nachricht sendet.

Registrieren einer Run Command-Aufgabe bei einem Wartungsfenster, die Benachrichtigungen zurückgibt (CLI)

Gehen Sie wie folgt vor, um eine Run Command-Aufgabe zu registrieren, die so konfiguriert ist, dass mithilfe der AWS CLI Statusbenachrichtigungen an Ihr Wartungsfenster zurückgegeben werden.

So registrieren Sie eine Run Command Aufgabe mit Ihrem Wartungsfenster, die Benachrichtigungen zurückgibt (CLI)

> **Note**
>
> Zur besseren Koordination der Aufgabenoptionen wird bei dieser Vorgehensweise die Befehlsoption `--cli-input-json` verwendet. Dabei sind die Optionswerte in einer JSON-Datei gespeichert.

1. Erstellen Sie auf dem lokalen Computer eine Datei mit dem Namen `RunCommandTask.json`.
2. Fügen Sie den folgenden Inhalt in die -Datei ein:

```
{
    "Name": "Name",
    "Description": "Description",
    "WindowId": "mw-0c50858d01EXAMPLE",
    "ServiceRoleArn": "arn:aws:iam::account-id:role/MaintenanceWindowIAMRole",
    "MaxConcurrency": "1",
    "MaxErrors": "1",
    "Priority": 3,
    "Targets": [
        {
            "Key": "WindowTargetIds",
            "Values": [
                "e32eecb2-646c-4f4b-8ed1-205fbEXAMPLE"
```

```
            ]
        }
    ],
    "TaskType": "RUN_COMMAND",
    "TaskArn": "CommandDocumentName",
    "TaskInvocationParameters": {
        "RunCommand": {
            "Comment": "Comment",
            "TimeoutSeconds": 3600,
            "NotificationConfig": {
                "NotificationArn": "arn:aws:sns:region:account-id:SNSTopicName",
                "NotificationEvents": [
                    "All"
                ],
                "NotificationType": "Command"
            },
            "ServiceRoleArn": "arn:aws:iam::account-id:role/SNSIAMRole"
        }
    }
}
```

3. Ersetzen Sie die Beispielwerte mit Informationen über Ihre eigenen Ressourcen.

   Sie können auch Optionen wiederherstellen, die bei diesem Beispiel ausgelassen wurden, sofern Sie diese verwenden möchten. So kann die Befehlsausgabe in einem S3-Bucket gespeichert werden.

   Weitere Informationen finden Sie unter register-task-with-maintenance-window in der Referenz zum AWS CLI -Befehl.

4. Speichern Sie die Datei.

5. Führen Sie auf dem lokalen Computer in dem Verzeichnis, in dem die Datei gespeichert wurde, folgenden Befehl aus.

   ```
   aws ssm register-task-with-maintenance-window --cli-input-json file://
   RunCommandTask.json
   ```

   > **Important**
   >
   > Achten Sie darauf, dass `file://` vor dem Dateinamen steht. Dies ist bei diesem Befehl erforderlich.

Wenn der Befehl erfolgreich ausgeführt wurde, sieht das Ergebnis in etwa wie folgt aus.

```
{
    "WindowTaskId": "j2l8d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE"
}
```

6. Sehen Sie nach der nächsten Ausführung Ihres Wartungsfensters in Ihrem Posteingang nach, ob Sie eine Nachricht von Amazon SNS erhalten haben und öffnen Sie die E-Mail-Nachricht. Es kann einige Minuten dauern, bis Amazon SNS die E-Mail-Nachricht sendet.

Weitere Informationen zum Registrieren von Aufgaben für ein Wartungsfenster in der Befehlszeile finden Sie unter Register tasks with the maintenance window (Registrieren von Aufgaben im Wartungsfenster).

Produkt- und Service-Integrationen mit Systems Manager

AWS Systems Manager ist standardmäßig mit AWS-Services sowie andere Produkte und Services integriert. Die folgenden Informationen können Ihnen die Konfiguration von Systems Manager zum Integrieren in die von Ihnen verwendeten Produkte und Services erleichtern.

- Integration mit AWS-Services
- Integration in andere Produkte und Services

Integration mit AWS-Services

Durch die Verwendung von Systems Manager Manager-Befehlsdokumenten (SSM-Dokumenten) und Automation-Runbooks können Sie die Integration AWS Systems Manager mit verwenden. AWS-Services Weitere Informationen zu diesen Ressourcen finden Sie unter AWS Systems Manager-Documents.

Systems Manager ist in Folgendes integriert AWS-Services.

Datenverarbeitung

Amazon Elastic Compute Cloud (Amazon EC2)	Amazon EC2 bietet skalierbare Rechenkapazität in der AWS Cloud. EC2 Durch die Nutzung von Amazon müssen Sie nicht im Voraus in Hardware investieren, sodass Sie Anwendungen schneller entwickeln und bereitstellen können. Sie können Amazon verwenden EC2 , um so viele oder so wenige virtuelle Server zu starten, wie Sie benötigen, Sicherheit und Netzwerke zu konfigurieren und Speicher zu verwalten. Mit Systems Manager können Sie mehrere Aufgaben auf EC2 Instanzen ausführen. Sie können beispielsweise Ihre EC2 Instances starten, konfigurieren, verwalten, warten, Fehler beheben und eine sichere Verbindung zu ihnen

	herstellen. Sie können Systems Manager auch verwenden, um Software bereitzustellen, den Compliance-Status zu ermitteln und Inventar Ihrer EC2 Instanzen zu sammeln. Weitere Informationen • Arbeiten mit verwalteten Knoten • AWS Systems Manager State Manager • AWS Systems Manager Run Command • AWS Systems Manager Patch Manager • AWS Systems Manager Session Manager • AWS Systems Manager Distributor • AWS Systems Manager-Compliance • AWS Systems Manager-Bestand
Amazon EC2 Auto Scaling	Mit Auto Scaling können Sie sicherstellen, dass Ihnen die richtige Anzahl von EC2 Instances zur Verfügung steht, um die Last für Ihre Anwendung zu bewältigen. Sie erstellen Sammlungen von EC2 Instances, die als Auto Scaling Scaling-Gruppen bezeichnet werden. Systems Manager ermöglicht es Ihnen, gängige Prozeduren wie das Patchen des Amazon Machine Image (AMI) zu automatisieren, die in Ihrer Auto Scaling-Vorlage für Ihre Auto-Scal ing-Gruppe verwendet wird. Weitere Informationen Aktualisieren von AMIs für Auto-Scaling-Grupp en

Amazon Elastic Container Service (Amazon ECS)	Amazon ECS ist ein hoch skalierbarer, schneller Container-Management-Service, der das Ausführen, Beenden und Verwalten von Docker-Containern in einem Cluster ermöglicht. Mit Systems Manager können Sie Container-Instance remote verwalten und sensible Daten in Ihre Container einspeisen, indem Sie Ihre sensiblen Daten in Parametern in Parameter Store speichern, ein Tool in Systems Manager, und dann in Ihrer Container-Definition darauf verweisen. Weitere Informationen • Remote-Verwaltung von Container-Instances mit AWS Systems Manager • Angeben sensibler Daten mithilfe des Systems Manager-Parameter Parameter Store

AWS Lambda	Lambda ist ein Datenverarbeitungsservice, mit dem Sie Code ausführen können, ohne Server bereitstellen oder verwalten zu müssen. Lambda führt Ihren Code nur bei Bedarf aus und skaliert automatisch – von einigen Anforderungen pro Tag bis zu Tausenden pro Sekunde. Systems Manager ermöglicht es Ihnen, Lambda-Funktionen in Automation-Runbook-Inhalten mithilfe der `aws:invokeLambdaFunction` -Aktion zu verwenden. Um Parameter aus Parameter Store AWS Lambda Funktionen zu verwenden, können Sie die Lambda-Erweiterung Parameters and Secrets verwenden, um Parameterwerte abzurufen und sie für die future Verwendung AWS zwischenzuspeichern. Weitere Informationen Aktualisieren Sie ein Golden AMI mithilfe von Automation, AWS Lambda, und Parameter Store Verwenden von Parameter Store-Parametern in AWS Lambda -Funktionen

Internet of Things (IoT)

AWS IoT Greengrass Kerngeräte	AWS IoT Greengrass ist ein Open-Source IoT-Edge-Laufzeit- und Cloud-Service, mit dem Sie IoT-Anwendungen auf Ihren Geräten entwickeln, bereitstellen und verwalten können. Systems Manager bietet native Unterstützung für AWS IoT Greengrass Kerngeräte.

	Weitere Informationen Verwalten von Edge-Geräten mit Systems Manager
AWS IoT Kerngeräte	AWS IoTstellt die Cloud-Dienste bereit, die Ihre IoT-Geräte mit anderen Geräten und AWS Cloud-Diensten verbinden. AWS IoT biete t Gerätesoftware, mit der Sie Ihre IoT-Gerät e in AWS IoT basierte Lösungen integrieren können. Wenn Ihre Geräte eine Verbindung herstellen können AWS IoT, AWS IoT können Sie sie mit den bereitgestellten Cloud-Diensten verbinden. AWS Systems Manager unterstützt AWS IoT Kerngeräte, sofern diese Geräte als verwaltete Knoten in einer Hybrid- und Multi-Cloud-Umgebung konfiguriert sind. Weitere Informationen Verwalten von Knoten in Hybrid- und Multi-Cloud-Umgebungen mit Systems Manager

Speicher

Amazon Simple Storage Service (Amazon-S3)	Bei Amazon S3 handelt es sich um Speicher für das Internet. Der Service ist darauf ausgelegt, Cloud Computing für Entwickler zu erleichtern. Amazon S3 besitzt eine einfache Web-Service-Schnittstelle zum Speichern und Abrufen einer beliebigen Datenmenge zu jeder Zeit und von jedem Ort im Internet aus. Mit Systems Manager können Sie Remote-Sk ripts und SSM-Dokumente ausführen, die in Amazon S3 gespeichert sind. Distributor, ein Tool in AWS Systems Manager, verwendet

	Amazon S3 zum Speichern von Paketen. Sie können auch eine Ausgabe an Amazon S3 für Run Command und Session Manager senden, Tools in AWS Systems Manager. Weitere Informationen • Ausführen von Skripts von Amazon S3 • Ausführen von -Dokumenten von Remote-Standorten • AWS Systems Manager Distributor • Protokollieren von Sitzungsdaten mithilfe von Amazon S3 (Konsole)

Entwicklertools

AWS CodeBuild	CodeBuildist ein vollständig verwalteter Build-Service in der Cloud. CodeBuild kompiliert Ihren Quellcode, führt Komponententests durch und erzeugt Artefakte, die sofort einsatzbereit sind. CodeBuild macht die Bereitstellung, Verwaltung und Skalierung Ihrer eigenen Build-Server überflüssig. Mit Parameter Store können Sie vertrauliche Informationen für Ihre Build-Spezifikationen und Projekte speichern. Weitere Informationen • Referenz zur Build-Spezifikation für CodeBuild • Erstellen Sie ein Build-Projekt in AWS CodeBuild

AWS CDK	Das AWS Cloud Development Kit (AWS CDK) ist ein Framework für die Definition der Cloud-Infrastruktur als Code mit Programmiersprachen und deren Bereitstellung AWS CloudFormation. Application Manager ermöglicht es Ihnen, Ihre CDK-Konstrukte als Anwendungen gruppiert zu betrachten, die Anwendungsstruktur einschließlich der zugrundeliegenden Ressourcen einzusehen, Warnungen anzuzeigen, Betriebsprobleme zu untersuchen und zu beheben und Kosten in der Application Manager-Konsole zu verfolgen. Weitere Informationen • Anzeigen von Übersichtsinformationen einer Anwendung • Anzeigen von Anwendungsressourcen

Sicherheit, Identität und Compliance

AWS Identity and Access Management (IAM)	IAM ist ein Webservice, mit dem Sie den Zugriff auf Ressourcen sicher kontrollieren können. AWS Sie verwenden IAM, um zu steuern, wer authentifiziert (angemeldet) und autorisiert (Berechtigungen besitzt) ist, Ressourcen zu nutzen. Systems Manager ermöglicht es Ihnen, den Zugriff auf Dienste mithilfe von IAM zu steuern. Weitere Informationen • Funktionsweise von AWS Systems Manager mit IAM

	• Aktionen, Ressourcen und Bedingungsschlüssel für AWS Systems Manager • Erforderliche Instance-Berechtigungen für Systems Manager konfigurieren
AWS Secrets Manager	Secrets Manager bietet eine einfachere Verwaltung von Secrets. Bei den Secrets kann es sich um Datenbank-Anmeldeinformationen, Passwörter, API-Schlüssel von Drittanbietern und sogar beliebigen Text handeln. Mit Parameter Store können Sie Secrets-Manager-Geheimnisse abrufen, wenn Sie andere AWS-Services verwenden, die bereits Referenzen zu Parameter Store-Parametern unterstützen. Weitere Informationen Referenzieren von AWS Secrets Manager-Geheimnissen über Parameter Store-Parameter

AWS Security Hub CSPM	Security Hub CSPM bietet Ihnen einen umfassenden Überblick über Ihre Sicherheitswarnungen mit hoher Priorität und den Compliance-Status in allen Bereichen. AWS-Konten Security Hub CSPM aggregiert, organisiert und priorisiert Ihre Sicherheitswarnungen oder Ergebnisse aus mehreren. AWS-Services Wenn Sie die Integration zwischen Security Hub CSPM und Security Hub CSPMPatch Manager, einem Tool in AWS Systems Manager, aktivieren, überwacht Security Hub CSPM den Patching-Status Ihrer Flotten aus Sicherheitsgründen. Details zur Patch-Konformität werden automatisch in Security Hub CSPM exportiert. Auf diese Weise können Sie mit einer einzigen Ansicht den Patch-Compliance-Status zentral überwachen und andere Sicherheitsergebnisse nachverfolgen. Sie können Benachrichtigungen erhalten, wenn Knoten in Ihrer Flotte die Patch-Konformität nicht mehr einhalten, und die Ergebnisse der Patch-Compliance in der Security Hub CSPM-Konsole überprüfen. Sie können Security Hub CSPM auch mit Explorer undOpsCenter, Tools in integrieren. AWS Systems Manager Die Integration mit Security Hub CSPM ermöglicht es Ihnen, Ergebnisse von Security Hub CSPM in und zu erhalten. Explorer OpsCenter Die CSPM-Ergebnisse von Security Hub bieten Sicherheitsinformationen, die Sie verwenden können, OpsCenter um Ihre Sicherheits Explorer-, Leistungs- und Betriebsprobleme in zu

	aggregieren und Maßnahmen zu ergreifen. AWS Systems Manager Die Nutzung von Security Hub CSPM ist kostenpflichtig. Weitere Informationen finden Sie unter Security Hub CSPM — Preise. Weitere Informationen • Empfangen von Ergebnissen von AWS Security Hub CSPM in Explorer • Verständnis der OpsCenter Integration mit AWS Security Hub CSPM • Integration von Patch Manager in AWS Security Hub CSPM

Kryptografie und PKI

AWS Key Management Service (AWS KMS)	AWS KMS ist ein verwalteter Service, der es Ihnen ermöglicht, kundenverwaltete Schlüssel zu erstellen und zu kontrollieren, d. h. die Verschlüsselungsschlüssel, die zur Verschlüsselung Ihrer Daten verwendet werden. Mit Systems Manager können AWS KMS Sie `SecureString` Parameter erstellen und Session Manager Sitzungsdaten verschlüsseln. Weitere Informationen • AWS KMS Verschlüsselung für AWS Systems ManagerParameter Store SecureString Parameter • So aktivieren Sie die KMS-Schlüsselverschlüsselung von Sitzungsdaten (Konsole)

Verwaltung und Governance

AWS CloudFormation	CloudFormation ist ein Service, der die Entwicklung und Einrichtung von Amazon Web Services-Ressourcen erleichtert, sodass Sie weniger Zeit für die Verwaltung dieser Ressourcen aufwenden müssen und sich stattdessen mehr auf Ihre Anwendungen, die in AWS ausgeführt werden, konzentrieren können. Parameter Store ist eine Quelle für dynamisch e Referenzen. Dynamische Verweise bieten eine kompakte und leistungsstarke Möglichke it, externe Werte anzugeben, die in anderen Diensten in Ihren CloudFormation Stack-Vor lagen gespeichert und verwaltet werden. Weitere Informationen Verwenden von dynamischen Referenzen zum Angeben von Vorlagenwerten
AWS CloudTrail	CloudTrailist ein Programm AWS-Service , das Ihnen dabei hilft, Unternehmensführun g, Compliance sowie Betriebs- und Risikoprü fungen Ihres AWS-Konto Unternehmens zu autorisieren. Von einem Benutzer, einer Rolle oder einem ausgeführte Aktionen AWS-Service werden als Ereignisse in CloudTrail aufgezeichnet. Zu den Ereignissen gehören Aktionen AWS-Managementkonsole, die in den Bereichen, AWS Command Line Interface (AWS CLI) und AWS SDKs und ausgeführt wurden APIs. Systems Manager ist integriert und erfasst CloudTrail die meisten Systems Manager

	Manager-API-Aufrufe als Ereignisse. Dazu gehören API-Aufrufe, die von der Systems Manager-Konsole aus initiiert werden, und Aufrufe an den Systems Manager APIs. Weitere Informationen Protokollierung von AWS Systems Manager-API-Aufrufen mit AWS CloudTrail
CloudWatch Amazon-Protokolle	Mit Amazon CloudWatch Logs können Sie die Protokolle all Ihrer Systeme und Anwendungen, AWS-Services die Sie verwenden, zentralisieren. Sie können sie dann anzeigen, nach bestimmten Fehlercodes oder Mustern suchen, sie anhand bestimmter Felder filtern oder sicher für zukünftige Analysen archivieren. Systems Manager unterstützt das Senden von Protokollen für die SSM AgentRun Command, und Session Manager an CloudWatch Logs. Weitere Informationen • Senden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent) • Konfigurieren von Amazon CloudWatch Logs für Run Command • Protokollierung von Sitzungsdaten mit Amazon CloudWatch Logs (Konsole)

Amazon EventBridge	EventBridgeliefert einen Stream von Systemereignissen nahezu in Echtzeit, der Änderungen an den Ressourcen von Amazon Web Services beschreibt. Mithilfe einfacher Regeln, die Sie schnell einrichten können, können Sie Ereignisse zuordnen und sie an eine oder mehrere Zielfunktionen oder Streams weiterleiten. EventBridge wird sich betrieblicher Änderungen bewusst, sobald sie auftreten. EventBridge reagiert auf diese betrieblichen Änderungen und ergreift gegebenenfalls Korrekturmaßnahmen. Dazu gehören das Senden von Nachrichten zur Reaktion auf die Umgebung, das Aktivieren von Funktionen und das Erfassen von Statusinformationen. Systems Manager verfügt über mehrere Ereignisse, die unterstützt werden, EventBridge sodass Sie auf der Grundlage des Inhalts dieser Ereignisse Maßnahmen ergreifen können. Weitere Informationen Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge **Note** Amazon EventBridge ist die bevorzugte Methode, um Ihre Veranstaltungen zu verwalten. CloudWatch Bei Events und EventBridge handelt es sich um denselben zugrunde liegenden Service und dieselbe API, EventBridge bieten aber mehr Funktionen. Änderungen, die Sie in einer der beiden CloudWatch oder in jeder Konsole vornehmen

	, EventBridge spiegeln sich in jeder Konsole wider. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch.
AWS Config	AWS Configbietet einen detaillierten Überblick über die Konfiguration der AWS Ressourcen in Ihrem AWS-Konto. Dazu gehört auch, wie die Ressourcen jeweils zueinander in Beziehung stehen und wie sie konfiguriert wurden. Auf diese Weise können Sie sehen, wie sich die Konfigurationen und Beziehungen im Laufe der Zeit ändern. Systems Manager ist in integriert AWS Config und bietet mehrere Regeln, die Ihnen helfen, einen Überblick über Ihre EC2 Instanzen zu erhalten. Mithilfe dieser Regeln können Sie feststellen, welche EC2 Instanzen von Systems Manager verwaltet werden, welche Betriebssystemkonfigurationen, Updates auf Systemebene, installierte Anwendungen, Netzwerkkonfigurationen und mehr vorhanden sind. Weitere Informationen • AWS Config unterstützte Ressourcentypen • Aufzeichnen der Software-Konfiguration für verwaltete Instances • Anzeigen von Bestandsverlauf und Änderungsnachverfolgung

AWS Trusted Advisor	Trusted Advisor ist ein Online-Tool, das Sie in Echtzeit dabei unterstützt, Ihre Ressourcen gemäß den bewährten Methoden von AWS bereitzustellen. Systems Manager hostet Trusted Advisor und Sie können Trusted Advisor Daten in anzeigenExplorer. Weitere Informationen • AWS Systems Manager Explorer • Erste Schritte mit AWS Trusted Advisor

AWS Organizations	Organizations ist ein Kontoverwaltungsdienst, mit dem Sie mehrere Konten zu einer Organisation AWS-Konten zusammenfassen können, die Sie erstellen und zentral verwalten. Organisationen umfasst Kontoverwaltungs- und konsolidierte Fakturierung, mit denen Sie die Budget-, Sicherheits- und Compliance-Anforderungen Ihres Unternehmens besser erfüllen können. Die Integration zwischen Change Manager, einem Tool in AWS Systems Manager, und Organizations ermöglicht die Verwendung eines delegierten Administratorkontos, um Änderungsanforderungen, Änderungsvorlagen und Genehmigungen für Ihre gesamte Organisation über dieses eine Konto zu verwalten. Organisationsintegration mit Inventory, einem Tool in AWS Systems Manager, und Explorerermöglicht es Ihnen, Bestands- und Betriebsdaten (OpsData) aus mehreren AWS-Regionen und zu aggregieren AWS-Konten. Die Integration zwischenQuick Setup, einem Tool in AWS Systems Manager und Organizations automatisiert allgemeine Aufgaben zur Einrichtung von Diensten und stellt Serviceko nfigurationen auf der Grundlage von Best Practices in Ihren Organisationseinheiten bereit ()OUs.

Netzwerk und Bereitstellung von Inhalten

AWS PrivateLink	AWS PrivateLinkermöglicht es Ihnen, Ihre Virtual Private Cloud (VPC) privat mit unterstüt

	zten AWS-Services und VPC-Endpunktdienst en zu verbinden, ohne dass ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder Direct Connect eine Verbindung erforderl ich ist. Systems Manager unterstützt verwaltete Knoten, die eine Verbindung zu Systems Manager AWS PrivateLink herstellen, APIs indem Dies verbessert die Sicherheitslage Ihrer verwalteten Knoten, da der gesamte Netzwerkv erkehr zwischen Ihren verwalteten Knoten, Systems Manager und Amazon EC2 auf das Amazon-Netzwerk AWS PrivateLink beschränk t wird. Dies bedeutet, dass verwaltete Knoten keinen Zugriff auf das Internet haben müssen. Weitere Informationen Verbessern Sie die Sicherheit von EC2 Instanzen mithilfe von VPC-Endpunkten für Systems Manager

Analysen

Amazon Athena	Athena ist ein interaktiver Abfragese rvice, der die direkte Analyse von Daten in Amazon Simple Storage Service (Amazon S3) mit Standard-SQL ermöglicht. Mit einigen Aktionen in der AWS-Managementkonsole können Sie Athena auf Ihre in Amazon S3 gespeicherten Daten verweisen und beginnen, Standard-SQL zu verwenden, um einmalige Abfragen auszuführen und innerhalb von Sekunden Ergebnisse zu erhalten.

	Systems Manager Inventory ist in Athena integriert, sodass Sie Inventardaten von mehreren AWS-Regionen und AWS-Konte n abfragen können. Die Athena-Integration verwendet Ressourcen-Datensynchronisierung, sodass Sie Bestandsdaten aus allen verwaltet en Knoten auf der Seite Detailed View (Detailan sicht) in der Systems-Manager-Inventory-K onsole anzeigen können. Weitere Informationen • Abfragen von Bestandsdaten aus mehreren Regionen und Konten • Walkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten
AWS Glue	AWS Glue ist ein vollständig verwalteter ETL-Service (Extrahieren, Transformieren und Laden), mit dessen Hilfe einfach und wirtschaftlich Ihre Daten kategorisiert, bereinigt , erweitert und zwischen verschiedenen Datenspeichern und Datenströmen verschoben werden können. Systems Manager wird verwendet AWS Glue , um die Inventardaten in Ihrem S3-Bucket zu crawlen. Weitere Informationen Abfragen von Bestandsdaten aus mehreren Regionen und Konten

Amazon Quick Suite	Quick Suite ist ein Geschäftsanalysedienst, mit dem Sie Visualisierungen erstellen, einmalige Analysen durchführen und geschäftliche Erkenntnisse aus Ihren Daten gewinnen können. Es kann automatisch AWS -Date nquellen erkennen und arbeitet auch mit Ihren Datenquellen. Die Ressourcen-Datensynchronisierung von Systems Manager sendet die von all Ihren verwalteten Knoten erfassten Bestandsdaten an einen einzigen S3-Bucket. Sie können Quick Suite verwenden, um die aggregierten Daten abzufragen und zu analysieren. Weitere Informationen • Erstellen einer Resource Data Sync für Inventory • Walkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten

Anwendungsintegration

Amazon-Simple-Notification-Service (Amazon-SNS)	Amazon SNS ist ein Webservice, der die Zustellung oder das Senden von Nachricht en an abonnierende Endpunkte oder Clients koordiniert und verwaltet. Systems Manager generiert Status für mehrere Dienste, die von Amazon SNS-Benachrichtigu ngen erfasst werden können.

	Weitere Informationen • Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen • Einrichten von Benachrichtigungen oder Auslösen von Aktionen basierend auf Parameter Store-Ereignissen

AWS-Managementkonsole

AWS -Ressourcengruppen	Resource Groups organisieren Ihre AWS Ressourcen. Ressourcengruppen vereinfachen die gleichzeitige Verwaltung, Überwachung und Automatisierung von Aufgaben für eine große Zahl von Ressourcen. Systems-Manager-Ressourcentypen wie verwaltete Knoten, SSM-Dokumente, Wartungsfenster, Parameter Store-Parameter und Patch-Baselines können Ressourcengruppen hinzugefügt werden. Weitere Informationen Was sind AWS -Ressourcengruppen?

Themen

- Ausführen von Skripts von Amazon S3
- Referenzieren von AWS Secrets Manager-Geheimnissen über Parameter Store-Parameter
- AWS KMS Verschlüsselung für AWS Systems ManagerParameter Store SecureString Parameter
- Verwenden von Parameter Store-Parametern in Amazon Elastic Kubernetes Service
- Verwenden von Parameter Store-Parametern in AWS Lambda -Funktionen

Ausführen von Skripts von Amazon S3

In diesem Abschnitt wird beschrieben, wie Skripts von Amazon Simple Storage Service (Amazon S3) heruntergeladen und ausgeführt werden. Das folgende Thema enthält Informationen und Terminologie zu Amazon S3. Weitere Informationen zu Amazon S3 finden Sie unter Was ist Amazon S3? Sie können verschiedene Arten von Skripten ausführen, darunter Ansible Playbooks, Python, Ruby, Shell und. PowerShell

Sie können auch ein Verzeichnis mit mehreren Skripts herunterladen. Wenn Sie das primäre Skript im Verzeichnis ausführen, werden AWS Systems Manager auch alle referenzierten Skripten ausgeführt, die im Verzeichnis enthalten sind.

Beachten Sie die folgenden wichtigen Hinweise zum Ausführen von Skripts von Amazon S3:

- Systems Manager prüft nicht, ob Ihr Skript auf einem Knoten ausgeführt werden kann. Stellen Sie sicher, dass die erforderliche Software auf dem Knoten installiert ist, bevor Sie das Skript herunterladen und ausführen. Sie können auch ein zusammengesetztes Dokument erstellen, das die Software mithilfe von Run Command oderState Manager, Tools in AWS Systems Manager installiert und anschließend das Skript herunterlädt und ausführt.
- Stellen Sie sicher, dass Ihrem Benutzer, Ihrer Rolle oder Gruppe die AWS Identity and Access Management (IAM)-Berechtigungen gewährt wurden, die zum Lesen aus dem S3-Bucket erforderlich sind.
- Stellen Sie sicher, dass das Instance-Profil auf Ihren Amazon Elastic Compute Cloud (Amazon EC2) -Instances über `s3:GetObject` Berechtigungen verfügt`s3:ListBucket`. Wenn das Instance-Profil nicht über diese Berechtigungen verfügt, kann das System Ihr Skript nicht aus dem S3-Bucket herunterladen. Weitere Informationen finden Sie unter Verwenden von Instance-Profilen im IAM-Benutzerhandbuch.

Ausführen von Shell-Scripts von Amazon S3

Die folgenden Informationen enthalten Verfahren, die Ihnen helfen, Skripts von Amazon Simple Storage Service (Amazon S3) entweder über die AWS Systems Manager Konsole oder die AWS Command Line Interface (AWS CLI) auszuführen. Obwohl Shell-Skripte in den Beispielen verwendet werden, können andere Arten von Skripten ersetzt werden.

Ausführen eines Shell-Skripts von Amazon S3 (Konsole)

Ausführen eines Shell-Skripts von Amazon S3

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Befehl ausführen aus.
4. Wählen Sie in der Liste Command document (Befehlsdokument) die Option **AWS-RunRemoteScript** aus.
5. Führen Sie unter Command parameters die folgenden Schritte aus:

   - Wählen Sie unter Source Type die Option S3 aus.
   - Geben Sie im Textfeld Source Info die für den Zugriff auf die Quelle erforderlichen Informationen im folgenden Format an. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

     > **Note**
     >
     > Ersetzen Sie https://s3. *aws-api-domain*mit der URL für Ihren Bucket. Sie können Ihre Bucket-URL in Amazon S3 auf der Registerkarte Objects (Objekte) kopieren.

     ```
     {"path":"https://s3.aws-api-domain/path to script"}
     ```

     Im Folgenden wird ein -Beispiel gezeigt.

     ```
     {"path":"https://amzn-s3-demo-bucket.s3.us-east-2.amazonaws.com/scripts/shell/
     helloWorld.sh"}
     ```

   - Geben Sie im Feld Command Line Parameter für die Skriptausführung ein. Ein Beispiel.

     ```
     helloWorld.sh argument-1 argument-2
     ```

   - (Optional) Geben Sie im Feld Working Directory (Arbeitsverzeichnis) den Namen eines Verzeichnisses auf dem Knoten ein, auf dem das Skript heruntergeladen und ausgeführt werden soll.

- (Optional) Geben Sie unter Execution Timeout die Dauer in Sekunden an, bis das System die Skriptbefehlausführung fehlschlagen lässt.

6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

> **Tip**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Weitere Parameter:

   - Geben Sie im Feld Kommentar Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (Sekunden) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

8. Für Ratenregelung:

   - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

   > **Note**
   >
   > Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, sind diejenigen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle (hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, nicht die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

10. Aktivieren Sie das Kontrollkästchen SNS-Benachrichtigungen aktivieren im Abschnitt SNS-Benachrichtigungen, wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon-SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

11. Klicken Sie auf Ausführen.

Ausführen eines Shell-Skripts von Amazon S3 (Befehlszeile)

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

> **Note**
>
> Ersetzen Sie https://s3. *aws-api-domain*/*script path*mit der URL für Ihren Bucket. Sie können Ihre Bucket-URL in Amazon S3 auf der Registerkarte Objects (Objekte) kopieren.

Linux & macOS

```
aws ssm send-command \
    --document-name "AWS-RunRemoteScript" \
    --output-s3-bucket-name "amzn-s3-demo-bucket" \
    --output-s3-key-prefix "key-prefix" \
    --targets "Key=InstanceIds,Values=instance-id" \
    --parameters '{"sourceType":["S3"],"sourceInfo":["{\"path\":\"https://
s3.aws-api-domain/script path\"}"],"commandLine":["script name and arguments"]}'
```

Windows

```
aws ssm send-command ^
    --document-name "AWS-RunRemoteScript" ^
    --output-s3-bucket-name "amzn-s3-demo-bucket" ^
    --output-s3-key-prefix "key-prefix" ^
    --targets "Key=InstanceIds,Values=instance-id" ^
    --parameters "sourceType"="S3",sourceInfo='{\"path\":\"https://s3.aws-api-
domain/script path\"}',"commandLine"="script name and arguments"
```

PowerShell

```
Send-SSMCommand `
    -DocumentName "AWS-RunRemoteScript" `
    -OutputS3BucketName "amzn-s3-demo-bucket" `
    -OutputS3KeyPrefix "key-prefix" `
    -Target @{Key="InstanceIds";Values=@("instance-id")} `
    -Parameter @{
        sourceType = "S3";
        sourceInfo = '{"path": "s3://bucket-name/path/to/script"}';
        commandLine = "script name and arguments"
    }
```

Referenzieren von AWS Secrets Manager-Geheimnissen über Parameter Store-Parameter

AWS Secrets Manager hilft Ihnen, wichtige Konfigurationsdaten wie Anmeldeinformationen, Passwörter und Lizenzschlüssel zu organisieren und zu verwalten. Parameter Store, ein Tool in AWS Systems Manager, ist jetzt in Secrets Manager integriert, sodass Sie Secrets-Manager-Geheimnisse abrufen können, wenn Sie andere AWS-Services verwenden, die bereits Verweise auf Parameter Store-Parameter unterstützen. Zu diesen Services gehören Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Container Service (Amazon ECS), AWS Lambda, CloudFormation, AWS CodeBuild, AWS CodeDeploy und andere Systems-Manager-Tools. Indem Sie Parameter Store zum Verweisen auf Secrets Manager-Geheimnisse verwenden, erstellen Sie einen konsistenten und sicheren Prozess zum Aufrufen und Verwenden von Geheimnissen und zum Referenzieren von Daten in Ihrem Code und den Konfigurationsskripts.

Weitere Informationen zu Secrets Manager finden Sie unter Was ist AWS Secrets Manager? im AWS Secrets Manager-Benutzerhandbuch.

Einschränkungen

Beachten Sie die folgenden Einschränkungen bei der Verwendung von Parameter Store zur Referenzierung von Secrets Manager-Geheimnissen:

- Sie können Secrets Manager-Geheimnisse nur mit den API-Aktionen GetParameter und GetParameters abrufen. Änderungsoperationen und frühzeitige Abfragen von API-Operationen, wie z. B. DescribeParameters oder GetParametersByPath werden für Secrets Manager nicht unterstützt.
- Sie können die AWS Command Line Interface (AWS CLI), AWS Tools for Windows PowerShell und die SDKs zum Abrufen eines Geheimnisses mit Hilfe von Parameter Store verwenden.
- Secrets Manager Secrets in Parameter Store müssen das Präfix `/aws/reference/secretsmanager` haben. Im Folgenden sind einige Beispiele aufgeführt:

  ```
  /aws/reference/secretsmanager/CFCreds1
  ```

  ```
  /aws/reference/secretsmanager/myapp/db/password
  ```

- Parameter Store berücksichtigt AWS Identity and Access Management (IAM)-Richtlinien, die Secrets Manager-Geheimnissen zugeordnet sind. Beispiel: Wenn Benutzer 1 keinen Zugriff auf Geheimnis A hat, kann er das Geheimnis nicht mithilfe von Parameter Store abrufen.

- Parameter, die Secrets Manager-Geheimnisse referenzieren, können die Parameter Store-Versioning- oder Verlaufsfunktionen nicht verwenden.
- Parameter Store berücksichtigt Secrets Manager Versionsstufen. Wenn Sie eine Versionsstufe referenzieren, verwendet diese Buchstaben, Zahlen, einen Punkt (.), Bindestrich (-) oder Unterstrich (_). Alle anderen Symbole, die in der Versionsstufe angegeben sind, führen dazu, dass die Referenz fehlschlägt.

Referenzieren eines Secrets Manager-Geheimnisses mit Parameter Store

Im folgenden Verfahren wird beschrieben, wie Sie ein Secrets Manager-Geheimnis mithilfe von Parameter Store-APIs referenzieren. Das Verfahren referenziert weitere Verfahren im AWS Secrets Manager-Benutzerhandbuch.

> **Note**
>
> Bevor Sie beginnen, stellen Sie sicher, dass Sie über die Berechtigung zum Referenzieren von Secrets Manager-Geheimnissen in Parameter Store-Parametern verfügen. Wenn Sie über Administratorrechte in Secrets Manager und Systems Manager verfügen, können Sie Geheimnisse mithilfe von Parameter Store-APIs. Wenn Sie ein Secrets Manager-Geheimnis in einem Parameter Store-Parameter referenzieren und nicht über die Berechtigung für den Zugriff auf dieses Geheimnis verfügen, schlägt die Referenz fehl. Weitere Informationen finden Sie unter Authentifizierung und Zugriffskontrolle für AWS Secrets Manager im AWS Secrets Manager-Benutzerhandbuch.

> **Important**
>
> Parameter Store funktioniert als Pass-Through-Service für Verweise auf Secrets Manager-Geheimnisse. Parameter Store bewahrt keine Daten oder Metadaten über Geheimnisse auf. Die Referenz ist zustandslos.

Referenzieren eines Secrets Manager-Geheimnisses mit Parameter Store

1. Erstellen Sie ein Geheimnis in Secrets Manager. Weitere Informationen finden Sie unter Erstellen und Verwalten von Secrets mit AWS Secrets Manager.

2. Referenzieren Sie ein Geheimnis über die AWS CLI, mit AWS Tools for Windows PowerShell oder mit SDK. Wenn Sie ein Secrets Manager-Geheimnis referenzieren, muss der Name mit dem folgenden reservierten Pfad beginnen: `/aws/reference/secretsmanager/`. Durch die Angabe dieses Pfads wird Systems Manager mitgeteilt, das Geheimnis von Secrets Manager anstelle von Parameter Store abzurufen. Hier sind einige Beispielnamen, die korrekt auf die Geheimnisse des Secrets Managers verweisen, `CFCreds1` und `DBPass` mithilfe von Parameter Store referenzieren.

   - `/aws/reference/secretsmanager/CFCreds1`
   - `/aws/reference/secretsmanager/DBPass`

   Das folgende Java-Codebeispiel referenziert einen in Secrets Manager gespeicherten access-key und einen secret-key. In diesem Codebeispiel wird ein Amazon DynamoDB-Client eingerichtet. Der Code ruft die Konfigurationsdaten und Anmeldeinformationen von Parameter Store ab. Die Konfigurationsdaten werden als Zeichenfolgeparameter in Parameter Store und die Anmeldeinformationen in Secrets Manager gespeichert. Auch wenn die Konfigurationsdaten und Anmeldeinformationen in separaten Diensten gespeichert sind, kann auf beide Datensätze über Parameter Store mithilfe der `GetParameter`-API zugegriffen werden.

```
/**
* Initialize Systems Manager client with default credentials
*/
AWSSimpleSystemsManagement ssm =
 AWSSimpleSystemsManagementClientBuilder.defaultClient();

...

/**
* Example method to launch DynamoDB client with credentials different from default
* @return DynamoDB client
*/
AmazonDynamoDB getDynamoDbClient() {
    //Getting AWS credentials from Secrets Manager using GetParameter
    BasicAWSCredentials differentAWSCreds = new BasicAWSCredentials(
            getParameter("/aws/reference/secretsmanager/access-key"),
            getParameter("/aws/reference/secretsmanager/secret-key"));

    //Initialize the DynamoDB client with different credentials
    final AmazonDynamoDB client = AmazonDynamoDBClient.builder()
            .withCredentials(new AWSStaticCredentialsProvider(differentAWSCreds))
```

```
            .withRegion(getParameter("region")) //Getting configuration from
 Parameter Store
            .build();
    return client;
}

/**
* Helper method to retrieve parameter value
* @param parameterName identifier of the parameter
* @return decrypted parameter value
*/
public GetParameterResult getParameter(String parameterName) {
    GetParameterRequest request = new GetParameterRequest();
    request.setName(parameterName);
    request.setWithDecryption(true);
    return ssm.newGetParameterCall().call(request).getParameter().getValue();
}
```

Hier sind einige AWS CLI-Beispiele. Verwenden des `aws secretsmanager list-secrets`-Befehls, um die Namen Ihrer Geheimnisse zu finden.

AWS CLI Beispiel 1: Referenz durch Verwenden des Geheimnisnamens

Linux & macOS

```
aws ssm get-parameter \
    --name /aws/reference/secretsmanager/s1-secret \
    --with-decryption
```

Windows

```
aws ssm get-parameter ^
    --name /aws/reference/secretsmanager/s1-secret ^
    --with-decryption
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "Parameter": {
        "Name": "/aws/reference/secretsmanager/s1-secret",
        "Type": "SecureString",
```

```
        "Value": "Fl*MEishm!al875",
        "Version": 0,
        "SourceResult":
              "{
                   \"CreatedDate\": 1526334434.743,
                   \"Name\": \"s1-secret\",
                   \"VersionId\": \"aaabbbccc-1111-222-333-123456789\",
                   \"SecretString\": \"Fl*MEishm!al875\",
                   \"VersionStages\": [\"AWSCURRENT\"],
                   \"ARN\": \"arn:aws:secretsmanager:us-
east-2:123456789012:secret:s1-secret-E18LRP\"
               }"
        "LastModifiedDate": 2018-05-14T21:47:14.743Z,
        "ARN": "arn:aws:secretsmanager:us-east-2:123456789012:secret:s1-secret-
              E18LRP",
      }
}
```

AWS CLI-Beispiel 2: Referenz mit der Versions-ID

Linux & macOS

```
aws ssm get-parameter \
    --name /aws/reference/secretsmanager/s1-secret:11111-aaa-bbb-ccc-123456789 \
    --with-decryption
```

Windows

```
aws ssm get-parameter ^
    --name /aws/reference/secretsmanager/s1-secret:11111-aaa-bbb-ccc-123456789 ^
    --with-decryption
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "Parameter": {
        "Name": "/aws/reference/secretsmanager/s1-secret",
        "Type": "SecureString",
        "Value": "Fl*MEishm!al875",
        "Version": 0,
        "SourceResult":
```

```
              "{
                   \"CreatedDate\": 1526334434.743,
                   \"Name\": \"s1-secret\",
                   \"VersionId\": \"11111-aaa-bbb-ccc-123456789\",
                   \"SecretString\": \"Fl*MEishm!al875\",
                   \"VersionStages\": [\"AWSCURRENT\"],
                   \"ARN\": \"arn:aws:secretsmanager:us-
east-2:123456789012:secret:s1-secret-E18LRP\"
               }"
        "Selector": ":11111-aaa-bbb-ccc-123456789"
      }
        "LastModifiedDate": 2018-05-14T21:47:14.743Z,
        "ARN": "arn:aws:secretsmanager:us-east-2:123456789012:secret:s1-secret-
               E18LRP",
}
```

AWS CLI Beispiel 3: Referenz mit der Versionsstufe

Linux & macOS

```
aws ssm get-parameter \
    --name /aws/reference/secretsmanager/s1-secret:AWSCURRENT \
    --with-decryption
```

Windows

```
aws ssm get-parameter ^
    --name /aws/reference/secretsmanager/s1-secret:AWSCURRENT ^
    --with-decryption
```

Der Befehl gibt Informationen wie die folgenden zurück.

```
{
    "Parameter": {
        "Name": "/aws/reference/secretsmanager/s1-secret",
        "Type": "SecureString",
        "Value": "Fl*MEishm!al875",
        "Version": 0,
        "SourceResult":
              "{
                   \"CreatedDate\": 1526334434.743,
```

```
                    \"Name\": \"s1-secret\",
                    \"VersionId\": \"11111-aaa-bbb-ccc-123456789\",
                    \"SecretString\": \"Fl*MEishm!al875\",
                    \"VersionStages\": [\"AWSCURRENT\"],
                    \"ARN\": \"arn:aws:secretsmanager:us-
east-2:123456789012:secret:s1-secret-E18LRP\"
                }"
          "Selector": ":AWSCURRENT"
       }
       "LastModifiedDate": 2018-05-14T21:47:14.743Z,
       "ARN": "arn:aws:secretsmanager:us-east-2:123456789012:secret:s1-secret-
                 E18LRP",
}
```

AWS KMS Verschlüsselung für AWS Systems ManagerParameter Store SecureString Parameter

Mit können Sie SecureString Parameter erstellen AWS Systems Manager Parameter Store, bei denen es sich um Parameter handelt, die einen Klartext-Parameternamen und einen verschlüsselten Parameterwert haben. Parameter Storeverwendet AWS KMS , um die Parameterwerte von Parametern zu verschlüsseln und zu entschlüsseln. `SecureString`

Mit Parameter Store können Sie Daten als Parameter mit Werten erstellen, speichern und verwalten. Sie können einen Parameter in Parameter Store erstellen und in mehreren Anwendungen und Services verwenden, die den von Ihnen entworfenen Richtlinien und Berechtigungen unterliegen. Wenn Sie einen Parameterwert ändern müssen, ändern Sie eine Instance und führen keine fehleranfälligen Änderungen an verschiedenen Quellen durch. Parameter Store unterstützt eine hierarchische Struktur für Parameternamen, sodass Sie einen Parameter für spezifische Zwecke qualifizieren können.

Um vertrauliche Daten zu verwalten, können Sie Parameter erstellen`SecureString`. Parameter Storeverwendet AWS KMS keys , um die Parameterwerte von `SecureString` Parametern zu verschlüsseln, wenn Sie sie erstellen oder ändern. Zudem werden KMS-Schlüssel zum Entschlüsseln der Parameterwerte eingesetzt, wenn Sie auf diese zugreifen. Sie können den Von AWS verwalteter Schlüssel verwenden, den Parameter Store für Ihr Konto erstellt, oder Ihren eigenen kundenverwalteten Schlüssel angeben.

> **Important**
>
> Parameter Store unterstützt nur symmetrische KMS-Schlüssel. Sie können keinen asymmetrischen KMS-Schlüssel verwenden, um Ihre Parameter zu verschlüsseln. Hilfe zum Bestimmen, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, finden Sie unter Identifizieren unterschiedlicher Schlüsseltypen im Entwicklerhandbuch für AWS Key Management Service .

Parameter Store unterstützt zwei Stufen von `SecureString`-Parametern: Standard und Erweitert. Standard-Parameter, die 4096 Bytes nicht überschreiten können, werden direkt mit dem von Ihnen angegebenen KMS-Schlüssel verschlüsselt und entschlüsselt. Zum Verschlüsseln und Entschlüsseln erweiterten `SecureString`-Parameter verwendet Parameter Store Envelope-Verschlüsselung mit dem AWS Encryption SDK. Es ist möglich, einen `SecureString`-Standardparameter in einen erweiterten Parameter zu konvertieren, aber nicht einen erweiterten Parameter in einen Standardparameter. Weitere Informationen zum Unterschied zwischen Standard- und erweiterten `SecureString`-Parametern finden Sie unter Verwalten von Parameterstufen.

Themen

- Schutz von Standardparametern SecureString
- Schutz erweiterter Parameter SecureString
- Festlegen der Berechtigungen zum Verschlüsseln und Entschlüsseln von Parameterwerten
- Parameter Store-Verschlüsselungskontext
- Beheben von Problemen mit KMS-Schlüsseln in Parameter Store

Schutz von Standardparametern SecureString

Parameter Store führt keine kryptografischen Operationen durch. Stattdessen nutzt es AWS KMS zur Ver- und Entschlüsselung von `SecureString`-Parameterwerten. Parameter StoreRuft den Vorgang AWS KMS Verschlüsseln auf, wenn Sie einen `SecureString` Standardparameterwert erstellen oder ändern. Diese Operation verwendet einen KMS-Schlüssel mit symmetrischer Verschlüsselung direkt zum Verschlüsseln des Parameterwerts, statt mit dem KMS-Schlüssel einen Datenschlüssel zu generieren.

Sie können den KMS-Schlüssel auswählen, den Parameter Store zum Verschlüsseln des Parameterwerts nutzt. Wenn Sie keinen KMS-Schlüssel angeben, verwendet Parameter Store den

Von AWS verwalteter Schlüssel , den Systems Manager automatisch in Ihrem Konto erstellt. Der KMS-Schlüssel hat den Alias `aws/ssm`.

Verwenden Sie den DescribeKeyVorgang in der AWS KMS API, um den `aws/ssm` Standard-KMS-Schlüssel für Ihr Konto anzuzeigen. Im folgenden Beispiel wird der `describe-key` Befehl in der AWS Command Line Interface (AWS CLI) mit dem `aws/ssm` Aliasnamen verwendet.

```
aws kms describe-key \
    --key-id alias/aws/ssm
```

Verwenden Sie den PutParameterVorgang in der Systems Manager Manager-API, um einen `SecureString` Standardparameter zu erstellen. Lassen Sie den Parameter `Tier` weg oder geben Sie als Wert `Standard` ein,wobei es sich um den Standardwert handelt. Schließen Sie einen `Type`-Parameter mit einem Wert von `SecureString` ein. Nutzen Sie zum Angeben eines KMS-Schlüssels den `KeyId`-Parameter. Die Standardeinstellung ist Von AWS verwalteter Schlüssel für Ihr Konto,`aws/ssm`.

Parameter Storeruft dann den AWS KMS `Encrypt` Vorgang mit dem KMS-Schlüssel und dem Klartext-Parameterwert auf. AWS KMS gibt den verschlüsselten Parameterwert zurück, der zusammen mit dem Parameternamen Parameter Store gespeichert wird.

Das folgende Beispiel verwendet den Put-Parameter-Befehl von Systems Manager und seinen `--type` Parameter in, um einen `SecureString` Parameter AWS CLI zu erstellen. Da der Befehl die optionalen Parameter `--tier` und `--key-id` weglässt, erstellt Parameter Store einen `SecureString`-Standardparameter und veschlüsselt ihn unter dem Von AWS verwalteter Schlüssel.

```
aws ssm put-parameter \
    --name MyParameter \
    --value "secret_value" \
    --type SecureString
```

Im folgenden ähnlichen Beispiel wird der `--key-id`-Parameter zur Angabe eines kundenverwalteten Schlüssels verwendet. Im Beispiel wird eine KMS-Schlüssel-ID verwendet, um den KMS-Schlüssel zu identifizieren, Sie können jedoch einen beliebigen gültigen KMS-Schlüsselbezeichner verwenden. Da der Befehl den Parameter `Tier` weglässt (`--tier`), erstellt Parameter Store einen `SecureString`-Standardparameter und keinen erweiterten.

```
aws ssm put-parameter \
```

```
    --name param1 \
    --value "secret" \
    --type SecureString \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Wenn Sie einen `SecureString`-Parameter von Parameter Store erhalten, ist sein Wert verschlüsselt. Verwenden Sie den GetParameter Vorgang in der Systems Manager Manager-API, um einen Parameter abzurufen.

Im folgenden Beispiel wird der Systems Manager Manager-Befehl get-parameter in der verwendet AWS CLI , um den `MyParameter` Parameter von abzurufen, Parameter Store ohne seinen Wert zu entschlüsseln.

```
aws ssm get-parameter --name MyParameter
```

```
{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value": "AQECAHgnOkMROh5LaLXkA4j0+vYi6tmM17Lg"
    }
}
```

Zum Entschlüsseln des Parameterwerts vor der Rückgabe setzen Sie den `WithDecryption`-Parameter `GetParameter` auf `true`. Wenn Sie verwenden`WithDecryption`, Parameter Store ruft in Ihrem Namen den Vorgang AWS KMS Decrypt auf, um den Parameterwert zu entschlüsseln. Infolgedessen gibt die `GetParameter`-Anforderung den Parameter mit einem Klartext-Parameterwert zurück, wie im folgenden Beispiel gezeigt.

```
aws ssm get-parameter \
    --name MyParameter \
    --with-decryption
```

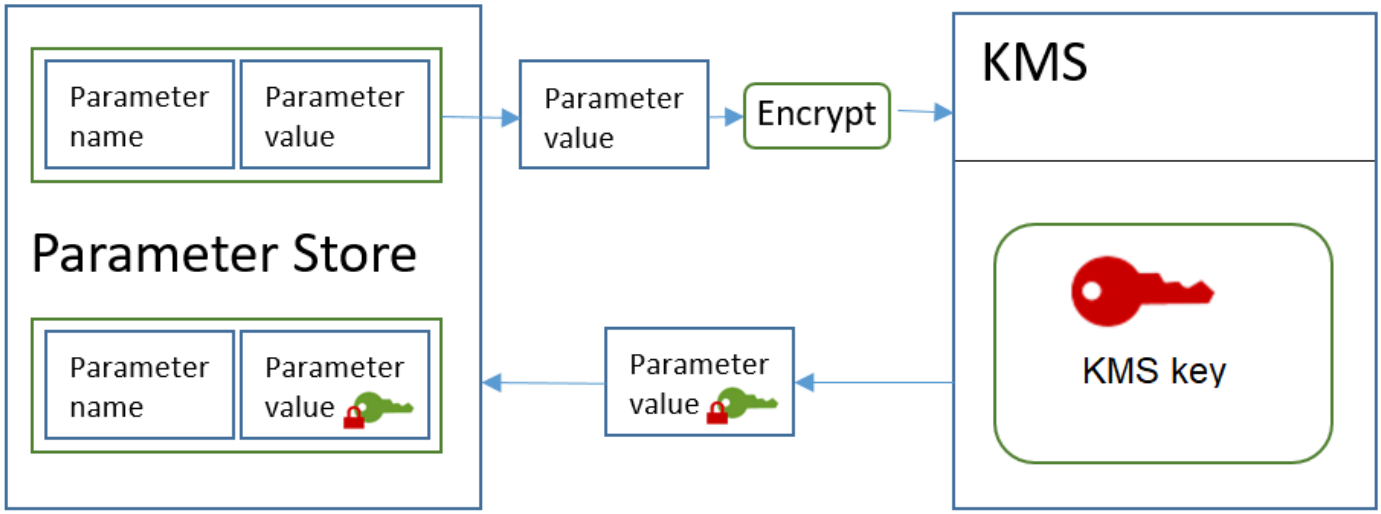
```
{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value": "secret_value"
    }
```

```
}
```

Der folgende Workflow zeigt, wie Parameter Store einen KMS-Schlüssel zum Verschlüsseln und Entschlüsseln eines Standard-`SecureString`-Parameters verwendet.

Verschlüsseln eines Standardparameters

1. Wenn Sie mit `PutParameter` einen `SecureString`-Parameter erstellen, sendet Parameter Store eine `Encrypt`-Anforderung an AWS KMS. Diese Anforderung enthält den Klartext-Parameterwert, den von Ihnen ausgewählten KMS-Schlüssel und den Parameter Store-Verschlüsselungskontext. Während der Übertragung an AWS KMS wird der Klartextwert im `SecureString` Parameter durch Transport Layer Security (TLS) geschützt.
2. AWS KMS verschlüsselt den Parameterwert mit dem angegebenen KMS-Schlüssel und Verschlüsselungskontext. Der Verschlüsselungstext wird an Parameter Store zurückgegeben. Dort werden der Parametername und der dazugehörige verschlüsselte Wert gespeichert.



Entschlüsseln eines Standardparameters

1. Wenn Sie den `WithDecryption` Parameter in eine `GetParameter` Anforderung aufnehmen, Parameter Store sendet eine `Decrypt` Anfrage AWS KMS mit dem verschlüsselten `SecureString` Parameterwert und dem Parameter StoreVerschlüsselungskontext an.
2. AWS KMS verwendet denselben KMS-Schlüssel und den angegebenen Verschlüsselungskontext, um den verschlüsselten Wert zu entschlüsseln. Es gibt den Klartext-Parameterwert (den entschlüsselten) an Parameter Store zurück. Während der Übertragung sind die Klartextdaten durch TLS geschützt.

3. Parameter Store gibt den Klartext-Parameterwert an Sie in der `GetParameter`-Antwort zurück.

Schutz erweiterter Parameter SecureString

Wenn Sie einen erweiterten `SecureString` Parameter erstellen, Parameter Store verwendet Envelope-Verschlüsselung mit dem AWS Encryption SDK und eine symmetrische Verschlüsselung, AWS KMS key um den Parameterwert zu schützen. `PutParameter` Jeder erweiterte Parameterwert ist mit einem eindeutigen Datenschlüssel verschlüsselt, der wiederum mit einem KMS-Schlüssel verschlüsselt ist. Sie können den Von AWS verwalteter Schlüssel für das Konto (`aws/ssm`) oder einen beliebigen kundenverwalteten Schlüssel verwenden.

Das AWS Encryption SDK ist eine clientseitige Open-Source-Bibliothek, mit der Sie Daten mithilfe von Branchenstandards und bewährten Methoden leichter verschlüsseln und entschlüsseln können. Sie wird auf mehreren Plattformen und in mehreren Programmiersprachen, einschließlich einer Befehlszeilenschnittstelle, unterstützt. In können Sie sich den Quellcode ansehen und zu seiner Entwicklung beitragen. GitHub

Parameter StoreRuft für jeden `SecureString` Parameterwert den AWS Encryption SDK auf, um den Parameterwert mithilfe eines eindeutigen Datenschlüssels zu verschlüsseln, der AWS KMS generiert (GenerateDataKey). Das AWS Encryption SDK kehrt zu Parameter Store einer verschlüsselten Nachricht zurück, die den verschlüsselten Parameterwert und eine verschlüsselte Kopie des eindeutigen Datenschlüssels enthält. Parameter Storespeichert die gesamte verschlüsselte Nachricht im `SecureString` Parameterwert. Wenn Sie dann einen erweiterten `SecureString` Parameterwert erhalten, Parameter Store verwendet den, AWS Encryption SDK um den Parameterwert zu entschlüsseln. Dazu ist ein Aufruf von erforderlich, AWS KMS um den verschlüsselten Datenschlüssel zu entschlüsseln.

Verwenden Sie den PutParameterVorgang in der Systems Manager Manager-API, um einen erweiterten `SecureString` Parameter zu erstellen. Stellen Sie den Wert des Parameters `Tier` auf `Advanced` ein. Schließen Sie einen `Type`-Parameter mit einem Wert von `SecureString` ein. Nutzen Sie zum Angeben eines KMS-Schlüssels den `KeyId`-Parameter. Die Standardeinstellung ist Von AWS verwalteter Schlüssel für Ihr Konto,`aws/ssm`.

```
aws ssm put-parameter \
    --name MyParameter \
    --value "secret_value" \
    --type SecureString \
    --tier Advanced
```

Im folgenden ähnlichen Beispiel wird der `--key-id`-Parameter zur Angabe eines kundenverwalteten KMS-Schlüssels verwendet. Das Beispiel verwendet den Amazon-Ressourcennamen (ARN) des KMS-Schlüssels, aber Sie können jeden gültigen KMS-Schlüsselbezeichner angeben.

```
aws ssm put-parameter \
    --name MyParameter \
    --value "secret_value" \
    --type SecureString \
    --tier Advanced \
    --key-id arn:aws:kms:us-
east-2:987654321098:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

Wenn Sie einen `SecureString` Parameter von abrufenParameter Store, entspricht sein Wert der verschlüsselten Nachricht, die AWS Encryption SDK zurückgegeben wurde. Verwenden Sie den GetParameterVorgang in der Systems Manager Manager-API, um einen Parameter abzurufen.

Das folgende Beispiel verwendet die Systems-Manager-Operation `GetParameter` zum Anfordern des `MyParameter`-Parameters von Parameter Store, ohne dessen Wert zu entschlüsseln.

```
aws ssm get-parameter --name MyParameter
```

```
{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value": "AQECAHgnOkMROh5LaLXkA4j0+vYi6tmM17Lg"
    }
}
```

Zum Entschlüsseln des Parameterwerts vor der Rückgabe setzen Sie den `WithDecryption`-Parameter `GetParameter` auf `true`. Wenn Sie verwenden`WithDecryption`, Parameter Store ruft in Ihrem Namen den Vorgang AWS KMS Decrypt auf, um den Parameterwert zu entschlüsseln. Infolgedessen gibt die `GetParameter`-Anforderung den Parameter mit einem Klartext-Parameterwert zurück, wie im folgenden Beispiel gezeigt.

```
aws ssm get-parameter \
    --name MyParameter \
    --with-decryption
```

```
{
```

```
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value": "secret_value"
    }
}
```

Es ist nicht möglich, einen erweiterten `SecureString`-Parameter in einen erweiterten Standardparameter zu konvertieren, aber es ist möglich einen `SecureString`-Standardparameter in einen erweiterten Parameter zu konvertieren. Um einen `SecureString`-Standardparameter in einen erweiterten `SecureString`-Parameter zu konvertieren, verwenden Sie die Operation `PutParameter` mit dem Parameter `Overwrite`. `Type` muss den Wert `SecureString` und `Tier` muss den Wert `Advanced` haben. Der `KeyId`-Parameter, der einen kundenverwalteten Schlüssel identifiziert, ist optional. Wenn Sie es weglassen, Parameter Store verwendet das Von AWS verwalteter Schlüssel für das Konto. Sie können jeden KMS-Schlüssel angeben, den der Prinzipal verwenden darf, selbst wenn der Standardparameter von Ihnen mit einem anderen KMS-Schlüssel verschlüsselt wurde.

Wenn Sie den Parameter `Overwrite` verwenden, verschlüsselt Parameter Store den Parameterwert mithilfe des AWS Encryption SDK . Anschließend wird die neu verschlüsselte Nachricht in Parameter Store gespeichert.

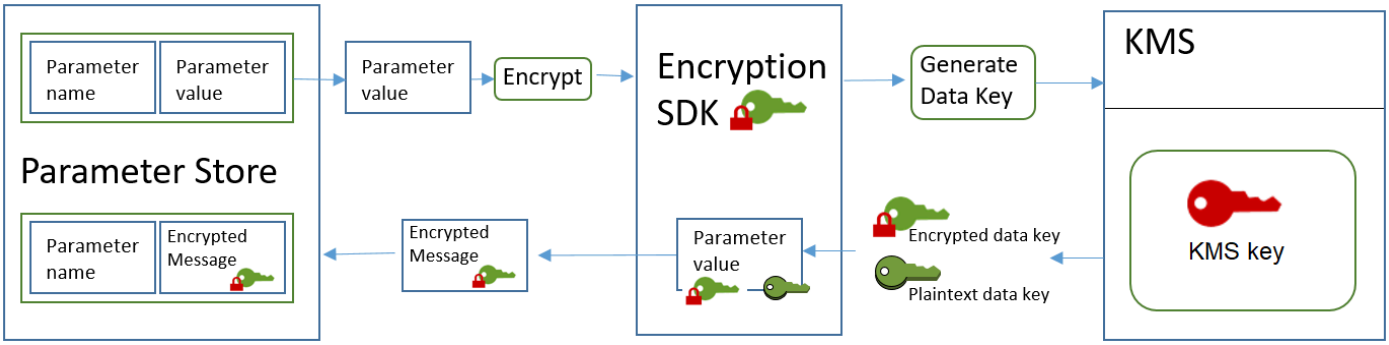
```
aws ssm put-parameter \
    --name myStdParameter \
    --value "secret_value"  \
    --type SecureString \
    --tier Advanced \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --overwrite
```

Der folgende Workflow zeigt, wie Parameter Store einen KMS-Schlüssel zum Verschlüsseln und Entschlüsseln eines erweiterten `SecureString`-Parameters verwendet.

Verschlüsseln eines erweiterten Parameters

1. Wenn Sie einen erweiterten `SecureString` Parameter erstellen, Parameter Store verwendet das AWS Encryption SDK und, um den Parameterwert AWS KMS zu verschlüsseln. `PutParameter` Parameter Storeruft den AWS Encryption SDK mit dem Parameterwert, dem von Ihnen angegebenen KMS-Schlüssel und dem Parameter StoreVerschlüsselungskontext auf.

2. Der AWS Encryption SDK sendet eine GenerateDataKeyAnfrage AWS KMS mit der ID des von Ihnen angegebenen KMS-Schlüssels und dem Parameter Store Verschlüsselungskontext an. AWS KMS gibt zwei Kopien des eindeutigen Datenschlüssels zurück: eine im Klartext und eine mit dem KMS-Schlüssel verschlüsselte Kopie. (Der Verschlüsselungskontext wird beim Verschlüsseln des Datenschlüssels verwendet.)
3. Der AWS Encryption SDK verwendet den Klartext-Datenschlüssel, um den Parameterwert zu verschlüsseln. Es wird eine verschlüsselte Nachricht mit dem verschlüsselten Parameterwert, dem verschlüsselten Datenschlüssel und anderen Daten, einschließlich des Parameter Store-Verschlüsselungskontextes, zurückgegeben.
4. Parameter Store speichert die verschlüsselte Nachricht als Parameterwert.



Entschlüsseln eines erweiterten Parameters

1. Sie können den `WithDecryption`-Parameter in eine `GetParameter`-Anforderung einschließen, um einen erweiterten `SecureString`-Parameter zu erhalten. In diesem Fall übergibt Parameter Store die verschlüsselte Nachricht aus dem Parameterwert an eine Entschlüsselungsmethode des AWS Encryption SDK.
2. Der AWS Encryption SDK ruft den AWS KMS Decrypt-Vorgang auf. Sie übergibt den verschlüsselten Datenschlüssel und den Parameter Store-Verschlüsselungskontext aus der verschlüsselten Nachricht.
3. AWS KMS verwendet den KMS-Schlüssel und den Parameter Store Verschlüsselungskontext, um den verschlüsselten Datenschlüssel zu entschlüsseln. Anschließend gibt die Operation den Klartext-Datenschlüssel (entschlüsselt) an das AWS Encryption SDK zurück.
4. Der AWS Encryption SDK verwendet den Klartext-Datenschlüssel, um den Parameterwert zu entschlüsseln. Der Klartext-Parameterwert wird an Parameter Store zurückgegeben.
5. Parameter Store überprüft den Verschlüsselungskontext und gibt den Klartext-Parameterwert in der `GetParameter`-Antwort an Sie zurück.

Festlegen der Berechtigungen zum Verschlüsseln und Entschlüsseln von Parameterwerten

Zum Verschlüsseln eines `SecureString`-Standardparameterwerts benötigt der Benutzer die Berechtigung `kms:Encrypt`. Zum Verschlüsseln eines erweiterten `SecureString`-Parameterwerts benötigt der Benutzer die Berechtigung `kms:GenerateDataKey`. Zum Entschlüsseln des `SecureString`-Parameterwerts beider Typen benötigt der Benutzer die Berechtigung `kms:Decrypt`.

Sie können AWS Identity and Access Management (IAM) -Richtlinien verwenden, um einem Benutzer die Erlaubnis zu gewähren oder zu verweigern, den Systems Manager `PutParameter` und `GetParameter` Operations aufzurufen.

Wenn Sie Ihre sicheren `SecureString`-Parameterwerte mit benutzerverwalteten Schlüsseln verschlüsseln, können Sie zum Verwalten der Verschlüsselungs- und Entschlüsselungs-Berechtigungen IAM-Richtlinien und Schlüsselrichtlinien verwenden. Sie können jedoch keine Zugriffssteuerungs-Richtlinien für den standardmäßigen `aws/ssm`-KMS-Schlüssel einrichten. Ausführliche Informationen zur Steuerung des Zugriffs auf vom Kunden verwaltete Schlüssel finden Sie unter KMS-Schlüsselzugriff und -Berechtigungen im AWS Key Management Service - Entwicklerhandbuch.

Das folgende Beispiel zeigt eine IAM-Richtlinie, die für `SecureString`-Standard-Parameter bestimmt ist. Sie ermöglicht dem Benutzer den Aufruf der Systems-Manager-Operation `PutParameter` für alle Parameter im Pfad `FinancialParameters`. Die Richtlinie ermöglicht es dem Benutzer auch, den AWS KMS `Encrypt` Vorgang mit einem vom Kunden verwalteten Beispielschlüssel aufzurufen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/
FinancialParameters/*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt"
            ],
            "Resource": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

Das folgende Beispiel zeigt eine IAM-Richtlinie an, die für erweiterte `SecureString`-Parameter bestimmt ist. Sie ermöglicht dem Benutzer den Aufruf der Systems-Manager-Operation `PutParameter` für alle Parameter im Pfad `ReservedParameters`. Die Richtlinie ermöglicht es dem Benutzer auch, den AWS KMS `GenerateDataKey` Vorgang mit einem vom Kunden verwalteten Beispielschlüssel aufzurufen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/
ReservedParameters/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-id"
        }
    ]
}
```

Das letzte Beispiel zeigt eine IAM-Richtlinie, die für standardmäßige oder erweiterte `SecureString`-Parameter verwendet werden kann. Sie ermöglicht dem Benutzer den Aufruf der Systems-Manager-Operation `GetParameter` (und verwandte Operationen) für alle Parameter im Pfad `ITParameters`. Die Richtlinie ermöglicht es dem Benutzer auch, den AWS KMS `Decrypt` Vorgang mit einem vom Kunden verwalteten Beispielschlüssel aufzurufen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter*"
            ],
            "Resource": "arn:aws:ssm:us-east-1:111122223333:parameter/
ITParameters/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

Parameter Store-Verschlüsselungskontext

Ein Verschlüsselungskontext ist eine Gruppe von Schlüssel/Wert-Paaren mit willkürlichen, nicht geheimen Daten. Wenn Sie einen Verschlüsselungskontext in eine Anfrage zur Datenverschlüsselung aufnehmen, wird der Verschlüsselungskontext AWS KMS kryptografisch an die verschlüsselten Daten gebunden. Zur Entschlüsselung der Daten müssen Sie denselben Verschlüsselungskontext übergeben.

Sie können den Verschlüsselungskontext auch nutzen, um eine kryptografische Operation in Audit-Datensätzen und -Protokollen zu identifizieren. Die Verschlüsselungskontext wird in Klartext-Protokollen wie AWS CloudTrail angezeigt.

Der benötigt AWS Encryption SDK ebenfalls einen Verschlüsselungskontext, der jedoch unterschiedlich behandelt wird. Parameter Storestellt den Verschlüsselungskontext für die Verschlüsselungsmethode bereit. Das bindet den Verschlüsselungskontext AWS Encryption SDK kryptografisch an die verschlüsselten Daten. Es schließt den Verschlüsselungskontext als Klartext im Header der von ihm zurückgegebenen verschlüsselten Nachricht ein. Im Gegensatz AWS KMS dazu verwenden die AWS Encryption SDK Entschlüsselungsmethoden jedoch keinen Verschlüsselungskontext als Eingabe. Stattdessen wird beim Entschlüsseln von Daten der Verschlüsselungskontext aus der verschlüsselten Nachricht AWS Encryption SDK abgerufen. Parameter Storeüberprüft, ob der Verschlüsselungskontext den erwarteten Wert enthält, bevor der Klartext-Parameterwert an Sie zurückgegeben wird.

Parameter Store verwendet den folgenden Verschlüsselungskontext in seinen kryptografischen Operationen:

- Schlüssel: `PARAMETER_ARN`
- Wert: Der Amazon-Ressourcenname (ARN) des Parameters, der verschlüsselt wird.

Das Format des Verschlüsselungskontexts sieht wie folgt aus:

```
"PARAMETER_ARN":"arn:aws:ssm:region-id:account-id:parameter/parameter-name"
```

Parameter StoreSchließt diesen Verschlüsselungskontext beispielsweise in Aufrufe zum Verschlüsseln und Entschlüsseln des `MyParameter` Parameters in einem Beispiel und einer Region ein. AWS-Konto

```
"PARAMETER_ARN":"arn:aws:ssm:us-east-2:111122223333:parameter/MyParameter"
```

Befindet sich der Parameter in einem hierarchischen Parameter Store-Pfad, sind sowohl der Pfad als auch der Name im Verschlüsselungskontext enthalten. Dieser Verschlüsselungskontext wird beispielsweise beim Verschlüsseln und Entschlüsseln des `MyParameter` Parameters im `/ReadableParameters` Pfad in einem Beispiel und einer Region verwendet. AWS-Konto

```
"PARAMETER_ARN":"arn:aws:ssm:us-east-2:111122223333:parameter/ReadableParameters/
MyParameter"
```

Sie können einen verschlüsselten `SecureString` Parameterwert entschlüsseln, indem Sie den AWS KMS `Decrypt` Vorgang mit dem richtigen Verschlüsselungskontext und dem verschlüsselten Parameterwert aufrufen, den der Systems Manager `GetParameter` Manager-Vorgang zurückgibt. Wir empfehlen jedoch, die Parameter Store-Parameterwerte mithilfe der `GetParameter`-Operation mit dem Parameter `WithDecryption` zu entschlüsseln.

Sie können den Verschlüsselungskontext auch in eine IAM-Richtlinie einschließen. So können Sie einen Benutzer beispielsweise nur dazu berechtigen, einen bestimmten Parameterwert oder eine bestimmte Gruppe von Parameterwerten zu entschlüsseln.

Das folgende Beispiel einer IAM-Richtlinie erlaubt es dem Benutzer, den Wert des `MyParameter`-Parameters abzurufen und den Wert mit dem angegebenen KMS-Schlüssel zu entschlüsseln. Diese Berechtigungen gelten jedoch nur, wenn der Verschlüsselungskontext mit der angegebenen Zeichenfolge übereinstimmt. Diese Berechtigungen gelten nicht für andere Parameter oder KMS-Schlüssel und der Aufruf an `GetParameter` schlägt fehl, wenn der Verschlüsselungskontext nicht mit der Zeichenfolge übereinstimmt.

Bevor Sie eine Richtlinienaussage wie diese verwenden, ersetzen Sie die *`example ARNs`* durch gültige Werte.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter*"
            ],
            "Resource": "arn:aws:ssm:us-
east-1:111122223333:parameter/MyParameter"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-id",
            "Condition": {
                "StringEquals": {
```

```
                    "kms:EncryptionContext:PARAMETER_ARN":"arn:aws:ssm:us-
east-1:111122223333:parameter/MyParameter"
                }
            }
        }
    ]
}
```

Beheben von Problemen mit KMS-Schlüsseln in Parameter Store

Um eine Operation mit einem `SecureString` Parameter ausführen zu können, Parameter Store müssen Sie in der Lage sein, den AWS KMS KMS-Schlüssel zu verwenden, den Sie für den gewünschten Vorgang angeben. Die meisten Parameter Store-Fehler im Zusammenhang mit KMS-Schlüsseln treten aus folgenden Gründen auf:

- Die Anmeldeinformationen, die eine Anwendung verwendet, sind nicht berechtigt, eine bestimmte Aktion mit einem KMS-Schlüssel durchzuführen.

  Zur Behebung dieses Problems führen Sie die Anwendung mit anderen Anmeldeinformationen aus oder ändern die IAM- oder Schlüssel-Richtlinie, die diese Operation verhindert. Hilfe zu AWS KMS IAM und wichtigen Richtlinien finden Sie unter KMS-Schlüsselzugriff und -berechtigungen im AWS Key Management Service Entwicklerhandbuch.
- Der KMS-Schlüssel wurde nicht gefunden.

  Dies geschieht in der Regel, wenn Sie einen falschen Bezeichner für den KMS-Schlüssel verwenden. Finden Sie die richtigen Bezeichner für den KMS-Schlüssel und wiederholen Sie den Befehl.
- Der KMS-Schlüssel ist nicht aktiviert. In diesem Fall wird eine InvalidKeyIdAusnahme mit einer ausführlichen Fehlermeldung von Parameter Store AWS KMS zurückgegeben. Wenn der Schlüsselstatus des KMS-Schlüssels `Disabled` lautet, aktivieren Sie ihn. Wenn der Status `Pending Import` lautet, führen Sie das Importverfahren durch. Lautet der Schlüsselstatus `Pending Deletion`, brechen Sie die Schlüssellöschung ab oder verwenden Sie einen anderen KMS-Schlüssel.

  Verwenden Sie den DescribeKeyVorgang, um den Schlüsselstatus eines KMS-Schlüssels zu ermitteln.

Verwenden von Parameter Store-Parametern in Amazon Elastic Kubernetes Service

Um Parameter aus Parameter Store, einem Tool in AWS Systems Manager, als in Amazon-EKS-Pods bereitgestellte Dateien anzuzeigen, können Sie AWS Secrets and Configuration Provider (ASCP) für die Kubernetes-Secrets-Store-CSI-Treiber verwenden. ASCP funktioniert mit Amazon Elastic Kubernetes Service 1.17+, der eine Knotengruppe von Amazon EC2 ausführt. Knotengruppen von AWS Fargate werden nicht unterstützt.

Mit ASCP können Sie Ihre Parameter in Parameter Store speichern und verwalten und sie dann über Ihre auf Amazon EKS ausgeführten Workloads abrufen. Wenn ein Parameter mehrere Schlüssel-Wert-Paare im JSON-Format enthält, können Sie auswählen, welche davon in Amazon EKS bereitgestellt werden sollen. ASCP verwendet JMESPath-Syntax, um die Schlüssel-Wert-Paare in Ihrem Secret abzufragen. ASCP funktioniert auch mit AWS Secrets Manager-Secrets.

ASCP bietet zwei Authentifizierungsmethoden mit Amazon EKS. Der erste Ansatz verwendet IAM Roles for Service Accounts (IRSA). Der zweite Ansatz verwendet Pod Identities. Jeder Ansatz hat eigene Vorteile und Anwendungsfälle.

ASCP mit IAM Roles for Service Accounts (IRSA)

ASCP mit IAM Roles for Service Accounts (IRSA) ermöglicht Ihnen, Parameter aus Parameter Store als Dateien in Ihren Amazon-EKS-Pods zu mounten. Dieser Ansatz ist für folgende Szenarien geeignet:

- Sie müssen Parameter als Dateien in Ihren Pods mounten.
- Sie verwenden Amazon-EKS-Version 1.17 oder höher mit Amazon-EC2-Knotengruppen.
- Sie möchten bestimmte Schlüssel-Wert-Paare aus JSON-formatierten Parametern abrufen.

Weitere Informationen finden Sie unter the section called “Integrieren von ASCP in IRSA für Amazon EKS”.

ASCP mit Pod Identity

ASCP mit EKS Pod Identity

ASCP mit Pod Identity erhöht die Sicherheit und vereinfacht die Konfiguration für den Zugriff auf Parameter in Parameter Store. Dieser Ansatz bietet Vorteile in folgenden Szenarien:

- Sie benötigen eine detailliertere Berechtigungsverwaltung auf Pod-Ebene.
- Sie verwenden Amazon-EKS-Version 1.24 oder höher.
- Sie benötigen eine höhere Leistung und Skalierbarkeit.

Weitere Informationen finden Sie unter the section called “Integrieren von ASCP in Pod Identity für Amazon EKS”.

Den richtigen Ansatz wählen

Bei der Entscheidung zwischen ASCP mit IRSA und ASCP mit Pod Identity sollten Sie die folgenden Faktoren berücksichtigen:

- Amazon-EKS-Version: Pod Identity erfordert Amazon EKS 1.24+, während der CSI-Treiber mit Amazon EKS 1.17+ funktioniert.
- Sicherheitsanforderungen: Pod Identity bietet eine detailliertere Kontrolle auf Pod-Ebene.
- Leistung: Pod Identity schneidet in umfassenden Umgebungen im Allgemeinen besser ab.
- Komplexität: Pod Identity vereinfacht die Einrichtung, da keine separaten Servicekonten erforderlich sind.

Wählen Sie die Methode, die am besten zu Ihren spezifischen Anforderungen und Ihrer Amazon-EKS-Umgebung passt.

Installieren von ASCP für Amazon EKS

In diesem Abschnitt wird erklärt, wie Sie AWS Secrets and Configuration Provider für Amazon EKS installieren. Mit ASCP können Sie Parameter aus Parameter Store und Secrets aus AWS Secrets Manager als Dateien in Amazon-EKS-Pods mounten.

Voraussetzungen

- Ein Amazon-EKS-Cluster
  - Version 1.24 oder höher für Pod Identity
  - Version 1.17 oder höher für IRSA
- Die AWS CLI, installiert und konfiguriert
- kubectl, für Ihren Amazon-EKS-Cluster installiert und konfiguriert
- Helm (Version 3.0 oder höher)

Installieren und Konfigurieren von ASCP

Der ASCP ist auf GitHub im Repository secrets-store-csi-provider-aws verfügbar. Das Repository enthält auch YAML-Beispieldateien zum Erstellen und Mounten eines Secrets durch das Ändern des `objectType`-Werts von `secretsmanager` zu `ssmparameter`.

Während der Installation können Sie festlegen, dass ASCP einen FIPS-Endpunkt verwenden soll. Eine Liste von Systems-Manager-Endpunkten finden Sie im Allgemeine Amazon Web Services-Referenz unter Systems-Manager-Serviceendpunkte.

Installieren Sie ASCP mit Helm wie folgt:

1. Um sicherzustellen, dass das Repository auf die neuesten Diagramme verweist, verwenden Sie `helm repo update.`.
2. Fügen Sie das Diagramm zum Secrets-Store-CSI-Treiber hinzu.

   ```
   helm repo add secrets-store-csi-driver https://kubernetes-sigs.github.io/secrets-store-csi-driver/charts
   ```

3. Installieren Sie das Diagramm. Um die Drosselung zu konfigurieren, fügen Sie das folgende Flag hinzu: `--set-json 'k8sThrottlingParams={"qps": "`*`number of queries per second`*`", "burst": "`*`number of queries per second`*`"}'`.

   ```
   helm install -n kube-system csi-secrets-store secrets-store-csi-driver/secrets-store-csi-driver
   ```

4. Fügen Sie das ASCP-Diagramm hinzu.

   ```
   helm repo add aws-secrets-manager https://aws.github.io/secrets-store-csi-driver-provider-aws
   ```

5. Installieren Sie das Diagramm. Um einen FIPS-Endpunkt zu verwenden, fügen Sie das folgende Flag hinzu: `--set useFipsEndpoint=true`.

   ```
   helm install -n kube-system secrets-provider-aws aws-secrets-manager/secrets-store-csi-driver-provider-aws
   ```

Installieren Sie ASCP mit YAML im Repository wie folgt:

- Verwenden Sie die folgenden Befehle.

```
helm repo add secrets-store-csi-driver https://kubernetes-sigs.github.io/secrets-
store-csi-driver/charts
helm install -n kube-system csi-secrets-store secrets-store-csi-driver/secrets-
store-csi-driver
kubectl apply -f https://raw.githubusercontent.com/aws/secrets-store-csi-driver-
provider-aws/main/deployment/aws-provider-installer.yaml
```

Überprüfen der Installationen

Gehen Sie folgendermaßen vor, um die Installationen Ihres EKS-Clusters, des Secrets Store CSI-Treibers und des ASCP-Plug-ins zu überprüfen:

1. Den EKS-Cluster überprüfen:

   ```
   eksctl get cluster --name clusterName
   ```

   Dieser Befehl muss Informationen zu Ihrem Cluster zurückgeben.

2. Die Secrets-Store-CSI-Treiberinstallation überprüfen:

   ```
   kubectl get pods -n kube-system -l app=secrets-store-csi-driver
   ```

   Sie sollten ausgeführte Pods mit Namen wie `csi-secrets-store-secrets-store-csi-driver-xxx` sehen.

3. Die ASCP-Plugin-Installation überprüfen:

   YAML installation

   ```
   $ kubectl get pods -n kube-system -l app=csi-secrets-store-provider-aws
   ```

   Beispielausgabe:

   ```
   NAME                                     READY   STATUS    RESTARTS   AGE
   csi-secrets-store-provider-aws-12345      1/1     Running   0          2m
   ```

Helm installation

```
$  kubectl get pods -n kube-system -l app=secrets-store-csi-driver-provider-aws
```

Beispielausgabe:

```
NAME                                                READY   STATUS    RESTARTS
 AGE
secrets-provider-aws-secrets-store-csi-driver-provider-67890       1/1
 Running   0          2m
```

Sie sollten Pods im Status `Running` sehen.

Wenn nach dem Ausführen dieser Befehle alles korrekt eingerichtet ist, sollten alle Komponenten fehlerfrei ausgeführt werden. Wenn Sie auf Probleme stoßen, müssen Sie diese möglicherweise beheben, indem Sie die Protokolle der jeweiligen Pods überprüfen, bei denen Probleme auftreten.

Fehlersuche

1. Führen Sie folgenden Befehl aus, um die Protokolle des ASCP-Anbieters zu überprüfen:

   ```
   kubectl logs -n kube-system -l app=csi-secrets-store-provider-aws
   ```

2. Überprüfen Sie den Status aller Pods im `kube-system` Namespace.

   Ersetzen Sie den *`Standard-Platzhaltertext`* durch Ihre eigene Pod-ID:

   ```
   kubectl -n kube-system get pods
   ```

   ```
   kubectl -n kube-system logs pod/pod-id
   ```

   Alle Pods, die sich auf den CSI-Treiber und ASCP beziehen, sollten sich im Status „In Ausführung“ befinden.

3. Die Version des CSI-Treibers überprüfen:

   ```
   kubectl get csidriver secrets-store.csi.k8s.io -o yaml
   ```

Dieser Befehl muss Informationen zu dem installierten CSI-Treiber zurückgeben.

Weitere Ressourcen

Weitere Informationen zur Verwendung von ASCP mit Amazon EKS finden Sie in den folgenden Ressourcen:

- Verwenden von Pod Identity mit Amazon EKS
- AWS-Secrets-Store-CSI-Treiber auf GitHub

Verwenden Sie AWS Secrets and Configuration Provider CSI mit Pod Identity für Amazon EKS

Die AWS Secrets and Configuration Provider-Integration mit dem Pod Identity Agent für Amazon Elastic Kubernetes Service bietet verbesserte Sicherheit, vereinfachte Konfiguration und verbesserte Leistung für Anwendungen, die auf Amazon EKS ausgeführt werden. Pod Identity vereinfacht die AWS Identity and Access Management (IAM) -Authentifizierung für Amazon EKS beim Abrufen von Parametern AWS Systems Manager Parameter Store oder Geheimnissen aus Secrets Manager.

Amazon EKS Pod Identity optimiert die Konfiguration von IAM-Berechtigungen für Kubernetes-Anwendungen, da Berechtigungen direkt über Amazon-EKS-Schnittstellen eingerichtet werden können. Das reduziert die Anzahl der Schritte und der Wechsel zwischen Amazon EKS und IAM-Services entfällt. Pod Identity ermöglicht die Verwendung einer einzigen IAM-Rolle in mehreren Clustern, ohne dass die Vertrauensrichtlinien aktualisiert werden müssen, und unterstützt Rollensitzungs-Tags für eine detailliertere Zugriffskontrolle. Dieser Ansatz vereinfacht nicht nur die Richtlinienverwaltung, indem er die rollenübergreifende Wiederverwendung von Berechtigungsrichtlinien ermöglicht, sondern erhöht auch die Sicherheit, indem der Zugriff auf AWS Ressourcen auf der Grundlage übereinstimmender Tags ermöglicht wird.

Funktionsweise

1. Pod Identity weist dem Pod eine IAM-Rolle zu.
2. ASCP verwendet diese Rolle zur Authentifizierung bei. AWS-Services
3. Wenn ASCP entsprechend autorisiert ist, ruft es die angeforderten Parameter ab und stellt sie dem Pod zur Verfügung.

Weitere Informationen finden Sie im Benutzerhandbuch für Amazon EKS unter Funktionsweise von Amazon EKS Pod Identity.

Voraussetzungen

> **Important**
>
> Pod Identity wird nur für Amazon EKS in der Cloud unterstützt. Es wird nicht für Amazon EKS Anywhere oder selbstverwaltete Kubernetes-Cluster auf Amazon-Instances unterstützt. Red Hat OpenShift Service in AWS EC2

- Amazon-EKS-Cluster (Version 1.24 oder höher)
- Zugriff auf AWS CLI einen Amazon EKS-Cluster über `kubectl`
- (Optional) Zugriff auf zwei AWS-Konten für kontoübergreifenden Zugriff

Installieren des Pod-Identity-Agents für Amazon EKS

Um Pod Identity in Ihrem Cluster zu verwenden, müssen Sie das Add-on „Amazon EKS Pod Identity Agent“ installieren.

Installieren des Pod-Identity-Agents

- Installieren Sie das Add-on „Pod-Identity-Agent“ in Ihrem Cluster.

  Ersetzen Sie die *`default placeholder text`* durch Ihre eigenen Werte:

  ```
  eksctl create addon \
    --name eks-pod-identity-agent \
    --cluster clusterName \
    --region region
  ```

Einrichten von ASCP mit Pod Identity

1. Erstellen Sie eine Berechtigungsrichtlinie, die den Parametern, auf die der Pod zugreifen muss, die Berechtigungen `ssm:GetParameters` und `ssm:DescribeParameters` gewährt.
2. Erstellen Sie eine IAM-Rolle, die vom Amazon-EKS-Service-Prinzipalen für Pod Identity übernommen werden kann:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "pods.eks.amazonaws.com"
        },
        "Action": [
          "sts:AssumeRole",
          "sts:TagSession"
        ]
      }
    ]
  }
```

Fügen Sie die IAM-Richtlinie an die Rolle an.

Ersetze die *default placeholder text* durch deine eigenen Werte:

```
aws iam attach-role-policy \
  --role-name MY_ROLE \
  --policy-arn POLICY_ARN
```

3. Erstellen Sie eine Pod-Identity-Zuordnung. Ein Beispiel finden Sie unter Erstellen einer Pod-Identity-Zuordnung im Benutzerhandbuch für Amazon EKS
4. Erstellen Sie die `SecretProviderClass`, die angibt, welche Parameter oder Secrets im Pod gemountet werden sollen:

```
kubectl apply -f kubectl apply -f https://raw.githubusercontent.com/aws/
secrets-store-csi-driver-provider-aws/main/examples/ExampleSecretProviderClass-
PodIdentity.yaml
```

Der Hauptunterschied in `SecretProviderClass` zwischen IRSA und Pod Identity ist der optionale Parameter `usePodIdentity`. Es ist ein optionales Feld, das den

Authentifizierungsansatz bestimmt. Wenn nicht angegeben, wird standardmäßig IAM Roles for Service Accounts (IRSA) verwendet.

- Verwenden Sie einen der folgenden Werte, um EKS Pod Identity zu verwenden: `"true", "True", "TRUE", "t", "T"`.
- Wenn Sie explizit IRSA nutzen möchten, verwenden Sie einen der folgenden Werte: `"false", "False", "FALSE", "f", or "F"`.

5. Stellen Sie den Pod, der die Parameter oder Secrets mountet, unter `/mnt/secrets-store` bereit:

```
kubectl apply -f kubectl apply -f https://raw.githubusercontent.com/aws/secrets-store-csi-driver-provider-aws/main/examples/ExampleDeployment-PodIdentity.yaml
```

6. Wenn Sie einen privaten Amazon EKS-Cluster verwenden, stellen Sie sicher, dass die VPC, in der sich der Cluster befindet, über einen AWS STS Endpunkt verfügt. Weitere Informationen zum Erstellen eines Endpunktes finden Sie unter Schnittstellen-VPC-Endpunkte im Benutzerhandbuch für AWS Identity and Access Management .

Überprüfen des Secret-Mountings

Führen Sie den folgenden Befehl aus, um zu überprüfen, dass der Parameter oder das Secret korrekt gemountet wurde.

Ersetzen Sie die *`default placeholder text`* durch Ihre eigenen Werte:

```
kubectl exec -it $(kubectl get pods | awk '/pod-identity-deployment/{print $1}' | head -1) -- cat /mnt/secrets-store/MyParameter
```

Richten Sie Amazon EKS Pod Identity für den Zugriff auf Parameter in Parameter Store wie folgt ein:

1. Erstellen Sie eine Berechtigungsrichtlinie, die den Parametern, auf die der Pod zugreifen muss, die Berechtigungen `ssm:GetParameters` und `ssm:DescribeParameters` gewährt.
2. Erstellen Sie einen Parameter in Parameter Store, sofern noch nicht geschehen. Weitere Informationen finden Sie unter Parameter Store-Parameter im Systems Manager erstellen.

Fehlerbehebung

Sie können die meisten Fehler anzeigen, indem Sie die Pod-Bereitstellung beschreiben.

Fehlermeldungen für Ihren Container anzeigen

1. Rufen Sie mit dem folgenden Befehl eine Liste der Pod-Namen ab. Wenn Sie nicht den Standard-Namespace verwenden, verwenden Sie `-n` *`namespace`*.

   ```
   kubectl get pods
   ```

2. *`pod-id`*Verwenden Sie zur Beschreibung des Pods im folgenden Befehl die Pod-ID der Pods, die Sie im vorherigen Schritt gefunden haben. Wenn Sie nicht den Standard-Namespace verwenden, verwenden Sie `-n` *`NAMESPACE`*.

   ```
   kubectl describe pod/pod-id
   ```

Fehler für den ASCP anzeigen

- Um weitere Informationen in den Anbieterprotokollen zu finden, *`PODID`* verwenden Sie im folgenden Befehl die ID des Pods csi-secrets-store-provider-aws.

  ```
  kubectl -n kube-system get pods
  kubectl -n kube-system logs pod/pod-id
  ```

Verwenden Sie AWS Secrets and Configuration Provider CSI mit IAM Roles for Service Accounts (IRSA)

Themen

- Voraussetzungen
- Einrichten der Zugriffssteuerung
- Identifizieren der Parameter, die gemountet werden sollen
- Fehlerbehebung

Voraussetzungen

- Amazon-EKS-Cluster (Version 1.17 oder höher)
- Zugriff auf die AWS CLI und einen Amazon-EKS-Cluster über `kubectl`

Einrichten der Zugriffssteuerung

ASCP ruft die Amazon-EKS-Pod-Identität ab und tauscht sie gegen eine IAM-Rolle. Sie legen in einer IAM-Richtlinie Berechtigungen für diese IAM-Rolle fest. Wenn ASCP die IAM-Rolle übernimmt, erhält der Service Zugriff auf die von Ihnen autorisierten Parameter. Andere Container können nur auf die Parameter zugreifen, wenn Sie diese auch der IAM-Rolle zuordnen.

Gewähren Sie Ihrem Amazon-EKS-Pod wie folgt Zugriff auf Parameter in Parameter Store:

1. Erstellen Sie eine Berechtigungsrichtlinie, die den Parametern, auf die der Pod zugreifen muss, die Berechtigungen `ssm:GetParameters` und `ssm:DescribeParameters` gewährt.
2. Erstellen Sie einen IAM OpenID Connect (OIDC)-Anbieter für den Cluster, wenn Sie noch keinen haben. Weitere Informationen finden Sie unter Erstellen eines IAM-OIDC-Anbieters für Ihren Cluster im Benutzerhandbuch für Amazon EKS.
3. Erstellen Sie eine IAM-Rolle für das Servicekonto und fügen Sie die Richtlinie an. Weitere Informationen finden Sie unter Erstellen einer IAM-Rolle für ein Servicekonto im Benutzerhandbuch für Amazon EKS.
4. Wenn Sie einen privaten Amazon-EKS-Cluster verwenden, stellen Sie sicher, dass die VPC, in der sich der Cluster befindet, über einen AWS STS-Endpunkt verfügt. Weitere Informationen zum Erstellen eines Endpunktes finden Sie unter Schnittstellen-VPC-Endpunkte im Benutzerhandbuch für AWS Identity and Access Management.

Identifizieren der Parameter, die gemountet werden sollen

Um zu bestimmen, welche Parameter ASCP in Amazon EKS als Dateien im Dateisystem mountet, erstellen Sie eine the section called “SecretProviderClass”-YAML-Datei. Die `SecretProviderClass` listet die zu mountenden Parameter und den Dateinamen auf, unter dem sie gemountet werden sollen. `SecretProviderClass` muss sich im gleichen Namespace wie der Amazon-EKS-Pod befinden, auf den verwiesen wird.

Mounten der Parameter als Dateien

In der folgenden Anleitung wird gezeigt, wie Sie Parameter mithilfe der YAML-Beispieldateien ExampleSecretProviderClass.yaml und ExampleDeployment.yaml als Dateien mounten.

Mounten Sie Parameter in Amazon EKS wie folgt:

1. Wenden Sie die `SecretProviderClass` auf den Pod an:

```
kubectl apply -f ExampleSecretProviderClass.yaml
```

2. Stellen Sie den Pod bereit:

```
kubectl apply -f ExampleDeployment.yaml
```

3. ASCP mountet die Dateien.

Fehlerbehebung

Sie können die meisten Fehler anzeigen, indem Sie die Pod-Bereitstellung beschreiben.

Fehlermeldungen für Ihren Container anzeigen

1. Rufen Sie mit dem folgenden Befehl eine Liste der Pod-Namen ab. Wenn Sie nicht den Standard-Namespace verwenden, verwenden Sie `-n` *`name-space`*.

```
kubectl get pods
```

2. Um den Pod zu beschreiben, geben Sie im folgenden Befehl für *`pod-id`* die Pod-ID aus den Pods an, die Sie im vorherigen Schritt gefunden haben. Wenn Sie nicht den Standard-Namespace verwenden, verwenden Sie `-n` *`nameSpace`*.

```
kubectl describe pod/pod-id
```

Fehler für den ASCP anzeigen

- Um weitere Informationen in den Anbieterprotokollen zu finden, verwenden Sie im folgenden Befehl für*`pod-id`* die ID des Pods csi-secrets-store-provider-aws.

```
kubectl -n kube-system get pods
kubectl -n kube-system logs Pod/pod-id
```

- Überprüfen Sie, ob die **`SecretProviderClass`**-CRD installiert wurde:

```
kubectl get crd secretproviderclasses.secrets-store.csi.x-k8s.io
```

Dieser Befehl muss Informationen zur benutzerdefinierten Ressourcendefinition von `SecretProviderClass` zurückgeben.

- Stellen Sie sicher, dass das SecretProviderClass-Objekt erstellt wurde.

```
kubectl get secretproviderclass SecretProviderClassName -o yaml
```

AWS Codebeispiele für Secrets und Configuration Provider

Beispiele für ASCP-Authentifizierung und Zugriffskontrolle

Beispiel: IAM-Richtlinie, die dem Service Amazon EKS Pod Identity (pods.eks.amazonaws.com) ermöglicht, die Rolle zu übernehmen und die Sitzung zu taggen:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pods.eks.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ]
    }
  ]
}
```

SecretProviderClass

Sie verwenden YAML, um zu beschreiben, welche Parameter mithilfe von ASCP in Amazon EKS gemountet werden sollen. Beispiele finden Sie unter the section called “SecretProviderClass-Nutzung”.

SecretProviderClass YAML-Struktur

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
   name: name
spec:
  provider: aws
  parameters:
    region:
    failoverRegion:
    pathTranslation:
    usePodIdentity:
    preferredAddressType:
    objects:
```

Das Feld Parameter enthält die Details der Mounting-Anfrage:

Region

(Optional) Der Wert AWS-Region des Parameters. Wenn Sie dieses Feld nicht verwenden, sucht der ASCP die Region aus der Anmerkung auf dem Knoten. Diese Suche steigert den Overhead von Mounting-Anfragen. Daher wird empfohlen, die Region für Cluster mit einer großen Anzahl von Pods anzugeben.

Wenn Sie auch `failoverRegion` angeben, versucht ASCP, den Parameter aus beiden Regionen abzurufen. Wenn eine Region einen `4xx`-Fehler zurückgibt, z. B. aufgrund eines Authentifizierungsproblems, mountet ASCP keinen der Parameter. Wenn der Parameter erfolgreich von `region` abgerufen wurde, mountet ASCP diesen Parameterwert. Wenn das Secret nicht erfolgreich von `region` abgerufen wurde, aber erfolgreich von `failoverRegion` abgerufen werden konnte, mountet ASCP diesen Parameter-Wert.

`failoverRegion`

(Optional) Wenn Sie dieses Feld angeben, versucht ASCP, das Secret aus den Regionen abzurufen, die in `region` und diesem Feld definiert sind. Wenn eine Region einen `4xx`-Fehler zurückgibt, z. B. aufgrund eines Authentifizierungsproblems, mountet ASCP keinen der Parameter. Wenn der Parameter erfolgreich von `region` abgerufen wurde, mountet ASCP diesen Parameterwert. Wenn das Secret nicht erfolgreich von `region` abgerufen wurde, aber erfolgreich von `failoverRegion` abgerufen werden konnte, mountet ASCP diesen Parameter-Wert. Ein Beispiel für die Nutzung dieses Felds finden Sie unter Multiregionaler Parameter-Failover.

pathTranslation (Pfadangabe)

(Optional) Ein einzelnes Ersetzungszeichen, das verwendet werden soll, wenn der Dateiname in Amazon EKS das Pfadtrennzeichen enthält, z. B. Schrägstrich (/) unter Linux. ASCP kann keine gemountete Datei erstellen, die ein Pfadtrennzeichen enthält. Stattdessen ersetzt ASCP das Pfadtrennzeichen durch ein anderes Zeichen. Wenn Sie dieses Feld nicht verwenden, ist das Ersatzzeichen ein Unterstrich (_), d. h. `My/Path/Parameter` wird als `My_Path_Parameter` gemountet.

Um die Zeichenersetzung zu verhindern, geben Sie die Zeichenfolge `False` ein.

usePodIdentity

(Optional) Legt den Authentifizierungsansatz fest. Wenn nicht angegeben, wird standardmäßig IAM Roles for Service Accounts (IRSA) verwendet.

- Verwenden Sie einen der folgenden Werte, um EKS Pod Identity zu verwenden: `"true"`, `"True"`, `"TRUE"`, `"t"` oder `"T"`.
- Wenn Sie explizit IRSA nutzen möchten, verwenden Sie einen der folgenden Werte: `"false"`, `"False"`, `"FALSE"`, `"f"` oder `"F"`.

preferredAddressType

(Optional) Gibt den bevorzugten IP-Adresstyp für die Pod-Identity-Agent-Endpunktkommunikation an. Das Feld ist nur bei Verwendung der EKS-Pod-Identity-Funktion relevant und wird ignoriert, wenn IAM-Roles for Service Accounts verwendet wird. Bei den Werten wird zwischen Groß- und Kleinschreibung unterschieden. Folgende sind gültige Werte:

- `"ipv4"`, `"IPv4"` „, oder `"IPV4"` — Erzwingt die Verwendung des Pod Identity IPv4 Agent-Endpunkts
- `"ipv6"`,`"IPv6"`, oder `"IPV6"` — Erzwingen Sie die Verwendung des Pod Identity IPv6 Agent-Endpunkts
- nicht spezifiziert — auto Endpunktauswahl verwenden, zuerst den IPv4 Endpunkt ausprobieren und bei einem Fehler auf den IPv6 Endpunkt IPv4 zurückgreifen

objects (Objekte)

Eine Zeichenfolge, die eine YAML-Deklaration der bereitzustellenden Secrets enthält. Wir empfehlen, eine mehrzeilige YAML-Zeichenfolge oder ein Pipe-Zeichen (|) zu verwenden.

objectName (Objektname)

Erforderlich Gibt den Namen des Parameters oder Secrets an, das abgerufen werden soll. Für Parameter Store ist dies der `Name` des Parameters und kann entweder der Name oder der vollständige ARN sein. Für Secrets Manager ist dies der Parameter `SecretId` und kann entweder der Anzeigename oder der vollständige ARN des Secrets sein.

objectType

Erforderlich, wenn Sie keinen Secrets Manager ARN für `objectName` verwenden. Geben Sie als Parameter Store `ssmparameter`ein. Verwenden Sie für Secrets Manager `secretsmanager`.

objectAlias

(Optional) Der Dateiname des Secrets im Amazon-EKS-Pod. Wenn Sie dieses Feld nicht angeben, wird `objectName` als Dateiname angezeigt.

objectVersion (Objektversion)

(Optional) Die Versions-ID des Parameters. Nicht empfohlen, da Sie jedes Mal, wenn Sie den Parameter aktualisieren, die Versions-ID aktualisieren müssen. Standardmäßig wird die neueste Version verwendet. Wenn Sie eine `failoverRegion` angeben, stellt dieses Feld den primären `objectVersion` dar.

objectVersionLabel

(Optional) Der Alias für die Version. Die Standardversion ist die neueste Version`AWSCURRENT`. Wenn Sie eine `failoverRegion` angeben, stellt dieses Feld den primären `objectVersionLabel` dar.

jmesPath

(Optional) Eine Zuordnung der Schlüssel im Parameter zu den Dateien, die in Amazon EKS gemountet werden sollen. Um dieses Feld zu verwenden, muss Ihr Parameterwert im JSON-Format vorliegen.

Das folgende Beispiel zeigt, wie ein JSON-kodierter Parameter aussieht.

```
{
    "username" : "myusername",
    "password" : "mypassword"
```

```
}
```

Die Schlüssel sind `username` und `password`. Der Wert, der mit `username` verbunden ist, ist `myusername`, und der Wert, der mit `password` verbunden ist, ist `mypassword`.

Wenn Sie dieses Feld verwenden, müssen Sie die Unterfelder `path` und `objectAlias` angeben.

Pfad

Ein Schlüssel aus einem Schlüssel-Wert-Paar im JSON des Parameterwerts. Wenn das Feld einen Bindestrich enthält, verwenden Sie einfache Anführungszeichen als Escape-Zeichen. Beispiel: `path: '"hyphenated-path"'`

objectAlias

Der Dateiname, der im Amazon-EKS-Pod gemountet werden soll. Wenn das Feld einen Bindestrich enthält, verwenden Sie einfache Anführungszeichen als Escape-Zeichen. Beispiel: `objectAlias: '"hyphenated-alias"'`

`failoverObject`

(Optional) Wenn Sie dieses Feld angeben, versucht ASCP, den im primären `objectName` angegebenen Parameter und den im `failoverObject-objectName`-Unterfeld angegebenen Parameter abzurufen. Wenn einer der Parameter einen `4xx`-Fehler zurückgibt, z. B. aufgrund eines Authentifizierungsproblems, mountet ASCP keinen der Parameter. Wenn der Parameter erfolgreich vom primären `objectName` abgerufen wurde, mountet ASCP diesen Parameterwert. Wenn der Parameter nicht erfolgreich vom primären `objectName` abgerufen wurde, aber erfolgreich vom Failover-`objectName` abgerufen werden konnte, mountet ASCP diesen Parameterwert. Wenn Sie dieses Feld angeben, müssen Sie auch das Feld `objectAlias` angeben. Ein Beispiel für die Nutzung dieses Felds finden Sie unter Failover auf einen anderen Parameter.

In der Regel verwenden Sie dieses Feld, wenn es sich bei dem Failover-Parameter nicht um ein Replikat handelt. Ein Beispiel dazu, wie Sie ein Replikat angeben, finden Sie unter Multiregionaler Parameter-Failover.

objectName (Objektname)

Der Name oder vollständige ARN des Failover-Parameters. Wenn Sie einen ARN verwenden, muss die Region im ARN mit dem Feld `failoverRegion` übereinstimmen.

objectVersion (Objektversion)

(Optional) Die Versions-ID des Parameters. Muss mit der primären `objectVersion` übereinstimmen. Nicht empfohlen, da Sie jedes Mal, wenn Sie den Parameter aktualisieren, die Versions-ID aktualisieren müssen. Standardmäßig wird die neueste Version verwendet.

objectVersionLabel

(Optional) Der Alias für die Version. Die Standardversion ist die neueste Version`AWSCURRENT`.

Erstellen Sie eine SecretProviderClass Basiskonfiguration, um Parameter in Ihren Amazon EKS-Pods zu mounten.

Pod Identity

SecretProviderClass um einen Parameter im selben Amazon EKS-Cluster zu verwenden:

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-parameter-store
spec:
  provider: aws
  parameters:
    objects: |
      - objectName: "MyParameter"
        objectType: "ssmparameter"
    usePodIdentity: "true"
```

IRSA

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: deployment-aws-parameter
spec:
  provider: aws
  parameters:
    objects: |
        - objectName: "MyParameter"
```

```
            objectType: "ssmparameter"
```

SecretProviderClass-Nutzung

Verwenden Sie diese Beispiele, um `SecretProviderClass`-Konfigurationen für verschiedene Szenarien zu erstellen.

Beispiel: Parameter nach Namen oder ARN mounten

Dieses Beispiel zeigt, wie drei verschiedene Parametertypen gemountet werden:

- Ein Parameter, der anhand des vollständigen ARN angegeben wird
- Ein Parameter, der anhand des Namens angegeben wird
- Eine Parameterversion eines Secrets

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-parameters
spec:
  provider: aws
  parameters:
    objects: |
      - objectName: "arn:aws:ssm:us-east-2:777788889999:parameter:MyParameter2-d4e5f6"
      - objectName: "MyParameter3"
        objectType: "ssmparameter"
      - objectName: "MyParameter4"
        objectType: "ssmparameter"
        objectVersionLabel: "AWSCURRENT"
```

Beispiel: Schlüssel-Wert-Paare aus einem Parameter mounten

Dieses Beispiel zeigt, wie bestimmte Schlüssel-Wert-Paare aus einem Parameter im JSON-Format gemountet werden:

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-parameters
spec:
```

```
  provider: aws
  parameters:
    objects: |
      - objectName: "arn:aws:ssm:us-east-2:777788889999:parameter:MyParameter-a1b2c3"
        jmesPath:
            - path: username
              objectAlias: dbusername
            - path: password
              objectAlias: dbpassword
```

Beispiel: verschiedene Failover-Konfigurationen

Diese Beispiele veranschaulichen, wie Failover für Parameter konfiguriert wird.

Multiregionaler Parameter-Failover

Dieses Beispiel zeigt, wie Sie einen automatischen Failover für einen Parameter konfigurieren, der über mehrere Regionen repliziert wird:

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-parameters
spec:
  provider: aws
  parameters:
    region: us-east-1
    failoverRegion: us-east-2
    objects: |
      - objectName: "MyParameter"
```

Failover auf einen anderen Parameter

Dieses Beispiel zeigt, wie Sie einen Failover auf einen anderen Parameter (kein Replikat) konfigurieren:

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-parameters
spec:
  provider: aws
  parameters:
```

```
    region: us-east-1
    failoverRegion: us-east-2
    objects: |
      - objectName: "arn:aws:ssm:us-east-1:777788889999:parameter:MyParameter-a1b2c3"
        objectAlias: "MyMountedParameter"
        failoverObject:
          - objectName: "arn:aws:ssm:us-
east-2:777788889999:parameter:MyFailoverParameter-d4e5f6"
```

Weitere Ressourcen

Weitere Informationen zur Verwendung von ASCP mit Amazon EKS finden Sie in den folgenden Ressourcen:

- Verwenden von Pod Identity mit Amazon EKS
- AWS Secrets Store CSI-Treiber auf GitHub

Verwenden von Parameter Store-Parametern in AWS Lambda -Funktionen

Parameter Store, ein Tool in AWS Systems Manager, bietet sicheren, hierarchischen Speicher für die Verwaltung von Konfigurationsdaten und Geheimnissen. Sie können Daten wie Kennwörter, Datenbankzeichenfolgen Amazon Machine Image (AMI) IDs und Lizenzcodes als Parameterwerte speichern.

Um Parameter aus Parameter Store AWS Lambda Funktionen zu verwenden, ohne ein SDK zu verwenden, können Sie die Lambda-Erweiterung AWS Parameters and Secrets verwenden. Diese Erweiterung ruft Parameterwerte ab und speichert sie zur späteren Verwendung. Durch die Verwendung der Lambda-Erweiterung können Sie Ihre Kosten senken, indem Sie die Anzahl der API-Aufrufe auf Parameter Store reduzieren. Durch die Verwendung der Erweiterung kann auch die Latenzzeit verbessert werden, da der Abruf eines zwischengespeicherten Parameters schneller ist als der Abruf von Parameter Store.

Eine Lambda-Erweiterung ist ein begleitender Prozess, der die Fähigkeiten einer Lambda-Funktion erweitert. Eine Erweiterung ist wie ein Client, der parallel zu einem Lambda-Aufruf ausgeführt wird. Dieser parallele Client kann jederzeit während seines Lebenszyklus mit Ihrer Funktion verbunden werden. Weitere Informationen zu Lambda-Erweiterungen finden Sie unter Lambda-Erweiterungs-API im AWS Lambda -Entwicklerhandbuch.

Die Lambda-Erweiterung AWS Parameters and Secrets funktioniert Parameter Store sowohl AWS Secrets Manager für als auch. Informationen zur Verwendung der Lambda-Erweiterung mit

Geheimnissen aus Secrets Manager finden Sie unter Verwenden von AWS Secrets Manager Geheimnissen in AWS Lambda Funktionen im AWS Secrets Manager Benutzerhandbuch.

Verwandte Informationen

Verwenden der Lambda-Erweiterung AWS Parameter and Secrets zum Zwischenspeichern von Parametern und Geheimnissen (AWS Compute Blog)

So funktioniert die Erweiterung

Um Parameter in einer Lambda-Funktion ohne die Lambda-Erweiterung zu verwenden, müssen Sie Ihre Lambda-Funktion so konfigurieren, dass sie Konfigurationsaktualisierungen erhält, indem Sie sie in die `GetParameter`-API-Aktion für Parameter Store integrieren.

Wenn Sie die Lambda-Erweiterung AWS Parameters and Secrets verwenden, ruft die Erweiterung den Parameterwert ab Parameter Store und speichert ihn im lokalen Cache. Dann wird der zwischengespeicherte Wert für weitere Aufrufe verwendet, bis er abläuft. Zwischengespeicherte Werte laufen ab, nachdem sie ihre time-to-live (TTL) überschritten haben. Sie können den TTL-Wert mithilfe der Umgebungsvariablen `SSM_PARAMETER_STORE_TTL` konfigurieren, wie weiter unten in diesem Thema erläutert.

Wenn die konfigurierte Cache-TTL nicht abgelaufen ist, wird der zwischengespeicherte Parameterwert verwendet. Wenn die Zeit abgelaufen ist, wird der zwischengespeicherte Wert ungültig und der Parameterwert wird von Parameter Store abgerufen.

Außerdem erkennt das System Parameterwerte, die häufig verwendet werden, und behält sie im Cache bei, während abgelaufene oder nicht verwendete Werte gelöscht werden.

> **Important**
>
> Die Erweiterung kann nur in der `INVOKE`-Phase des Lambda-Vorgangs und nicht während der `INIT`-Phase aufgerufen werden.

Implementierungsinformationen

Verwenden Sie die folgenden Details, um Ihnen bei der Konfiguration der Lambda-Erweiterung AWS Parameters and Secrets zu helfen.

Authentifizierung

Um Parameter Store-Anfragen zu autorisieren und zu authentifizieren, verwendet die Erweiterung dieselben Anmeldeinformationen wie diejenigen, die zum Ausführen der Lambda-Funktion verwendet werden. Daher muss die AWS Identity and Access Management (IAM) -Rolle, mit der die Funktion ausgeführt wird, über die folgenden Berechtigungen für die Interaktion verfügen: Parameter Store

- `ssm:GetParameter` – Erforderlich, um Parameter von Parameter Store abzurufen
- `kms:Decrypt` – Erforderlich, wenn Sie `SecureString`-Parameter von Parameter Store abrufen

Weitere Informationen finden Sie unter AWS Lambda -Ausführungsrolle im AWS Lambda -Entwicklerhandbuch.

Instanziierung

Lambda instanziiert separate Instances, die der Gleichzeitigkeitsstufe entsprechen, die Ihre Funktion benötigt. Jede Instance ist isoliert und verwaltet ihren eigenen lokalen Cache Ihrer Konfigurationsdaten. Weitere Informationen über Lambda-Instances und Gleichzeitigkeit finden Sie unter Konfigurieren der reservierten Währung im AWS Lambda -Entwicklerhandbuch.

Keine SDK-Abhängigkeit

Die Lambda-Erweiterung AWS Parameters and Secrets funktioniert unabhängig von jeder AWS SDK-Sprachbibliothek. Ein AWS SDK ist nicht erforderlich, um GET-Anfragen an zu Parameter Store stellen.

Localhost-Port

Verwenden Sie `localhost` in Ihren GET-Anfragen. Die Erweiterung stellt Anfragen an den localhost-Port 2773. Sie müssen keinen externen oder internen Endpunkt angeben, um die Erweiterung zu verwenden. Sie können den Port konfigurieren, indem Sie die Umgebungsvariable auf `PARAMETERS_SECRETS_EXTENSION_HTTP_PORT` setzen.

In Python könnte Ihre GET-URL beispielsweise wie im folgenden Beispiel aussehen.

```
parameter_url = ('http://localhost:' + port + '/systemsmanager/parameters/get/?
name=' + ssm_parameter_path)
```

Änderungen an einem Parameterwert, bevor TTL abläuft

Die Erweiterung erkennt keine Änderungen am Parameterwert und führt keine automatische Aktualisierung durch, bevor die TTL abläuft. Wenn Sie einen Parameterwert ändern, schlagen Vorgänge, die den zwischengespeicherten Parameterwert verwenden, möglicherweise fehl, bis der Cache das nächste Mal aktualisiert wird. Wenn Sie häufige Änderungen an einem Parameterwert erwarten, empfehlen wir Ihnen, einen kürzeren TTL-Wert einzustellen.

Anfordern eines Headers

Um Parameter aus dem Erweiterungs-Cache abzurufen, muss der Header Ihrer GET-Anfrage eine `X-Aws-Parameters-Secrets-Token`-Referenz enthalten. Setzen Sie das Token auf `AWS_SESSION_TOKEN`, das von Lambda für alle laufenden Funktionen bereitgestellt wird. Die Verwendung dieses Headers zeigt an, dass sich der Anrufer in der Lambda-Umgebung befindet.

Beispiel

Das folgende Beispiel in Python demonstriert eine einfache Anfrage zum Abrufen des Wertes eines zwischengespeicherten Parameters.

```
import urllib.request
import os
import json

aws_session_token = os.environ.get('AWS_SESSION_TOKEN')

def lambda_handler(event, context):
    # Retrieve /my/parameter from Parameter Store using extension cache
    req = urllib.request.Request('http://localhost:2773/systemsmanager/parameters/
get?name=%2Fmy%2Fparameter')
    req.add_header('X-Aws-Parameters-Secrets-Token', aws_session_token)
    config = urllib.request.urlopen(req).read()

    return json.loads(config)
```

ARM-Unterstützung

Die Erweiterung unterstützt die ARM-Architektur in den meisten AWS-Regionen , in denen die x86_64- und x86-Architekturen unterstützt werden. Wenn Sie die ARM-Architektur verwenden, empfehlen wir Ihnen, zu überprüfen, ob Ihre Architektur unterstützt wird. Vollständige Listen der Erweiterungen finden ARNs Sie unterAWS Lambda-Erweiterung für Parameter und Geheimnisse ARNs.

Protokollierung

Lambda protokolliert Ausführungsinformationen über die Erweiterung zusammen mit der Funktion mithilfe von Amazon CloudWatch Logs. Standardmäßig protokolliert die Erweiterung eine minimale Menge an CloudWatch Informationen unter. Um weitere Details zu protokollieren, setzen Sie die Umgebungsvariable `PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL` auf `DEBUG`.

Hinzufügen der Erweiterung zu einer Lambda-Funktion

Um die Lambda-Erweiterung AWS Parameters and Secrets zu verwenden, fügen Sie die Erweiterung als Ebene zu Ihrer Lambda-Funktion hinzu.

Verwenden Sie eine der folgenden Methoden, um die Erweiterung zu Ihrer Funktion hinzuzufügen.

AWS-Managementkonsole (Option „Ebene hinzufügen“)

1. Öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.
2. Wählen Sie Ihre Funktion. Wählen Sie im Bereich Layers (Ebenen) die Option Add a layer (Ebene hinzufügen) aus.
3. Wählen Sie im Bereich Eine Ebene auswählen die Option AWS -Ebenen aus.
4. Wählen Sie für AWS -Ebenen AWS-Parameter und Secrets-Lambda-Erweiterung aus, wählen Sie eine Version und wählen Sie anschließend Hinzufügen aus.

AWS-Managementkonsole (ARN-Option angeben)

1. Öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.
2. Wählen Sie Ihre Funktion. Wählen Sie im Bereich Layers (Ebenen) die Option Add a layer (Ebene hinzufügen) aus.
3. Wählen Sie im Bereich Choose a layer (Ebene auswählen) die Option Specify an ARN (ARN angeben) aus.
4. Geben Sie für Specify an ARN die Erweiterung ARN für Ihre AWS-Region und Architektur ein, und wählen Sie dann Hinzufügen aus.

AWS Command Line Interface

Führen Sie in der AWS CLI den folgenden aus. Ersetzen Sie jeden *example resource placeholder* durch Ihre Informationen.

```
aws lambda update-function-configuration \
    --function-name function-name \
```

```
    --layers layer-ARN
```

Ähnliche Informationen

[Verwenden von Ebenen mit Ihrer Lambda-Funktion]

[Konfigurieren von Erweiterungen (ZIP-Dateiarchiv)]

AWS Parameter und Geheimnisse Umgebungsvariablen der Lambda-Erweiterung

Sie können die Erweiterung konfigurieren, indem Sie die folgenden Umgebungsvariablen ändern. Um die aktuellen Einstellungen zu sehen, setzen Sie `PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL` auf `DEBUG`. Weitere Informationen finden Sie unter [Verwenden von AWS Lambda Umgebungsvariablen] im AWS Lambda Entwicklerhandbuch.

> **Note**
>
> AWS Lambda zeichnet Betriebsdetails zur Lambda-Erweiterung und Lambda-Funktion in Amazon CloudWatch Logs auf.

Umgebungsvariable	Details	Erforderlich	Zulässige Werte	Standardwert
`SSM_PARAMETER_STORE_TIMEOUT_MILLIS`	Timeout in Millisekunden für Anfragen an Parameter Store. Ein Wert von 0 (null) gibt an, dass kein Timeout vorliegt.	Nein	Alle ganzen Zahlen	0 (Null)
`SECRETS_MANAGER_TI`	Timeout in Millisekunden für Anfragen	Nein	Alle ganzen Zahlen	0 (Null)

Umgebungsvariable	Details	Erforderlich	Zulässige Werte	Standardwert
MEOUT_MILLIS	an Secrets Manager. Ein Wert von 0 (null) gibt an, dass kein Timeout vorliegt.			
SSM_PARAMETER_STORE_TTL	Maximal gültige Lebensdauer eines Parameters im Cache in Sekunden, bevor er ungültig wird. Ein Wert von 0 (Null) gibt an, dass der Cache umgangen werden soll. Diese Variable wird ignoriert, wenn der Wert für PARAMETERS_SECRETS_EXTENSION_CACHE_SIZE 0 (Null) ist.	Nein	0 (Null) bis 300 Sek. (Fünf Minuten)	300 Sek. (Fünf Minuten)

Umgebungsvariable	Details	Erforderlich	Zulässige Werte	Standardwert
SECRETS_MANAGER_TTL	Maximal gültige Lebensdauer eines Secrets im Cache in Sekunden, bevor es ungültig wird. Ein Wert von 0 (Null) gibt an, dass der Cache umgangen wurde. Diese Variable wird ignoriert, wenn der Wert für PARAMETERS_SECRETS_EXTENSION_CACHE_SIZE 0 (Null) ist.	Nein	0 (Null) bis 300 Sek. (Fünf Minuten)	300 Sek. (5 Minuten)
PARAMETERS_SECRETS_EXTENSION_CACHE_ENABLED	Bestimmt, ob der Cache für die Erweiterung aktiviert ist. Gültige Werte: TRUE | FALSE	Nein	TRUE | FALSE	TRUE

Umgebungsvariable	Details	Erforderlich	Zulässige Werte	Standardwert
PARAMETERS_SECRETS_EXTENSION_CACHE_SIZE	Die maximale Größe des Caches in Bezug auf die Anzahl der Elemente. Ein Wert von 0 (Null) gibt an, dass der Cache umgangen wurde. Diese Variable wird ignoriert, wenn beide Cache-TTL-Werte 0 (Null) sind.	Nein	0 (Null) bis 1 000	1000
PARAMETERS_SECRETS_EXTENSION_HTTP_PORT	Der Port für den lokalen HTTP-Server.	Nein	1–65 535	2773

Umgebungsvariable	Details	Erforderlich	Zulässige Werte	Standardwert
`PARAMETERS_SECRETS_EXTENSION_MAX_CONNECTIONS`	Maximale Anzahl von Verbindungen für die HTTP-Clients, die die Erweiterung verwendet, um Anfragen an Parameter Store oder Secrets Manager zu stellen. Dies ist eine Konfiguration pro Client für die Anzahl der Verbindungen, die sowohl der Secrets-Manager-Client als auch der Parameter Store-Client zu den Backend-Diensten herstellen.	Nein	Mindestens 1; Keine Höchstgrenze.	3

Umgebungsvariable	Details	Erforderlich	Zulässige Werte	Standardwert
`PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL`	Der Detaillierungsgrad, der in Protokollen für die Erweiterung gemeldet wird. Wir empfehlen die Verwendung von `DEBUG` für die meisten Details zu Ihrer Cache-Konfiguration, während Sie die Erweiterung einrichten und testen. Protokolle für Lambda-Operationen werden automatisch an eine zugehörige CloudWatch Logs-Protokollgruppe übertragen.	Nein	`DEBUG | WARN | ERROR | NONE | INFO`	`INFO`

Beispielbefehle für die Verwendung der Erweiterung AWS Systems ManagerParameter Store und AWS Secrets Manager

Die Beispiele in diesem Abschnitt zeigen API-Aktionen zur Verwendung mit der AWS Secrets Manager Erweiterung AWS Systems Manager Parameter Store und.

Beispielbefehle für Parameter Store

Die Lambda-Erweiterung verwendet schreibgeschützten Zugriff auf die GetParameterAPI-Aktion.

Führen Sie zum Aufrufen dieser Aktion einen HTTP-GET-Aufruf ähnlich dem folgenden durch. Dieses Befehlsformat bietet Zugriff auf Parameter in der Standardparameterebene.

```
GET http://localhost:port/systemsmanager/parameters/get?name=parameter-
name&version=version&label=label&withDecryption={true|false}
```

In diesem Beispiel *parameter-name* steht sie für den vollständigen Parameternamen, z. B. für einen Parameter`MyParameter`, der sich nicht in einer Hierarchie befindet, oder `%2FDev%2FProduction%2FEast%2FProject-ABC%2FMyParameter` für einen Parameter mit dem Namen`/Dev/Production/East/Project-ABC/MyParameter`, der Teil einer Hierarchie ist.

> **Note**
>
> Bei Verwendung von GET-Aufrufen müssen Parameterwerte für HTTP codiert werden, um Sonderzeichen zu erhalten. Anstatt beispielsweise einen hierarchischen Pfad wie `/a/b/c` zu formatieren, codieren Sie Zeichen, die als Teil der URL interpretiert werden könnten, wie z. B. `%2Fa%2Fb%2Fc`.

*version*und *label* sind die Selektoren für die Verwendung mit der `GetParameter` Aktion verfügbar.

```
GET http://localhost:port/systemsmanager/parameters/get/?name=MyParameter&version=5
```

Um einen Parameter in einer Hierarchie aufzurufen, führen Sie einen HTTP-GET-Aufruf ähnlich dem folgenden durch.

```
GET http://localhost:port/systemsmanager/parameters/get?name=%2Fa%2Fb%2F&label=release
```

Um einen öffentlichen (globalen) Parameter aufzurufen, führen Sie einen HTTP-GET-Aufruf ähnlich dem folgenden durch.

```
GET http://localhost:port/systemsmanager/parameters/get/?name=%2Faws%2Fservice%20list
%2F…
```

Um einen HTTP-GET-Aufruf an ein Secrets-Manager-Secret mithilfe von Parameter Store-Referenzen durchzuführen, führen Sie einen HTTP-GET-Aufruf ähnlich dem folgenden durch.

```
GET http://localhost:port/systemsmanager/parameters/get?name=%2Faws%2Freference
%2Fsecretsmanager%2F…
```

Um einen Aufruf unter Verwendung des Amazon-Ressourcennamens (ARN) für einen Parameter zu tätigen, führen Sie einen HTTP-GET-Aufruf ähnlich dem folgenden durch.

```
GET http://localhost:port/systemsmanager/parameters/get?name=arn:aws:ssm:us-
east-1:123456789012:parameter/MyParameter
```

Um einen Aufruf zu tätigen, der auf einen `SecureString`-Parameter mit Entschlüsselung zugreift, führen Sie einen HTTP-GET-Aufruf ähnlich dem folgenden durch.

```
GET http://localhost:port/systemsmanager/parameters/get?
name=MyParameter&withDecryption=true
```

Sie können angeben, dass Parameter nicht entschlüsselt werden, indem Sie `withDecryption` weglassen oder explizit auf `false` setzen. Sie können auch entweder eine Version oder ein Label angeben, aber nicht beides. Wenn Sie dies tun, wird nur der erste davon verwendet, der in der URL nach dem Fragezeichen (?) steht.

AWS Lambda-Erweiterung für Parameter und Geheimnisse ARNs

Die folgenden Tabellen enthalten Erweiterungen ARNs für unterstützte Architekturen und Regionen.

Themen

- Erweiterung ARNs für die x86_64 und -Architekturen x86
- Erweiterung ARNs für ARM64 und Architekturen Mac with Apple silicon

Erweiterung ARNs für die x86_64 und -Architekturen x86

Letzte Aktualisierung: 22. Oktober 2025

Region	ARN
US East (Ohio)	`arn:aws:lambda:us-east-2:590474943231:layer:AWS-Parame`

Region	ARN
	ters-and-Secrets-Lambda-Ext ension:25
USA Ost (Nord-Virginia)	arn:aws:lambda:us-east-1:17 7933569100:layer:AWS-Parame ters-and-Secrets-Lambda-Ext ension:21
USA West (Nordkalifornien)	arn:aws:lambda:us-west-1:99 7803712105:layer:AWS-Parame ters-and-Secrets-Lambda-Ext ension:21
USA West (Oregon)	arn:aws:lambda:us-west-2:34 5057560386:layer:AWS-Parame ters-and-Secrets-Lambda-Ext ension:21
Africa (Cape Town)	arn:aws:lambda:af-south-1:3 17013901791:layer:AWS-Param eters-and-Secrets-Lambda-Ex tension:21
Asien-Pazifik (Hongkong)	arn:aws:lambda:ap-east-1:76 8336418462:layer:AWS-Parame ters-and-Secrets-Lambda-Ext ension:23
Asien-Pazifik (Taipeh)	arn:aws:lambda:ap-east-2:89 0742577149:layer:AWS-Parame ters-and-Secrets-Lambda-Ext ension:9

Region	ARN
Region Asien-Pazifik (Hyderabad)	arn:aws:lambda:ap-south-2:070087711984:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18
Asien-Pazifik (Jakarta)	arn:aws:lambda:ap-southeast-3:490737872127:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21
Asien-Pazifik (Melbourne)	arn:aws:lambda:ap-southeast-4:090732460067:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11
Asien-Pazifik (Malaysia)	arn:aws:lambda:ap-southeast-5:381492012281:layer:AWS-Parameters-and-Secrets-Lambda-Extension:10
Asien-Pazifik (Neuseeland)	arn:aws:lambda:ap-southeast-6:995508174458:layer:AWS-Parameters-and-Secrets-Lambda-Extension:5
Asien-Pazifik (Mumbai)	arn:aws:lambda:ap-south-1:176022468876:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21
Asien-Pazifik (Osaka)	arn:aws:lambda:ap-northeast-3:576959938190:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21

Region	ARN
Asien-Pazifik (Seoul)	`arn:aws:lambda:ap-northeast-2:738900069198:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21`
Asien-Pazifik (Singapur)	`arn:aws:lambda:ap-southeast-1:044395824272:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21`
Asien-Pazifik (Sydney)	`arn:aws:lambda:ap-southeast-2:665172237481:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21`
Asien-Pazifik (Thailand)	`arn:aws:lambda:ap-southeast-7:941377119484:layer:AWS-Parameters-and-Secrets-Lambda-Extension:9`
Asien-Pazifik (Tokio)	`arn:aws:lambda:ap-northeast-1:133490724326:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21`
Canada (Central)	`arn:aws:lambda:ca-central-1:200266452380:layer:AWS-Parameters-and-Secrets-Lambda-Extension:23`
Kanada West (Calgary)	`arn:aws:lambda:ca-west-1:243964427225:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11`

Region	ARN
China (Peking)	`arn:aws-cn:lambda:cn-north-1:287114880934:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21`
China (Ningxia)	`arn:aws-cn:lambda:cn-northwest-1:287310001119:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21`
Europe (Frankfurt)	`arn:aws:lambda:eu-central-1:187925254637:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21`
Europa (Irland)	`arn:aws:lambda:eu-west-1:015030872274:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21`
Europa (London)	`arn:aws:lambda:eu-west-2:133256977650:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21`
Europa (Milan)	`arn:aws:lambda:eu-south-1:325218067255:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21`
Europa (Paris)	`arn:aws:lambda:eu-west-3:780235371811:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21`

Region	ARN
Region Europa (Spanien)	arn:aws:lambda:eu-south-2:524103009944:layer:AWS-Parameters-and-Secrets-Lambda-Extension:20
Europa (Stockholm)	arn:aws:lambda:eu-north-1:427196147048:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21
Israel (Tel Aviv)	arn:aws:lambda:il-central-1:148806536434:layer:AWS-Parameters-and-Secrets-Lambda-Extension:11
Region Europa (Zürich)	arn:aws:lambda:eu-central-2:772501565639:layer:AWS-Parameters-and-Secrets-Lambda-Extension:18
Region Mexiko (Zentral)	arn:aws:lambda:mx-central-1:241533131596:layer:AWS-Parameters-and-Secrets-Lambda-Extension:8
Middle East (Bahrain)	arn:aws:lambda:me-south-1:832021897121:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21
Naher Osten (VAE)	arn:aws:lambda:me-central-1:858974508948:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21

Region	ARN
Südamerika (São Paulo)	`arn:aws:lambda:sa-east-1:933737806257:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21`
AWS GovCloud (US-Ost)	`arn:aws-us-gov:lambda:us-gov-east-1:129776340158:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21`
AWS GovCloud (US-West)	`arn:aws-us-gov:lambda:us-gov-west-1:127562683043:layer:AWS-Parameters-and-Secrets-Lambda-Extension:21`

Erweiterung ARNs für ARM64 und Architekturen Mac with Apple silicon

Letzte Aktualisierung: 22. Oktober 2025

Region	ARN
US East (Ohio)	`arn:aws:lambda:us-east-2:590047494323 1:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:25`
USA Ost (Nord-Virginia)	`arn:aws:lambda:us-east-1:177933569100:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:21`
Region USA West (Nordkalifornien)	`arn:aws:lambda:us-west-1:997803712105:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18`

Region	ARN
USA West (Oregon)	arn:aws:lambda:us-west-2:345057560386:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:21
Region Afrika (Kapstadt)	arn:aws:lambda:af-south-1:317013901791:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18
Region Asien-Pazifik (Hongkong)	arn:aws:lambda:ap-east-1:768336418462:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:20
Asien-Pazifik (Taipeh)	arn:aws:lambda:ap-east-2:890742577149:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:5
Region Asien-Pazifik (Hyderabad)	arn:aws:lambda:ap-south-2:070087711984:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:10
Region Asien-Pazifik (Jakarta)	arn:aws:lambda:ap-southeast-3:490737872127:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18
Asien-Pazifik (Melbourne)	arn:aws:lambda:ap-southeast-4:090732460067:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:10

Region	ARN
Asien-Pazifik (Malaysia)	arn:aws:lambda:ap-southeast-5:381492012281:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:10
Asien-Pazifik (Neuseeland)	arn:aws:lambda:ap-southeast-6:995508174458:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:3
Asien-Pazifik (Mumbai)	arn:aws:lambda:ap-south-1:176022468876:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:21
Asien-Pazifik (Osaka)	arn:aws:lambda:ap-northeast-3:576959938190:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18
Region Asien-Pazifik (Seoul)	arn:aws:lambda:ap-northeast-2:738900069198:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18
Asien-Pazifik (Singapur)	arn:aws:lambda:ap-southeast-1:044395824272:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:21
Asien-Pazifik (Sydney)	arn:aws:lambda:ap-southeast-2:665172237481:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:21

Region	ARN
Asien-Pazifik (Thailand)	arn:aws:lambda:ap-southeast-7:941377119484:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:8
Asien-Pazifik (Tokio)	arn:aws:lambda:ap-northeast-1:133490724326:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:21
Region Kanada (Zentral)	arn:aws:lambda:ca-central-1:200266452380:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:20
Kanada West (Calgary)	arn:aws:lambda:ca-west-1:243964427225:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:10
China (Peking)	arn:aws-cn:lambda:cn-north-1:287114880934:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:10
China (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:287310001119:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:10
Europe (Frankfurt)	arn:aws:lambda:eu-central-1:187925254637:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:21

Region	ARN
Europa (Irland)	`arn:aws:lambda:eu-west-1:015030872274:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:21`
Europa (London)	`arn:aws:lambda:eu-west-2:133256977650:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:21`
Region Europa (Mailand)	`arn:aws:lambda:eu-south-1:325218067255:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18`
Region Europa (Paris)	`arn:aws:lambda:eu-west-3:780235371811:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18`
Region Europa (Spanien)	`arn:aws:lambda:eu-south-2:524103009944:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:12`
Region Europa (Stockholm)	`arn:aws:lambda:eu-north-1:427196147048:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:148806536434:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:10`

Region	ARN
Region Europa (Zürich)	arn:aws:lambda:eu-central-2:772501565639:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:10
Region Mexiko (Zentral)	arn:aws:lambda:mx-central-1:241533131596:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:7
Region Naher Osten (Bahrain)	arn:aws:lambda:me-south-1:832021897121:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18
Naher Osten (VAE)	arn:aws:lambda:me-central-1:858974508948:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:10
Region Südamerika (São Paulo)	arn:aws:lambda:sa-east-1:933737806257:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:18
AWS GovCloud (US-Ost)	arn:aws-us-gov:lambda:us-gov-east-1:129776340158:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:10
AWS GovCloud (US-West)	arn:aws-us-gov:lambda:us-gov-west-1:127562683043:layer:AWS-Parameters-and-Secrets-Lambda-Extension-Arm64:10

Integration in andere Produkte und Services

AWS Systems Manager bietet integrierte Integration für die Produkte und Services in der folgenden Tabelle:

Ansible	Ansible ist eine IT-Automatisierungsplattform, mit der Ihre Anwendungen und Systeme einfacher bereitgestellt werden können. Systems Manager stellt das Systems-Manager-Dokument zur Verfügung (SSM-Dokument) `AWS-ApplyAnsiblePlaybooks` , mit dem Sie State Manager-Zuordnungen erstellen können, die Ansible Playbooks ausführen. Weitere Informationen Erstellen von Zuordnungen, die Ansible-Playbooks ausführen
Chef	Chef ist ein IT-Automatisierungstool, mit der Ihre Anwendungen und Systeme einfacher bereitgestellt werden können. Systems Manager stellt das `AWS-ApplyChefRecipes` SSM-Dokument bereit, mit dem Sie Zuordnungen in State Manager erstellen können, einem Tool in AWS Systems Manager, die Chef-Rezepte ausführen. Weitere Informationen Erstellen von Zuordnungen, die Chef-Rezepte ausführen Systems Manager lässt sich auch in Chef InSpec-Profile integrieren, sodass Sie Compliance-Scans ausführen und konforme und nicht konforme Knoten anzeigen können.

	Weitere Informationen Verwenden von Chef InSpec-Profilen mit Systems Manager Compliance
GitHub	GitHub bietet Hosting für Software-Entwicklung-Versionskontrolle und Zusammenarbeit. Systems Manager stellt das SSM-Dokument `AWS-RunDocument` bereit, mit dem Sie andere SSM-Dokumente ausführen können, die in GitHub gespeichert sind, und das SSM-Dokument `AWS-RunRemoteScript`, mit dem Sie Skripte ausführen können, die in GitHub gespeichert sind. Weitere Informationen • Ausführen von -Dokumenten von Remote-Standorten • Ausführen von Skripts von GitHub
Jenkins	Jenkins ist ein Open-Source-Automatisierungsserver, mit dem Entwickler ihre Software zuverlässig erstellen, testen und bereitstellen können. Automation, ein Tool in Systems Manager, kann als Post-Build-Schritt verwendet werden, um Anwendungsversionen in Amazon Machine Images (AMIs) vorzuinstallieren. Weitere Informationen Aktualisierung von AMIs mithilfe von Automation und Jenkins

ServiceNow	ServiceNow ist ein Enterprise-Service-Management-System, mit dem Sie Ihre IT-Services und -Abläufe verwalten können. Automation, Change Manager, Incident Manager und OpsCenter, Tools in Systems Manager, lassen sich mithilfe des AWS Service Management Connectors in ServiceNow integrieren. Mit dieser Integration können Sie von ServiceNow aus Korrespondenz anzeigen, erstellen, aktualisieren, Korrespondenz hinzufügen und AWS -Support-Fälle lösen. Weitere Informationen Integration mit ServiceNow

Terraform	HashiCorp Terraform ist ein Open-Source-Softwaretool für Infrastructure as Code (IaC), das einen Befehlszeilenschnittstellen (CLI)-Workflow zur Verwaltung verschiedener Cloud-Services bereitstellt. Für Systems Manager können Sie Terraform Folgendes verwalten oder bereitstellen: Ressourcen • aws_ssm_activation • aws_ssm_association • aws_ssm_default_patch_baseline • aws_ssm_document • aws_ssm_maintenance_window • aws_ssm_maintenance_window_target • aws_ssm_maintenance_window_task • aws_ssm_parameter • aws_ssm_patch_baseline • aws_ssm_patch_group • aws_ssm_resource_data_sync • aws_ssm_service_setting Datenquellen • aws_ssm_document • aws_ssm_instances • ssm_maintenance_windows • aws_ssm_parameter • aws_ssm_parameters_by_path • aws_ssm_patch_baseline

Themen

- Ausführen von Skripts von GitHub
- Verwenden von Chef InSpec-Profilen mit Systems Manager Compliance
- Integration mit ServiceNow

Ausführen von Skripts von GitHub

In diesem Thema wird beschrieben, wie Sie das vordefinierte Systems-Manager-Dokument (SSM-Dokument) `AWS-RunRemoteScript` zum Herunterladen von Skripts von GitHub verwenden, einschließlich Ansible Playbooks-, Python-, Ruby- und PowerShell-Skripts. Mit diesem SSM-Dokument müssen Sie Skripts nicht länger manuell in Amazon Elastic Compute Cloud (Amazon EC2) portieren oder in SSM-Dokumente einschließen. Die AWS Systems Manager-Integration mit GitHub fördert Infrastruktur als Code. Dadurch wird die benötigte Zeit zum Verwalten von Knoten bei gleichzeitiger Standardisierung von Konfigurationen in Ihrer Flotte reduziert.

Sie können auch benutzerdefinierte Systems Manager-Dokumente erstellen, mit denen Sie Skripts oder andere Systems Manager-Dokumente von Remote-Speicherorten herunterladen und ausführen können. Weitere Informationen finden Sie unter Erstellen von zusammengesetzten Dokumenten.

Sie können auch ein Verzeichnis mit mehreren Skripts herunterladen. Wenn Sie das primäre Skript im Verzeichnis ausführen, führt Systems Manager auch alle referenzierten Skripts aus, die im Verzeichnis enthalten sind.

Beachten Sie die folgenden wichtigen Hinweise zum Ausführen von Skripts von GitHub.

- Systems Manager prüft nicht, ob Ihr Skript auf einem Knoten ausgeführt werden kann. Stellen Sie sicher, dass die erforderliche Software auf dem Knoten installiert ist, bevor Sie das Skript herunterladen und ausführen. Alternativ können Sie ein zusammengesetztes Dokument erstellen, das die Software über Run Command oder State Manager installiert, Tools in AWS Systems Manager, und das Skript anschließend herunterlädt und ausführt.
- Sie sind dafür verantwortlich, sicherzustellen, dass alle GitHub-Anforderungen erfüllt sind. Dies umfasst die Aktualisierung Ihres Zugriffstokens, wenn erforderlich. Stellen Sie sicher, dass Sie die Anzahl an authentifizierten oder nicht authentifizierten Anfragen nicht überschreiten. Weitere Informationen finden Sie in der Dokumentation zu GitHub.
- GitHub Enterprise-Repositorys werden nicht unterstützt.

Themen

- Ansible-Playbooks ausführen von GitHub

- Python-Skripte von GitHub ausführen

Ansible-Playbooks ausführen von GitHub

Dieser Abschnitt beschreibt Verfahren zum Ausführen von Ansible Playbooks von GitHub mithilfe der Konsole oder der AWS Command Line Interface (AWS CLI).

Bevor Sie beginnen

Wenn Sie ein in einem privaten GitHub Repository gespeichertes Skript ausführen möchten, erstellen Sie einen AWS Systems Manager `SecureString` Parameter für Ihr GitHub Sicherheitszugriffstoken. Sie können nicht auf ein Skript in einem privaten GitHub-Repository zugreifen, indem Sie Ihr Token manuell über SSH übergeben. Das Zugriffstoken muss als `SecureString`-Systems Manager-Parameter übertragen werden. Weitere Informationen zum Erstellen eines `SecureString`-Parameters finden Sie unter Parameter Store-Parameter im Systems Manager erstellen.

Führen Sie ein Ansible-Playbook von GitHub (Konsole) aus

Ansible-Playbooks von GitHub ausführen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Befehl ausführen aus.
4. Wählen Sie in der Liste Command document (Befehlsdokument) die Option **`AWS-RunRemoteScript`** aus.
5. Führen Sie unter Command parameters die folgenden Schritte aus:
   - Wählen Sie unter Quelltyp die GitHub aus.
   - Geben Sie im Feld Source Info die für den Zugriff auf die Quelle erforderlichen Informationen im folgenden Format an.

```
{
  "owner": "owner_name",
  "repository": "repository_name",
  "getOptions": "branch:branch_name",
  "path": "path_to_scripts_or_directory",
  "tokenInfo": "{{ssm-secure:SecureString_parameter_name}}"
```

```
}
```

Dieses Beispiel lädt eine Datei mit dem Namen `webserver.yml` herunter.

```
{
    "owner": "TestUser1",
    "repository": "GitHubPrivateTest",
    "getOptions": "branch:myBranch",
    "path": "scripts/webserver.yml",
    "tokenInfo": "{{ssm-secure:mySecureStringParameter}}"
}
```

> **Note**
>
> `"branch"` ist nur erforderlich, wenn Ihr SSM-Dokument in einer anderen Verzweigung als `master` gespeichert ist.
> Um die Version Ihrer Skripts zu verwenden, die sich in einem bestimmten Commit in Ihrem Repository befinden, verwenden Sie `commitID` mit `getOptions`statt `branch`. Zum Beispiel:
> `"getOptions": "commitID:bbc1ddb94...b76d3bEXAMPLE",`

- Geben Sie im Feld Command Line Parameter für die Skriptausführung ein. Ein Beispiel.

  `ansible-playbook -i “localhost,” --check -c local webserver.yml`

- (Optional) Geben Sie im Feld Working Directory (Arbeitsverzeichnis) den Namen eines Verzeichnisses auf dem Knoten ein, auf dem das Skript heruntergeladen und ausgeführt werden soll.
- (Optional) Geben Sie unter Execution Timeout die Dauer in Sekunden an, bis das System die Skriptbefehlsausführung fehlschlagen lässt.

6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

> **Tip**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Weitere Parameter:

   - Geben Sie im Feld Kommentar Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (Sekunden) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

8. Für Ratenregelung:

   - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, sind diejenigen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle

(hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, nicht die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

10. Aktivieren Sie das Kontrollkästchen SNS-Benachrichtigungen aktivieren im Abschnitt SNS-Benachrichtigungen, wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon-SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

11. Klicken Sie auf Ausführen.

Führen Sie ein Ansible Playbook aus, indem Sie GitHubAWS CLI

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um ein Skript von GitHub herunterzuladen und auszuführen.

```
aws ssm send-command \
    --document-name "AWS-RunRemoteScript" \
    --instance-ids "instance-IDs"\
    --parameters '{"sourceType":["GitHub"],"sourceInfo":["{\"owner
\":\"owner_name\", \"repository\": \"repository_name\", \"path
\": \"path_to_file_or_directory\", \"tokenInfo\":\"{{ssm-
secure:name_of_your_SecureString_parameter}}\" }"],"commandLine":
["commands_to_run"]}'
```

   Hier ist ein Beispielbefehl für die Ausführung auf einem lokalen Linux-Computer.

```
aws ssm send-command \
    --document-name "AWS-RunRemoteScript" \
    --instance-ids "i-02573cafcfEXAMPLE" \
    --parameters '{"sourceType":["GitHub"],"sourceInfo":["{\"owner\":\"TestUser1\",
 \"repository\": \"GitHubPrivateTest\", \"path\": \"scripts/webserver.yml\",
 \"tokenInfo\":\"{{ssm-secure:mySecureStringParameter}}\" }"],"commandLine":
["ansible-playbook -i “localhost,” --check -c local webserver.yml"]}'
```

Python-Skripte von GitHub ausführen

Dieser Abschnitt enthält Verfahren, mit denen Sie Python-Skripte entweder über die AWS Systems Manager Konsole oder die AWS Command Line Interface (AWS CLI) ausführen können. GitHub

Ausführen eines Python-Skripts von GitHub (Konsole)

Ausführen eines Python-Skripts von GitHub

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Run Command aus.
3. Wählen Sie Befehl ausführen aus.
4. Wählen Sie in der Liste Command document (Befehlsdokument) die Option **AWS-RunRemoteScript** aus.
5. Führen Sie unter Command parameters (Befehlsparameter) die folgenden Schritte aus:
   - Wählen Sie unter Quelltyp die GitHub aus.
   - Geben Sie im Feld Source Info die für den Zugriff auf die Quelle erforderlichen Informationen im folgenden Format an:

```
{
    "owner": "owner_name",
    "repository": "repository_name",
    "getOptions": "branch:branch_name",
    "path": "path_to_document",
    "tokenInfo": "{{ssm-secure:SecureString_parameter_name}}"
}
```

Im Folgenden wird beispielsweise ein Verzeichnis von Skripts mit dem Namen complex-script heruntergeladen.

```
{
    "owner": "TestUser1",
    "repository": "SSMTestDocsRepo",
    "getOptions": "branch:myBranch",
    "path": "scripts/python/complex-script",
    "tokenInfo": "{{ssm-secure:myAccessTokenParam}}"
}
```

> **Note**
>
> `"branch"` ist nur erforderlich, wenn Ihre Skripts in einer anderen Verzweigung als `master` gespeichert sind.
> Um die Version Ihrer Skripts zu verwenden, die sich in einem bestimmten Commit in Ihrem Repository befinden, verwenden Sie `commitID` mit `getOptions`statt `branch`. Zum Beispiel:
> `"getOptions": "commitID:bbc1ddb94...b76d3bEXAMPLE",`

- Geben Sie für Command Line (Befehlszeile) Parameter für die Skriptausführung ein. Ein Beispiel.

```
mainFile.py argument-1 argument-2
```

  In diesem Beispiel wird `mainFile.py` ausgeführt. Diese Datei kann anschließend andere Skripts im Verzeichnis `complex-script` ausführen.

- (Optional) Geben Sie für Working Directory (Arbeitsverzeichnis) den Namen eines Verzeichnisses auf dem Knoten ein, auf dem das Skript heruntergeladen und ausgeführt werden soll.
- (Optional) Geben Sie für Execution Timeout (Ausführungstimeout) die Dauer in Sekunden an, bis das System die Skriptbefehlausführung fehlschlagen lässt.

6. Identifizieren Sie für den Abschnitt Targets (Ziele) die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

> **Tip**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

7. Für Weitere Parameter:

   - Geben Sie im Feld Kommentar Informationen zu diesem Befehl ein.
   - Geben Sie für Timeout (Sekunden) in Sekunden an, wie lange gewartet werden soll, bis für die gesamte Befehlsausführung ein Fehler auftritt.

8. Für Ratenregelung:

   - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

     > **Note**
     >
     > Wenn Sie Ziele ausgewählt haben, indem Sie Tags für verwaltete Knoten oder AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

   - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

9. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

   > **Note**
   >
   > Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, sind diejenigen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle

(hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, nicht die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

10. Aktivieren Sie das Kontrollkästchen SNS-Benachrichtigungen aktivieren im Abschnitt SNS-Benachrichtigungen, wenn Sie über den Status der Befehlsausführung benachrichtigt werden möchten,

    Weitere Informationen zum Konfigurieren von Amazon-SNS-Benachrichtigungen für Run Command finden Sie unter Überwachung von Systems Manager-Statusänderungen mit Amazon SNS-Benachrichtigungen.

11. Klicken Sie auf Ausführen.

Führen Sie ein Python-Skript aus, GitHub indem Sie den AWS CLI

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

   Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.

2. Führen Sie den folgenden Befehl aus, um ein Skript von GitHub herunterzuladen und auszuführen.

```
aws ssm send-command --document-name "AWS-RunRemoteScript" --instance-
ids "instance-IDs" --parameters '{"sourceType":["GitHub"],"sourceInfo":
["{\"owner\":\"owner_name\", \"repository\":\"repository_name\", \"path\":
 \"path_to_script_or_directory\"}"],"commandLine":["commands_to_run"]}'
```

   Ein Beispiel.

```
aws ssm send-command --document-name "AWS-RunRemoteScript" --instance-ids
 "i-02573cafcfEXAMPLE" --parameters '{"sourceType":["GitHub"],"sourceInfo":
["{\"owner\":\"TestUser1\", \"repository\":\"GitHubTestPublic\", \"path\":
```

```
\"scripts/python/complex-script\"}"],"commandLine":["mainFile.py argument-1
argument-2 "]}'
```

In diesem Beispiel wird ein Verzeichnis von Skripts mit dem Namen `complex-script` heruntergeladen. Der `commandLine`-Eintrag führt `mainFile.py` aus. Diese Datei kann anschließend andere Skripts im Verzeichnis `complex-script` ausführen.

Verwenden von Chef InSpec-Profilen mit Systems Manager Compliance

AWS Systems Manager integriert mit Chef InSpec. Chef InSpecist ein Open-Source-Testframework, mit dem Sie menschenlesbare Profile erstellen können, um sie in GitHub oder im Amazon Simple Storage Service (Amazon S3) zu speichern. Anschließend können Sie Systems Manager verwenden, um Compliance-Scans auszuführen und konforme und nicht konforme Knoten anzuzeigen. Ein Profil ist eine Sicherheits-, Compliance- oder Richtlinienanforderung für Ihre Datenverarbeitungsumgebung. Sie können beispielsweise Profile erstellen, die folgende Überprüfungen durchführen, wenn Sie Ihre Knoten mit Compliance scannen, einem Tool in AWS Systems Manager:

- Überprüfen Sie, ob bestimmte Ports geöffnet oder geschlossen sind.
- Überprüfen Sie, ob bestimmte Anwendungen ausgeführt werden.
- Überprüfen Sie, ob bestimmte Pakete installiert sind.
- Prüfen Sie die Windows-Registry-Schlüssel auf spezifische Eigenschaften.

Sie können InSpec Profile nur für Amazon Elastic Compute Cloud (Amazon EC2) -Instances erstellen, die Sie mit Systems Manager verwalten. Lokale Server oder virtuelle Maschinen (VMs) werden nicht unterstützt. Das folgende Chef InSpec-Beispielprofil prüft, ob Port 22 geöffnet ist.

```
control 'Scan Port' do
impact 10.0
title 'Server: Configure the service port'
desc 'Always specify which port the SSH server should listen to.
Prevent unexpected settings.'
describe sshd_config do
its('Port') { should eq('22') }
end
end
```

InSpec enthält eine Sammlung von Ressourcen, mit denen Sie schnell Schecks und Prüfkontrollen erstellen können. InSpec verwendet die InSpecdomänenspezifische Sprache (DSL) zum Schreiben dieser Steuerelemente in Ruby. Sie können auch Profile verwenden, die von einer großen Benutzergemeinschaft erstellt wurden. InSpec Zum Beispiel GitHub umfasst das DevSec chef-os-hardeningProjekt auf Dutzende von Profilen, mit denen Sie Ihre Knoten schützen können. Sie können Profile in GitHub oder Amazon S3 erstellen und speichern.

Funktionsweise

So funktioniert die Verwendung von InSpec Profilen mit Compliance:

1. Identifizieren Sie entweder vordefinierte InSpec Profile, die Sie verwenden möchten, oder erstellen Sie Ihre eigenen. Sie können zunächst vordefinierte Profile auf GitHub verwenden. Informationen zum Erstellen eigener InSpec Profile finden Sie unter Chef InSpecChef-Profile.
2. Speichern Sie Profile entweder in einem öffentlichen oder privaten GitHub-Repository oder in einem S3-Bucket.
3. Führen Sie Compliance mit Ihren InSpec Profilen mithilfe des Systems Manager Manager-Dokuments (SSM-Dokument) `AWS-RunInspecChecks` durch. Sie können einen Konformitätsscan startenRun Command, indem Sie ein Tool in AWS Systems Manager für On-Demand-Scans verwenden, oder Sie können regelmäßige Konformitätsscans mithilfe State Manager eines Tools in AWS Systems Manager planen.
4. Identifizieren Sie nicht konforme Knoten, indem Sie die Compliance-API oder Compliance-Konsole verwenden.

> **Note**
>
> Notieren Sie die folgenden Informationen:
>
> - Chef verwendet einen Client auf Ihren Knoten zur Verarbeitung des Profils. Sie müssen den Client nicht installieren. Wenn Systems Manager das SSM-Dokument `AWS-RunInspecChecks` ausführt, prüft das System, ob der Client installiert ist. Falls nicht, installiert Systems Manager den Chef-Client während des Scans und deinstalliert den Client wieder, nachdem der Scan abgeschlossen ist.
> - Ausführen des SSM-Dokuments `AWS-RunInspecChecks`, weist, wie in diesem Thema beschrieben, einen Compliance-Eintrag vom Typ `Custom:Inspec` zu jedem

Ziel-Knoten zu. Um diesen Konformitätstyp zuzuweisen, ruft das Dokument den PutComplianceItemsAPI-Vorgang auf.

Einen InSpec Konformitätsscan ausführen

Dieser Abschnitt enthält Informationen zum Ausführen eines InSpec Konformitätsscans mithilfe der Systems Manager Manager-Konsole und der AWS Command Line Interface (AWS CLI). In der Konsolenprozedur wird angezeigt, wie Sie State Manager konfigurieren, um den Scan auszuführen. Das AWS CLI Verfahren zeigt, wie die Konfiguration für Run Command die Ausführung des Scans konfiguriert wird.

Ausführen eines InSpec Konformitätsscans mit State Manager (Konsole)

Um einen InSpec Konformitätsscan State Manager mit der AWS Systems Manager Konsole auszuführen

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich State Manager aus.
3. Wählen Sie Create association (Zuordnung erstellen) aus.
4. Geben Sie im Abschnitt Provide association details (Zuordungsdetails bereitstellen) einen Namen ein.
5. Wählen Sie in der Liste Dokument die Option **`AWS-RunInspecChecks`**. aus.
6. Wählen Sie in der Liste Document version (Dokumentversion) die Option Latest at runtime (Neueste zur Laufzeit) aus.
7. Klicken Sie im Abschnitt Parameter, in der Liste Quelltyp entweder auf GitHub oder S3.

   Wenn Sie möchten GitHub, geben Sie den Pfad zu einem InSpec Profil in einem öffentlichen oder privaten GitHub Repository in das Feld Quellinformationen ein. Hier ist ein Beispielpfad zu einem öffentlichen Profil, das vom Systems Manager Manager-Team vom folgenden Ort aus bereitgestellt wurde: https://github.com/awslabs/amazon- ssm/tree/master/Compliance/InSpec/PortCheck.

   ```
   {"owner":"awslabs","repository":"amazon-ssm","path":"Compliance/InSpec/
   PortCheck","getOptions":"branch:master"}
   ```

Wenn Sie S3 wählen, geben Sie im Feld Quellinformationen eine gültige URL zu einem InSpec Profil in einem S3-Bucket ein.

Weitere Informationen zur Integration von Systems Manager mit GitHub und Amazon S3 finden Sie unter Ausführen von Skripts von GitHub.

8. Identifizieren Sie für den Abschnitt Ziele die verwalteten Knoten, auf denen Sie diese Operation ausführen möchten, indem Sie Tags angeben, Instances oder Edge-Geräte manuell auswählen oder eine Ressourcengruppe angeben.

> **Tip**
>
> Wenn ein verwalteter Knoten, den Sie erwarten, nicht aufgeführt ist, finden Sie weitere Informationen unter Problembehandlung bei der Verfügbarkeit verwalteter Knoten Tipps zur Fehlerbehebung.

9. Verwenden Sie im Abschnitt Specify schedule (Zeitplan festlegen) die Zeitplan-Builder-Optionen, um einen Zeitplan für das Ausführen des Compliance-Scans zu erstellen.
10. Für Ratenregelung:

    - Geben Sie unter Nebenläufigkeit entweder eine Zahl oder einen Prozentsatz der verwalteten Knoten an, auf denen der Befehl gleichzeitig ausgeführt werden soll.

      > **Note**
      >
      > Wenn Sie Ziele ausgewählt haben, indem Sie Tags angegeben haben, die auf verwaltete Knoten angewendet wurden, oder indem Sie AWS Ressourcengruppen angegeben haben und Sie sich nicht sicher sind, wie viele verwaltete Knoten das Ziel sind, schränken Sie die Anzahl der Ziele ein, die das Dokument gleichzeitig ausführen können, indem Sie einen Prozentsatz angeben.

    - Geben Sie unter Fehlerschwellenwert an, wann die Ausführung des Befehls auf anderen verwalteten Knoten beendet werden soll, nachdem dafür entweder auf einer bestimmten Anzahl oder einem Prozentsatz von Knoten ein Fehler aufgetreten ist. Falls Sie beispielsweise drei Fehler angeben, sendet Systems Manager keinen Befehl mehr, wenn der vierte Fehler empfangen wird. Von verwalteten Knoten, auf denen der Befehl noch verarbeitet wird, werden unter Umständen ebenfalls Fehler gesendet.

11. (Optional) Wenn Sie im Abschnitt Ausgabeoptionen die Befehlsausgabe in einer Datei speichern möchten, aktivieren Sie das Kontrollkästchen Schreiben in einen S3-Bucket aktivieren. Geben Sie die Namen für den Bucket und das Präfix (Ordner) in die Textfelder ein.

    > **Note**
    >
    > Die S3-Berechtigungen, die das Schreiben der Daten in einen S3-Bucket ermöglichen, sind diejenigen des Instanzprofils (für EC2 Instances) oder der IAM-Servicerolle (hybridaktivierte Maschinen), die der Instanz zugewiesen wurden, nicht die des IAM-Benutzers, der diese Aufgabe ausführt. Weitere Informationen finden Sie unter Instance-Berechtigungen für Systems Manager konfigurieren oder Eine IAM-Servicerolle für eine Hybrid-Umgebung erstellen. Wenn sich der angegebene S3-Bucket in einem anderen AWS-Konto befindet, stellen Sie außerdem sicher, dass das Instance-Profil oder die IAM-Servicerolle, die dem verwalteten Knoten zugeordnet ist, über die erforderlichen Berechtigungen zum Schreiben in diesen Bucket verfügt.

12. Wählen Sie Zuordnung erstellen. Das System erstellt die Zuordnung und führt den Compliance-Scan automatisch aus.
13. Warten Sie einige Minuten, bis der Scan abgeschlossen ist, und wählen Sie dann Compliance im Navigationsbereich aus.
14. Suchen Sie unter Corresponding managed instances (Entsprechende verwaltete Instances) die Knoten, in denen die Spalte Compliance Type (Compliance-Typ) Custom:Inspec lautet.
15. Wählen Sie eine Knoten-ID aus, um die Details von nicht konformen Status anzuzeigen.

Einen InSpec Konformitätsscan mit () ausführen Run CommandAWS CLI

1. Installieren und konfigurieren Sie AWS Command Line Interface (AWS CLI), falls Sie dies noch nicht getan haben.

    Weitere Informationen finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI.
2. Führen Sie einen der folgenden Befehle aus, um ein InSpec Profil GitHub entweder von Amazon S3 aus auszuführen.

    Der -Befehl verwendet die folgenden Parameter:

    - sourceType: GitHub oder Amazon S3

- sourceInfo: URL zum InSpec Profilordner entweder in GitHub oder einem S3-Bucket. Der Ordner muss die InSpec Basisdatei (*.yml) und alle zugehörigen Steuerelemente (*.rb) enthalten.

GitHub

```
aws ssm send-command --document-name "AWS-RunInspecChecks" --targets
 '[{"Key":"tag:tag_name","Values":["tag_value"]}]' --parameters '{"sourceType":
["GitHub"],"sourceInfo":["{\"owner\":\"owner_name\", \"repository\":
\"repository_name\", \"path\": \"Inspec.yml_file"}"]}'
```

Ein Beispiel.

```
aws ssm send-command --document-name "AWS-RunInspecChecks" --targets
 '[{"Key":"tag:testEnvironment","Values":["webServers"]}]' --parameters
 '{"sourceType":["GitHub"],"getOptions":"branch:master","sourceInfo":["{\"owner\":
\"awslabs\", \"repository\":\"amazon-ssm\", \"path\": \"Compliance/InSpec/PortCheck
\"}"]}'
```

Amazon S3

```
aws ssm send-command --document-name "AWS-RunInspecChecks" --targets
 '[{"Key":"tag:tag_name","Values":["tag_value"]}]' --parameters'{"sourceType":
["S3"],"sourceInfo":["{\"path\":\"https://s3.aws-api-domain/amzn-s3-demo-
bucket/Inspec.yml_file\"}"]}'
```

Ein Beispiel.

```
aws ssm send-command --document-name "AWS-RunInspecChecks" --targets
 '[{"Key":"tag:testEnvironment","Values":["webServers"]}]' --
parameters'{"sourceType":["S3"],"sourceInfo":["{\"path\":\"https://s3.aws-api-
domain/amzn-s3-demo-bucket/InSpec/PortCheck.yml\"}"]}'
```

3. Führen Sie den folgenden Befehl aus, um eine Übersicht des Compliance-Scans anzuzeigen.

```
aws ssm list-resource-compliance-summaries --filters
 Key=ComplianceType,Values=Custom:Inspec
```

4. Führen Sie den folgenden Befehl aus, um Details eines Knotens anzuzeigen, der nicht konform ist.

```
aws ssm list-compliance-items --resource-ids node_ID --resource-type
 ManagedInstance --filters Key=DocumentName,Values=AWS-RunInspecChecks
```

Integration mit ServiceNow

ServiceNow bietet ein Cloud-basiertes Servicemanagement-System zum Erstellen und Verwalten von Workflows auf Organisationsebene, z. B. für IT-Services, Ticketing-Systeme und Support. Der AWS Service Management Connector integriert ServiceNow mit Systems Manager für Bereitstellung, Verwaltung und Betrieb von AWS-Ressourcen von ServiceNow. Sie können den AWS Service Management Connector zur Integration von ServiceNow in Automation, Change Manager, Incident Manager und OpsCenter verwenden, alles Tools in AWS Systems Manager.

Sie können mit ServiceNow folgende Aufgaben ausführen:

- Führen Sie Automatisierungs-Playbooks aus Systems Manager aus.
- Zeigen Sie Incidents über Systems Manager OpsItems an, aktualisieren diese und beheben Sie sie.
- Zeigen Sie Betriebselemente, z. B. Vorfälle, über Systems Manager OpsCenter an und verwalten Sie diese.
- Zeigen Sie aus einer kuratierten Liste vorab genehmigter Änderungsvorlagen Systems-Manager-Änderungsanforderungen an und führen Sie diese aus.
- Verwalten und beheben Sie Vorfälle, an denen von AWS gehostete Anwendungen beteiligt sind, durch die Integration mit Incident Manager.

> **Note**
>
> Weitere Informationen über die Integration mit ServiceNow finden Sie unter Konfiguration von AWS-Serviceintegrationen im Administratorhandbuch zu AWS Service Management Connector.

AWS Systems Manager Referenz

Die folgenden Informationen und Themen können Ihnen dabei helfen, AWS Systems Manager - Lösungen besser zu implementieren.

Auftraggeber

In AWS Identity and Access Management (IAM) können Sie einem Dienst mithilfe des Principal-Richtlinienelements Zugriff auf Ressourcen gewähren oder verweigern. Der Wert des Prinzipal-Richtlinienelements für Systems Manager lautet `ssm.amazonaws.com`.

Unterstützte Geräte AWS-Regionen und Endgeräte

Siehe Service-Endpunkte von Systems Manager im Allgemeine Amazon Web Services-Referenz.

Service Quotas

Weitere Informationen finden Sie unter Systems Manager Service Quotas im Allgemeine Amazon Web Services-Referenz.

API Reference

Siehe AWS Systems Manager -API-Referenz.

AWS CLI Befehlsreferenz

Siehe AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz.

AWS -Tools für PowerShell Cmdlet-Referenz

Weitere Informationen finden Sie AWS Systems Manager im Abschnitt der AWS -Tools für PowerShell Cmdlet-Referenz.

SSM Agent-Repository in GitHub

Siehe aws/. amazon-ssm-agent

Stellen Sie ein Frage

Probleme mit Systems Manager in AWS re:Post

AWS Nachrichten-Blog

Verwaltungs-Tools

Weitere Referenzthemen

- Verwenden dieses Dienstes mit einem AWS SDK
- Referenz: Amazon-S3-Buckets für Patch-Operationen
- Referenz: EventBridge Amazon-Ereignistypen und -muster für Systems Manager
- Referenz: Cron- und Rate-Ausdrücke für System Manager
- Referenz: ec2messages, ssmmessages und andere API-Operationen
- Referenz: Datums- und Uhrzeitzeichenfolgen für Systems Manager

Verwenden dieses Dienstes mit einem AWS SDK

AWS Software Development Kits (SDKs) sind für viele beliebte Programmiersprachen verfügbar. Jedes SDK bietet eine API, Codebeispiele und Dokumentation, die es Entwicklern erleichtern, Anwendungen in ihrer bevorzugten Sprache zu erstellen.

SDK-Dokumentation	Codebeispiele
AWS SDK für C++	AWS SDK für C++ Codebeispiele
AWS CLI	AWS CLI Code-Beispiele
AWS SDK für Go	AWS SDK für Go Code-Beispiele
AWS SDK für Java	AWS SDK für Java Code-Beispiele
AWS SDK für JavaScript	AWS SDK für JavaScript Code-Beispiele
AWS SDK für Kotlin	AWS SDK für Kotlin Code-Beispiele
AWS SDK für .NET	AWS SDK für .NET Code-Beispiele
AWS SDK für PHP	AWS SDK für PHP Code-Beispiele
AWS -Tools für PowerShell	AWS -Tools für PowerShell Code-Beispiele
AWS SDK für Python (Boto3)	AWS SDK für Python (Boto3) Code-Beispiele
AWS SDK für Ruby	AWS SDK für Ruby Code-Beispiele
AWS SDK für Rust	AWS SDK für Rust Code-Beispiele

SDK-Dokumentation	Codebeispiele
AWS SDK für SAP ABAP	AWS SDK für SAP ABAP Code-Beispiele
AWS SDK für Swift	AWS SDK für Swift Code-Beispiele

> **Beispiel für die Verfügbarkeit**
>
> Sie können nicht finden, was Sie brauchen? Fordern Sie ein Codebeispiel an, indem Sie unten den Link Provide feedback (Feedback geben) auswählen.

Referenz: Amazon-S3-Buckets für Patch-Operationen

Im Zuge der Durchführung verschiedener Patch Manager Patch-Operationen greift AWS Systems Manager-Agent (SSM Agent) auf bestimmte Amazon Simple Storage Service (Amazon S3)-Buckets zu, die Amazon Web Services gehören und von Amazon Web Services (AWS) verwaltet werden. Diese S3-Buckets sind öffentlich zugänglich und SSM Agent verbindet sich standardmäßig über HTTP-Aufrufe mit ihnen.

Wenn Sie jedoch einen Virtual Private Cloud (VPC)-Endpunkt für Ihre Systems-Manager-Operationen verwenden, müssen Sie eine explizite Berechtigung in einem Amazon Elastic Compute Cloud (Amazon EC2)-Instance-Profil für Systems Manager oder in einer Servicerolle für Nicht-EC2-Maschinen in einer Hybrid- und Multi-Cloud-Umgebung gewähren. Andernfalls haben Ihre Ressourcen keinen Zugriff auf diese öffentlichen Buckets.

In diesem Referenzthema sind die Patch-Buckets für jeden unterstützten AWS-Region aufgeführt.

Informationen zur Angabe dieser S3-Buckets in EC2-Instance-Profilen finden Sie unter SSM Agent-Kommunikationen mit AWS -verwalteten S3-Buckets.

Informationen zur Verwendung von VPC-Endpunkten mit Systems Manager finden Sie unter Verbessern Sie die Sicherheit von EC2 Instanzen mithilfe von VPC-Endpunkten für Systems Manager.

Themen

- Buckets, die SSM-Befehlsdokumente für Patch-Operationen enthalten (Linux und Windows Server)
- Buckets, die SSM-Befehlsdokumente für Patch-Operationen enthalten (macOS)

- Buckets mit verwalteten Patch-Baseline-Snapshots AWS

Buckets, die SSM-Befehlsdokumente für Patch-Operationen enthalten (Linux und Windows Server)

Buckets mit diesem Format `aws-patch-manager-`*`region`*`-`*`unique-suffix` enthalten die folgenden Dokumente, die bei Patch Manager Patch-Vorgängen unter Linux- und Windows Server-Betriebssystemen verwendet werden:

- `AWS-RunPatchBaseline`
- `AWS-RunPatchBaselineAssociation`
- `AWS-RunPatchBaselineWithHooks`
- `AWS-InstanceRebootWithHooks`
- `AWS-PatchAsgInstance`
- `AWS-PatchInstanceWithRollback`

Name der Region	Regionscode	**`aws-patch-manager-`*`region`*`-`*`suffix`**-Bucket
USA Ost (Ohio)	us-east-2	aws-patch-manager-us-east-2-552881074
USA Ost (Nord-Virginia)	us-east-1	aws-patch-manager-us-east-1-1970c647d
USA West (Nordkalifornien)	us-west-1	aws-patch-manager-us-west-1-8badb4304
USA West (Oregon)	us-west-2	aws-patch-manager-us-west-2-34d7f99f8
Afrika (Kapstadt)	af-south-1	aws-patch-manager-af-south-1-bdd5f65a9
Asien-Pazifik (Hongkong)	ap-east-1	aws-patch-manager-ap-east-1-632356271

Name der Region	Regionscode	**aws-patch-manager-*region*-*suffix***-Bucket
Asien-Pazifik (Hyderabad)	ap-south-2	aws-patch-manager-ap-south-2-32f4b4128
Asien-Pazifik (Jakarta)	ap-southeast-3	aws-patch-manager-ap-southeast-3-aa48fc462
Asien-Pazifik (Melbourne)	ap-southeast-4	aws-patch-manager-ap-southeast-4-01e2c40d3
Asien-Pazifik (Mumbai)	ap-south-1	aws-patch-manager-ap-south-1-cb7c62ff9
Asia Pacific (Osaka)	ap-northeast-3	aws-patch-manager-ap-northeast-3-67373598a
Asien-Pazifik (Seoul)	ap-northeast-2	aws-patch-manager-ap-northeast-2-10467995c
Asien-Pazifik (Singapur)	ap-southeast-1	aws-patch-manager-ap-southeast-1-7fdfd9ef7
Asien-Pazifik (Sydney)	ap-southeast-2	aws-patch-manager-ap-southeast-2-17283a275
Asien-Pazifik (Tokio)	ap-northeast-1	aws-patch-manager-ap-northeast-1-4849fa78f
Kanada (Zentral)	ca-central-1	aws-patch-manager-ca-central-1-3148e69e3
Kanada West (Calgary)	ca-west-1	aws-patch-manager-ca-west-1-9e3a4b2f9
Europa (Frankfurt)	eu-central-1	aws-patch-manager-eu-central-1-9163fdaaf

Name der Region	Regionscode	**aws-patch-manager-*region*-*suffix***-Bucket
Europa (Irland)	eu-west-1	aws-patch-manager-eu-west-1-5522fb710
Europa (London)	eu-west-2	aws-patch-manager-eu-west-2-902a2bc74
Europa (Mailand)	eu-south-1	aws-patch-manager-eu-south-1-c52f3f594
Europa (Paris)	eu-west-3	aws-patch-manager-eu-west-3-29bf85721
Europa (Spanien)	eu-south-2	aws-patch-manager-eu-south-2-a4cf248b1
Europa (Stockholm)	eu-north-1	aws-patch-manager-eu-north-1-795879e9b
Europa (Zürich)	eu-central-2	aws-patch-manager-eu-central-2-184ce43c8
Israel (Tel Aviv)	il-central-1	aws-patch-manager-il-central-1-e221cb57b
Naher Osten (Bahrain)	me-south-1	aws-patch-manager-me-south-1-a53fc9dce
Naher Osten (VAE)	me-central-1	aws-patch-manager-me-central-1-2932f2f80
Südamerika (São Paulo)	sa-east-1	aws-patch-manager-sa-east-1-ddf4b6a09

Buckets, die SSM-Befehlsdokumente für Patch-Operationen enthalten (macOS)

Buckets mit diesem Format `aws-patchmanager-macos-`*`region`*`-`*`unique-suffix` enthalten die folgenden Dokumente, die bei Patch Manager Patch-Vorgängen unter macOS-Betriebssystemen verwendet werden:

- `AWS-RunPatchBaseline`
- `AWS-RunPatchBaselineAssociation`
- `AWS-RunPatchBaselineWithHooks`
- `AWS-InstanceRebootWithHooks`
- `AWS-PatchAsgInstance`
- `AWS-PatchInstanceWithRollback`

Name der Region	Regionscode	**`aws-patchmanager-macos-`** ***`region-suffix`*** **-Bucket**
USA Ost (Ohio)	us-east-2	aws-patchmanager-macos-us-east-2-552881074
USA Ost (Nord-Virginia)	us-east-1	aws-patchmanager-macos-us-east-1-1970c647d
USA West (Nordkalifornien)	us-west-1	aws-patchmanager-macos-us-west-1-8badb4304
USA West (Oregon)	us-west-2	aws-patchmanager-macos-us-west-2-34d7f99f8
Afrika (Kapstadt)	af-south-1	aws-patchmanager-macos-af-south-1-bdd5f65a9
Asien-Pazifik (Hongkong)	ap-east-1	aws-patchmanager-macos-ap-east-1-632356271

Name der Region	Regionscode	**aws-patchmanager-macos-** *region-suffix*-Bucket
Asien-Pazifik (Hyderabad)	ap-south-2	aws-patchmanager-macos-ap-south-2-32f4b4128
Asien-Pazifik (Jakarta)	ap-southeast-3	aws-patchmanager-macos-ap-southeast-3-aa48fc462
Asien-Pazifik (Melbourne)	ap-southeast-4	aws-patchmanager-macos-ap-southeast-4-01e2c40d3
Asien-Pazifik (Mumbai)	ap-south-1	aws-patchmanager-macos-ap-south-1-cb7c62ff9
Asia Pacific (Osaka)	ap-northeast-3	aws-patchmanager-macos-ap-northeast-3-67373598a
Asien-Pazifik (Seoul)	ap-northeast-2	aws-patchmanager-macos-ap-northeast-2-10467995c
Asien-Pazifik (Singapur)	ap-southeast-1	aws-patchmanager-macos-ap-southeast-1-7fdfd9ef7
Asien-Pazifik (Sydney)	ap-southeast-2	aws-patchmanager-macos-ap-southeast-2-17283a275
Asien-Pazifik (Tokio)	ap-northeast-1	aws-patchmanager-macos-ap-northeast-1-4849fa78f
Kanada (Zentral)	ca-central-1	aws-patchmanager-macos-ca-central-1-3148e69e3
Kanada West (Calgary)	ca-west-1	aws-patchmanager-macos-ca-west-1-9e3a4b2f9
Europa (Frankfurt)	eu-central-1	aws-patchmanager-macos-eu-central-1-9163fdaaf

Name der Region	Regionscode	**aws-patchmanager-macos-** *region-suffix*-Bucket
Europa (Irland)	eu-west-1	aws-patchmanager-macos-eu-west-1-5522fb710
Europa (London)	eu-west-2	aws-patchmanager-macos-eu-west-2-902a2bc74
Europa (Mailand)	eu-south-1	aws-patchmanager-macos-eu-south-1-c52f3f594
Europa (Paris)	eu-west-3	aws-patchmanager-macos-eu-west-3-29bf85721
Europa (Spanien)	eu-south-2	aws-patchmanager-macos-eu-south-2-a4cf248b1
Europa (Stockholm)	eu-north-1	aws-patchmanager-macos-eu-north-1-795879e9b
Europa (Zürich)	eu-central-2	aws-patchmanager-macos-eu-central-2-184ce43c8
Israel (Tel Aviv)	il-central-1	aws-patchmanager-macos-il-central-1-e221cb57b
Naher Osten (Bahrain)	me-south-1	aws-patchmanager-macos-me-south-1-a53fc9dce
Naher Osten (VAE)	me-central-1	aws-patchmanager-macos-me-central-1-2932f2f80
Südamerika (São Paulo)	sa-east-1	aws-patchmanager-macos-sa-east-1-ddf4b6a09

Buckets mit verwalteten Patch-Baseline-Snapshots AWS

Buckets mit dem Format `patch-baseline-snapshot-`*`region`* oder `patch-baseline-snapshot-`*`region`*`-`*`unique-suffix`* enthalten von AWS verwaltete Patch-Baseline-Snapshots. Der Zugriff auf diesen S3-Bucket ist erforderlich, wenn Sie eines der folgenden SSM-Dokumente verwenden:

- `AWS-RunPatchBaseline`
- `AWS-RunPatchBaselineAssociation`
- `AWS-RunPatchBaselineWithHooks`
- `AWS-ApplyPatchBaseline` (ein altes SSM-Dokument)

Name der Region	Regionscode	**`patch-baseline-snapshot-*`** -Bucket
USA Ost (Ohio)	us-east-2	patch-baseline-snapshot-us-east-2
USA Ost (Nord-Virginia)	us-east-1	patch-baseline-snapshot-us-east-1
USA West (Nordkalifornien)	us-west-1	patch-baseline-snapshot-us-west-1
USA West (Oregon)	us-west-2	patch-baseline-snapshot-us-west-2
Afrika (Kapstadt)	af-south-1	patch-baseline-snapshot-af-south-1-tbxdb5b9
Asien-Pazifik (Hongkong)	ap-east-1	patch-baseline-snapshot-ap-east-1
Asien-Pazifik (Hyderabad)	ap-south-2	patch-baseline-snapshot-ap-south-2-50209442

Name der Region	Regionscode	**patch-baseline-snapshot-*** -Bucket
Asien-Pazifik (Jakarta)	ap-southeast-3	patch-baseline-snapshot-ap-southeast-3-be0a3174
Asien-Pazifik (Melbourne)	ap-southeast-4	patch-baseline-snapshot-ap-southeast-4-dc6f76ce
Asien-Pazifik (Mumbai)	ap-south-1	patch-baseline-snapshot-ap-south-1
Asia Pacific (Osaka)	ap-northeast-3	patch-baseline-snapshot-ap-northeast-3
Asien-Pazifik (Seoul)	ap-northeast-2	patch-baseline-snapshot-ap-northeast-2
Asien-Pazifik (Singapur)	ap-southeast-1	patch-baseline-snapshot-ap-southeast-1
Asien-Pazifik (Sydney)	ap-southeast-2	patch-baseline-snapshot-ap-southeast-2
Asien-Pazifik (Tokio)	ap-northeast-1	patch-baseline-snapshot-ap-northeast-1
Kanada (Zentral)	ca-central-1	patch-baseline-snapshot-ca-central-1
Kanada West (Calgary)	ca-west-1	patch-baseline-snapshot-ca-west-1
Europa (Frankfurt)	eu-central-1	patch-baseline-snapshot-eu-central-1
Europa (Irland)	eu-west-1	patch-baseline-snapshot-eu-west-1

Name der Region	Regionscode	**patch-baseline-snapshot-*** -Bucket
Europa (London)	eu-west-2	patch-baseline-snapshot-eu-west-2
Europa (Mailand)	eu-south-1	patch-baseline-snapshot-eu-south-1
Europa (Paris)	eu-west-3	patch-baseline-snapshot-eu-west-3
Europa (Spanien)	eu-south-2	patch-baseline-snapshot-eu-south-2-df2c9d70
Europa (Stockholm)	eu-north-1	patch-baseline-snapshot-eu-north-1
Europa (Zürich)	eu-central-2	patch-baseline-snapshot-eu-central-2
Israel (Tel Aviv)	il-central-1	patch-baseline-snapshot-il-central-1
Naher Osten (Bahrain)	me-south-1	patch-baseline-snapshot-me-south-1-uduvl7q8
Naher Osten (VAE)	me-central-1	patch-baseline-snapshot-me-central-1
Südamerika (São Paulo)	sa-east-1	patch-baseline-snapshot-sa-east-1

Referenz: EventBridge Amazon-Ereignistypen und -muster für Systems Manager

> **Note**
>
> Amazon EventBridge ist die bevorzugte Methode, um Ihre Veranstaltungen zu verwalten . CloudWatch Bei Events und EventBridge handelt es sich um denselben zugrunde liegenden Service und dieselbe API, EventBridge bieten aber mehr Funktionen. Änderungen, die Sie in einer CloudWatch oder mehreren Konsolen vornehmen, EventBrid ge spiegeln sich in jeder Konsole wider. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch.

Mit Amazon können Sie Regeln erstellen EventBridge, die eingehenden Ereignissen entsprechen, und diese zur Verarbeitung an Ziele weiterleiten.

Ein Ereignis weist auf eine Änderung in einer Umgebung in Ihren eigenen Anwendungen, Software as a Service (SaaS)- Anwendungen oder einem AWS-Service hin. Ereignisse werden auf bestmögliche Weise ausgegeben. Nachdem ein in einer Regel spezifizierter Ereignistyp erkannt wurde, EventBridge wird er zur Verarbeitung an ein bestimmtes Ziel weitergeleitet. Zu den Zielen können Amazon Elastic Compute Cloud (Amazon EC2) -Instances, AWS Lambda Funktionen, Amazon Kinesis Kinesis-Streams, Amazon Elastic Container Service (Amazon ECS) -Aufgaben, AWS Step Functions Zustandsmaschinen, Amazon Simple Notification Service (Amazon SNS) -Themen, Amazon Simple Queue Service (Amazon SQS) -Warteschlangen, integrierte Ziele und vieles mehr gehören.

Informationen zum Erstellen von EventBridge Regeln finden Sie in den folgenden Themen:

- Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge
- EventBridge Amazon-Veranstaltungsbeispiele für Systems Manager
- Erste Schritte mit Amazon EventBridge im EventBridge Amazon-Benutzerhandbuch

Im Rest dieses Themas werden die Typen von Systems Manager Manager-Ereignissen beschrieben, die Sie in Ihre EventBridge Regeln aufnehmen können.

Ereignistyp: Automation

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
EC2 Benachrichtigung über Änderung des Status der Automatisierungsausführung	Der Gesamtstatus eines Automation-Workflows ändert sich. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • Approved • Canceled • Fehlgeschlagen • PendingApproval • PendingChangeCalendarOverride • Rejected (Abgelehnt) • Geplant • Herzlichen Glückwunsch • TimedOut
EC2 Benachrichtigung über Statusänderung des Automatisierungsschritts	Der Status eines bestimmten Schrittes in einem Automation-Workflows ändert sich. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • Canceled • Fehlgeschlagen • Herzlichen Glückwunsch • TimedOut

Ereignistyp: Change Calendar

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Änderung des Kalenderstatus	Der Status des Change Calendar ändert sich. Sie können einer Ereignisregel eine oder beide der folgenden Statusänderungen hinzufügen: • OPEN • CLOSED Statusänderungen für Kalender, die von anderen AWS-Konten freigegeben werden, werden nicht unterstützt.

Ereignistyp: Change Manager

> **Change ManagerÄnderung der Verfügbarkeit**
>
> AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestehende Kunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Aktualisierung des Status der Änderungsanforderung	Der Status der Change Manager-Änderungsanforderung. Sie können die folgenden Status in einer Ereignisregel verwenden: • Approved • Rejected (Abgelehnt)

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
	• InProgress

Ereignistyp: Configuration Compliance

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Configuration Compliance-Statusänderung	Der Zustand eines verwalteten Knotens ändert sich je nach Zuordnungs-Compliance oder Patch-Compliance. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • compliant • non_compliant

Ereignistyp: Inventory

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Inventory Resource-Statusänderung	Das Löschen von benutzerdefiniertem Inventar und ein PutInventoryAufruf, der eine alte Schemaversion verwendet. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • Beim benutzerdefinierten Inventartyp wurde ein Ereignis auf einem bestimmten Knoten gelöscht. EventBridge sendet ein Ereignis pro Knoten und Benutzerdefiniert Inventory Type.

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
	• Löschereignis für benutzerdefinierten Inventartyp auf allen Knoten. • PutInventory ruft mit dem Ereignis der alten Schemaversion auf. EventBridgesendet dieses Ereignis, wenn die Schemaversion kleiner als die aktuelle Schemaversion ist. Dieses Ereignis gilt für alle Inventararten. Weitere Informationen finden Sie unter Verwenden von EventBridge zum Überwachen von Inventory-Ereignissen.

Ereignistyp: Wartungsfenster

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Benachrichtigung über die Statusänderung des Wartungsfensters	Der Gesamtstatus eines oder mehrerer Wartungsfenster ändert sich. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • DISABLED (DEAKTIVIERT) • ENABLED (AKTIVIERT)
Benachrichtigung zur Zielregistrierung des Wartungsfensters	Der Status eines oder mehrerer Wartungsf enster ändert sich. Sie können einer Ereignisr egel eine oder mehrere der folgenden Statusänderungen hinzufügen: • DEREGISTERED • REGISTERED

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
	• UPDATED
Benachrichtigung über Ausführungsstatusänderung des Wartungsfensters	Der Gesamtstatus eines Wartungsfensters ändert sich während der Ausführung. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • CANCELLED • CANCELLING • FEHLGESCHLAGEN • IN_PROGRESS • PENDING • SKIPPED_OVERLAPPING • ERFOLG • TIMED_OUT
Benachrichtigung über Aufgaben-Ausführungsstatusänderung des Wartungsfensters	Der Status einer Aufgabe in einem Wartungsfenster ändert sich während der Ausführung. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • CANCELLED • CANCELLING • FEHLGESCHLAGEN • IN_PROGRESS • ERFOLG • TIMED_OUT

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Benachrichtigung über Zielaufrufungsstatusänderung der Wartungsfenstersaufgabe	Der Status einer Wartungsfensteraufgabe für ein bestimmtes Ziel ändert sich. Diese Benachrichtigung wird nur für Run Command-Aufgaben vollständig unterstützt. Sie können bei dieser Art von Aufgabe einer Ereignisregel eine oder mehrere der folgenden Zustandsänderungen hinzufügen: • CANCELLED • CANCELLING • FEHLGESCHLAGEN • IN_PROGRESS • ERFOLG • TIMED_OUT Für Automatisierung AWS Lambda, und AWS Step Functions Aufgaben werden nur die Status `IN_PROGRESS` und EventBridge gemeldet. `COMPLETE` `COMPLETE`wird gemeldet, ob die Aufgabe erfolgreich war oder nicht.
Benachrichtigung über Aufgabenregistrierung des Wartungsfenster	Der Status einer oder mehrerer Wartungsfensteraufgaben ändert sich. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • DEREGISTERED • REGISTERED • UPDATED

Ereignistyp: OpsCenter

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
OpsItem erstellen	Tritt auf, wenn OpsItem erstellt wird. Sie können Regeln für einen der folgenden OpsItem-Typen hinzufügen: • /aws/issue • /aws/task • /aws/insight • /aws/actionitem
OpsItem updaten	Tritt auf, wenn OpsItem aktualisiert wird. Sie können Regeln für einen der folgenden OpsItem-Typen hinzufügen: • /aws/issue • /aws/task • /aws/insight • /aws/actionitem

Ereignistyp: Parameter Store

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
Änderung des Parameterspeichers	Der Status eines Paramaters ändert sich. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • Create • Aktualisierung • Delete

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
	• LabelParameterVersion Weitere Informationen finden Sie unter Konfigurieren von EventBridge-Regeln für Parameter und Parameter-Richtlinien.
Parameterstore-Richtlinienaktion	Eine Bedingung für eine erweiterte Parameter richtlinienänderung ist erfüllt. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • Ablauf • ExpirationNotification • NoChangeNotification Weitere Informationen finden Sie unter Konfigurieren von EventBridge-Regeln für Parameter und Parameter-Richtlinien.

Ereignistyp: Run Command

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
EC2 Benachrichtigung über Statusänderung bei Befehlsaufruf	Der Status eines an eine einzelne verwaltete Instance gesendeten Befehls ändert sich. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • Herzlichen Glückwunsch • InProgress • TimedOut

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
	• Canceled • Fehlgeschlagen
EC2 Benachrichtigung über Änderung des Befehlsstatus	Der Gesamtstatus eines Befehls ändert sich. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • Herzlichen Glückwunsch • InProgress • TimedOut • Canceled • Fehlgeschlagen

Ereignistyp: State Manager

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
EC2 State ManagerÄnderung des Verbandsstaats	Die gesamte Status einer Zuordnung ändert sich, wenn sie angewendet wird. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen: • Fehlgeschlagen • Ausstehend • Herzlichen Glückwunsch
EC2 State ManagerÄnderung des Status der Instanzzugehörigkeit	Der Zustand einer einzeln verwalteten Instance, auf die eine Zuordnung abzielt, ändert sich. Sie können einer Ereignisregel eine oder mehrere der folgenden Statusänderungen hinzufügen:

Ereignistyp-Name	Beschreibung der Ereignisse, die Sie einer Regel hinzufügen können
	• Fehlgeschlagen • Ausstehend • Herzlichen Glückwunsch

Referenz: Cron- und Rate-Ausdrücke für System Manager

Wenn Sie eine State Manager Zuordnung oder ein Wartungsfenster in erstellen AWS Systems Manager, geben Sie einen Zeitplan an, nach dem das Fenster oder die Zuordnung ausgeführt werden soll. Sie können einen Zeitplan als zeitbasierten Eintrag, einen sogenannten Cron-Ausdruck, oder als häufigkeitsbasierten Eintrag, einen sogenannten Rate-Ausdruck angeben.

Allgemeine Informationen zu Cron- und Rate-Ausdrücken

Die folgenden Informationen gelten für Cron- und Rate-Ausdrücke sowohl für Wartungsfenster als auch für Zuordnungen.

Zeitpläne für Einzelläufe

Wenn Sie ein eine Zuordnung oder ein Wartungsfenster erstellen, können Sie einen Zeitstempel in koordinierter Weltzeit (Coordinated Universal Time, UTC) angeben, damit es einmalig zum angegebenen Zeitpunkt ausgeführt wird. Beispiel: `"at(2020-07-07T15:55:00)"`

Offsets planen

Assoziationen und Wartungsfenster unterstützen nur für Cron-Ausdrücke zudem auch Zeitplanversätze. Ein Zeitplanversatz ist die Anzahl der Tage, die nach dem über einen CRON-Ausdruck angegebenen Datum und der angegebenen Uhrzeit gewartet werden soll, bevor die Assoziation oder das Wartungsfenster ausgeführt wird.

Maintenance window example

Im folgenden Beispiel wird mit dem CRON-Ausdruck die Ausführung eines Wartungsfensters um 23.30 Uhr am dritten Dienstag jedes Monats geplant. Wenn der Zeitplanversatz jedoch 2 lautet, wird das Wartungsfenster erst zwei Tage später um 23:30 Uhr ausgeführt.

```
aws ssm create-maintenance-window \
    --name "My-Cron-Offset-Maintenance-Window" \
```

```
    --allow-unassociated-targets \
    --schedule "cron(30 23 ? * TUE#3 *)" \
    --duration 4 \
    --cutoff 1 \
    --schedule-offset 2
```

Association example

Im folgenden Befehl plant der Cron-Ausdruck, dass eine Zuordnung am zweiten Donnerstag eines jeden Monats ausgeführt wird. Da der Zeitplanversatz jedoch 3 ist, wird die Zuordnung erst am nächsten Sonntag, also drei Tage später, ausgeführt.

```
aws ssm create-association \
    --name "AWS-UpdateSSMAgent" \
    --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f" \
    --schedule-expression "cron(0 0 ? * THU#2 *)"  \
    --schedule-offset 3
    --apply-only-at-cron-interval
```

> **Note**
>
> Um einen Offset mit einer Zuordnung zu verwenden, müssen Sie die `--apply-only-at-cron-interval`-Option angeben. Diese Option sagt dem System eine Assoziation nicht unmittelbar nach der Erstellung auszuführen.

Wenn Sie eine Zuordnung oder ein Wartungsfenster mit einem Cron-Ausdruck erstellen, das sich auf einen im aktuellen Zeitraum bereits vergangenen Tag bezieht, jedoch ein Zeitplanversatzdatum hinzufügen, das in der Zukunft liegt, wird die Assoziation oder das Wartungsfenster in dem betreffenden Zeitraum nicht ausgeführt. Es wird im folgenden Zeitraum in Kraft treten. Wenn Sie beispielsweise einen Cron-Ausdruck angeben, der gestern ein Wartungsfenster ausgeführt hätte, und einen Zeitplanversatz von zwei Tagen hinzufügen, wird das Wartungsfenster morgen nicht ausgeführt.

Pflichtfelder

Cron-Ausdrücke für Wartungsfenster haben sechs erforderliche Felder. Cron-Ausdrücke für Zuordnungen haben fünf. (State Manager unterstützt derzeit nicht die Angabe von Monaten in Cron-Ausdrücken für Zuordnungen.) Ein zusätzliches Feld, das Feld `Seconds` (das erste in einem Cron-Ausdruck) ist optional. Die Felder werden durch Leerzeichen voneinander getrennt.

Beispiele für Cron-Ausdrücke

Minuten	Stunden	Tag des Monats	Monat	Wochentag	Jahr	Bedeutung
0	10	*	*	?	*	Ausführung jeden Tag um 10:00 Uhr (UTC)
15	12	*	*	?	*	Ausführung jeden Tag um 12:15 Uhr (UTC)
0	18	?	*	MO-FR	*	Wird jeden Montag bis Freitag um 18:00 Uhr (UTC) ausgeführt (nur Wartungsfenster)
0	8	1	*	?	*	Ausführung jeden 1. Tag des Monats um 08:00 Uhr (UTC)

> **Note**
>
> Day-of-week Bereiche (wie`MON-FRI`) und durch Kommas getrennte Tageslisten (z. B.`MON,TUE,WED`) werden nur für Wartungsfenster unterstützt. Für Assoziationen müssen Sie einen einzelnen Tag (z. B.`MON`) angeben oder `*` für jeden Tag verwenden.

Unterstützte Werte

Die folgende Tabelle zeigt die Werte, die für erforderliche Cron-Einträge unterstützt werden.

Unterstützte Werte für Cron-Ausdrücke

Feld	Werte	Platzhalter
Minuten	`0-59`	`, - * /`
Stunden	`0-23`	`, - * /`
D ay-of-month	`1-31`	`, - * ? / L W`
Monat (nur Wartungsfenster)	`1–12` oder `JAN–DEC`	`, - * /`
D ay-of-week	`1–7` oder `SUN–SAT`	`, - * ? / L #`
Jahr	`1970-2199`	`, - * /`

> **Note**
>
> Sie können keinen Wert in den Feldern day-of-month und in den day-of-week Feldern desselben Cron-Ausdrucks angeben. Wenn Sie einen Wert in einem der Felder angeben, verwenden Sie ? (Fragezeichen) im anderen Feld.

Platzhalter für Cron-Ausdrücke

Die folgende Tabelle zeigt die Platzhalterwerte, die von Cron-Ausdrücken unterstützt werden.

> **Note**
>
> Cron-Ausdrücke, die zu schnelleren Häufigkeiten als fünf (5) führen, werden nicht unterstützt. Die Support für die Angabe von day-of-week sowohl einem day-of-month Wert als auch eines Werts ist nicht vollständig. Verwenden Sie das Fragezeichen (?) in einem dieser Felder.

Unterstützte Platzhalter für Cron-Ausdrücke

Platzhalter	Description
,	Das Platzhalterzeichen , (Komma) umfasst zusätzliche Werte. Im Feld "Monat" steht JAN, FEB, MAR für Januar, Februar und März.
-	Das Platzhalterzeichen - (Bindestrich) gibt einen Bereich an. Im Feld "Tag" steht 1-15 für die Tage 1 bis 15 des angegebenen Monats.
*	Das Platzhalterzeichen * (Sternchen) steht für alle Werte im Feld. Im Feld für die Stundenangaben steht * für alle Stunden.
/	Das Platzhalterzeichen / (Schrägstrich) steht für schrittweise Steigerungen. Im Feld „Minuten“ könnten Sie 1/10 eingeben, um jede 10. Minute beginnend mit der ersten Minute der Stunde anzugeben. 1/10 gibt daher die erste, 11., 21., 31. usw. Minute an.
?	Das Platzhalterzeichen ? (Fragezeichen) steht für einen bestimmten Wert. In das Day-of-month Feld könnten Sie 7 eingeben und wenn es Ihnen egal wäre, welcher Wochentag der 7. ist, könnten Sie eingeben? auf dem Day-of-week Feld.

Platzhalter	Description
L	Der `L` Platzhalter in den Day-of-week Feldern Day-of-month oder gibt den letzten Tag des Monats oder der Woche an.
W	Der `W` Platzhalter in dem Day-of-month Feld gibt einen Wochentag an. In dem Day-of-mo nth Feld gibt 3W den Tag an, der dem dritten Wochentag des Monats am nächsten liegt.
#	Der # Platzhalter in dem day-of-week Feld, gefolgt von einer Zahl zwischen eins und fünf, gibt einen bestimmten Tag des Monats an. 5 #3 gibt den 3. Donnerstag des Monats an.

Rate-Ausdrücke

Rate-Ausdrücke bestehen aus den folgenden zwei Pflichtfeldern. Felder werden durch Leerzeichen voneinander getrennt.

Pflichtfelder für Rate-Ausdrücke

Feld	Werte
Wert	positive Zahl, z. B. 1 oder 15

Einheit	`minute`
	`minutes`
	`hour`
	`hours`
	`day`
	`days`

Wenn der Wert gleich 1 ist, muss die Einheit im Singular stehen. Wenn die Werte größer als 1 sind, muss die Einheit im Plural stehen. Beispielsweise sind `rate(1 hours)` und `rate(5 hour)` ungültige, `rate(1 hour)` und `rate(5 hours)` jedoch gültige Werte.

Themen

- Cron- und Rate-Ausdrücke für Zuordnungen
- Cron- und Rate-Ausdrücke für Wartungsfenster

Cron- und Rate-Ausdrücke für Zuordnungen

Dieser Abschnitt enthält Beispiele für Cron- und Rate-Ausdrücke für State Manager-Zuordnungen. Bevor Sie einen dieser Ausdrücke erstellen, beachten Sie die folgenden Informationen:

- Zuordnungen unterstützen die folgenden Cron-Ausdrücke: Alle 1/2, 1, 2, 4, 8 oder 12 Stunden; jeden Tag, jede Woche oder jeden angegebenen Tag und jede bestimmte Uhrzeit der Woche; ein bestimmter Tag in einer bestimmten Woche des Monats oder der letzte x-Tag des Monats zu einer bestimmten Zeit.
- Zuordnungen unterstützen die folgenden Rate-Ausdrücke: Intervalle von mindestens 30 Minuten und weniger als 31 Tagen.
- Wenn Sie das optionale Feld `Seconds` angeben, kann dessen Wert 0 (null) sein. Zum Beispiel: `cron(0 */30 * * * ? *)`
- Für einen Verband, der Metadaten für Inventory, ein Tool in, sammelt AWS Systems Manager, empfehlen wir die Verwendung eines Preisausdrucks.
- State Manager unterstützt derzeit nicht die Angabe von Monaten in Cron-Ausdrücken für Assoziationen.

Assoziationen unterstützen Cron-Ausdrücke, die einen Wochentag und das Zahlenzeichen (#) enthalten, um den x-ten Tag eines Monats für die Ausführung einer Assoziation anzugeben. Hier ist ein Beispiel, das am dritten Dienstag jeden Monats um 23:30 Uhr UTC einen Cron-Zeitplan ausführt:

```
cron(30 23 ? * TUE#3 *)
```

Hier ist ein Beispiel, das am zweiten Donnerstag jeden Monats um Mitternacht UTC läuft:

```
cron(0 0 ? * THU#2 *)
```

Assoziationen unterstützen auch das (L)-Zeichen, um dens letzten XTag des Monats anzugeben. Hier ist ein Beispiel, das am letzten Dienstag jeden Monats um Mitternacht UTC einen Cron-Zeitplan ausführt:

```
cron(0 0 ? * 3L *)
```

Um weiter zu steuern, wann eine Assoziation ausgeführt wird, z. B. wenn Sie zwei Tage nach dem Patch-Dienstag eine Assoziation ausführen möchten, können Sie einen Offset angeben. Ein Offset definiert, wie viele Tage nach dem geplanten Tag gewartet werden müssen, um eine Assoziation auszuführen. Wenn Sie beispielsweise einen Cron-Zeitplan mit `cron(0 0 ? * THU#2 *)` angegeben haben, können Sie die Nummer 3 im Schedule offset (Planversatz)-Feld angeben, um die Assoziation jeden Sonntag nach dem zweiten Donnerstag im Monat auszuführen.

Um Offsets zu verwenden, müssen Sie entweder Apply association only at the next specified Cron interval (Übernehmen der Assoziation erst für das nächste angegebene Cron-Intervall)-Option in der Konsole auswählen oder Sie müssen den Nutze`---apply-only-at-cron-interval`-Parameter über die Befehlszeile angeben. Diese Option sagt State Manager eine Assoziation nicht unmittelbar nach der Erstellung auszuführen.

Die folgende Tabelle zeigt die Cron-Beispiele für Zuordnungen.

Cron-Beispiele für Zuordnungen

Beispiel	Details
cron(0/30 * * * ? *)	Alle 30 Minuten
cron(0 0/1 * * ? *)	Stündlich
cron(0 0/2 * * ? *)	Alle 2 Stunden
cron(0 0/4 * * ? *)	Alle 4 Stunden
cron(0 0/8 * * ? *)	Alle 8 Stunden
cron(0 0/12 * * ? *)	Alle 12 Stunden
cron(15 13 ? * * *)	Täglich um 13:15 Uhr
cron(15 13 ? * MON *)	Jeden Montag um 13:15 Uhr
cron(30 23 ? * TUE#3 *)	Jeden dritten Dienstag im Monat um 23:30 Uhr

Hier sind einige Rate-Beispiele für Zuordnungen.

Rate-Beispiele für Zuordnungen

Beispiel	Details
rate(30 minutes)	Alle 30 Minuten
rate(1 hour)	Stündlich
rate(5 hours)	Alle 5 Stunden
rate(15 days)	Alle 15 Tage

AWS CLI Beispiele für Assoziationen

Um mithilfe von State Manager Assoziationen zu erstellen AWS CLI, fügen Sie den `--schedule-expression` Parameter mit einem Cron- oder Rate-Ausdruck hinzu. In den folgenden Beispielen wird der AWS CLI auf einem lokalen Linux-Computer verwendet.

> **Note**
>
> Wenn Sie eine neue Zuordnung erstellen, führt das System diese standardmäßig sofort nach der Erstellung und dann nach dem angegebenen Zeitplan aus. Geben Sie `--apply-only-at-cron-interval` an, damit die Zuordnung nicht unmittelbar nach der Erstellung ausgeführt wird. Dieser Parameter wird nicht für Rate-Ausdrücke unterstützt.

```
aws ssm create-association \
    --association-name "My-Cron-Association" \
    --schedule-expression "cron(0 2 ? * SUN *)" \
    --targets Key=tag:ServerRole,Values=WebServer \
    --name AWS-UpdateSSMAgent
```

```
aws ssm create-association \
    --association-name "My-Rate-Association" \
    --schedule-expression "rate(7 days)" \
    --targets Key=tag:ServerRole,Values=WebServer \
    --name AWS-UpdateSSMAgent
```

```
aws ssm create-association \
    --association-name "My-Rate-Association" \
    --schedule-expression "at(2020-07-07T15:55:00)" \
    --targets Key=tag:ServerRole,Values=WebServer \
    --name AWS-UpdateSSMAgent \
    --apply-only-at-cron-interval
```

Cron- und Rate-Ausdrücke für Wartungsfenster

Dieser Abschnitt enthält Beispiele für Cron- und Rate-Ausdrücke für Wartungsfenster.

Anders als State Manager-Zuordnungen unterstützen Wartungsfenster alle Cron- und Rate-Ausdrücke. Dies umfasst die Unterstützung für Werte im Sekundenfeld.

Beispielsweise führt der folgende Cron-Ausdruck mit 6 Feldern jeden Tag um 9:30 Uhr ein Wartungsfenster aus.

```
cron(30 09 ? * * *)
```

Durch Hinzufügen eines Werts zum Feld `Seconds` führt der folgende Cron-Ausdruck mit 7 Feldern jeden Tag um 9:30:24 Uhr ein Wartungsfenster aus.

```
cron(24 30 09 ? * * *)
```

Die folgende Tabelle enthält zusätzliche Beispiele für Cron-Ausdrücke mit 6 Feldern für Wartungsfenster.

Cron-Beispiele für Wartungsfenster

Beispiel	Details
cron(0 2 ? * THU#3 *)	02:00 Uhr jeden dritten Donnerstag im Monat
cron(15 10 ? * * *)	10:15 Uhr jeden Tag
cron(15 10 ? * MON-FRI *)	10:15 Uhr jeden Montag, Dienstag, Mittwoch, Donnerstag und Freitag
cron(0 2 L * ? *)	02:00 Uhr jeden letzten Tag im Monat
cron(15 10 ? * 6L *)	10:15 Uhr jeden letzten Freitag im Monat

Die folgende Tabelle enthält Beispiele für Raten von Wartungsfenstern.

Rate-Beispiele für Wartungsfenster

Beispiel	Details
rate(30 minutes)	Alle 30 Minuten
rate(1 hour)	Stündlich
rate(5 hours)	Alle 5 Stunden
rate(25 days)	Alle 25 Tage

AWS CLI Beispiele für Wartungsfenster

Um Wartungsfenster mit dem zu erstellen AWS CLI, fügen Sie dem `--schedule` Parameter einen Cron- oder Rate-Ausdruck oder einen Zeitstempel hinzu. In den folgenden Beispielen wird der AWS CLI auf einem lokalen Linux-Computer verwendet.

```
aws ssm create-maintenance-window \
    --name "My-Cron-Maintenance-Window" \
    --allow-unassociated-targets \
    --schedule "cron(0 16 ? * TUE *)" \
    --schedule-timezone "America/Los_Angeles" \
    --start-date 2021-01-01T00:00:00-08:00 \
    --end-date 2021-06-30T00:00:00-08:00 \
    --duration 4 \
    --cutoff 1
```

```
aws ssm create-maintenance-window \
    --name "My-Rate-Maintenance-Window" \
    --allow-unassociated-targets \
    --schedule "rate(7 days)" \
    --duration 4 \
    --schedule-timezone "America/Los_Angeles" \
    --cutoff 1
```

```
aws ssm create-maintenance-window \
    --name "My-TimeStamp-Maintenance-Window" \
    --allow-unassociated-targets \
```

```
    --schedule "at(2021-07-07T13:15:30)" \
    --duration 4 \
    --schedule-timezone "America/Los_Angeles" \
    --cutoff 1
```

Weitere Informationen

CRON-Ausdruck bei der Wikipedia-Webseite

Referenz: ec2messages, ssmmessages und andere API-Operationen

Beim Überwachen von API-Vorgängen werden Ihnen möglicherweise Aufrufe der folgenden Vorgänge angezeigt:

- `ec2messages:AcknowledgeMessage`
- `ec2messages:DeleteMessage`
- `ec2messages:FailMessage`
- `ec2messages:GetEndpoint`
- `ec2messages:GetMessages`
- `ec2messages:SendReply`
- `ssmmessages:CreateControlChannel`
- `ssmmessages:CreateDataChannel`
- `ssmmessages:OpenControlChannel`
- `ssmmessages:OpenDataChannel`
- `ssm:DescribeDocumentParameters`
- `ssm:DescribeInstanceProperties`
- `ssm:GetCalendar`
- `ssm:GetManifest`
- `ssm:ListInstanceAssociations`
- `ssm:PutCalendar`
- `ssm:PutConfigurePackageResult`
- `ssm:RegisterManagedInstance`

- `ssm:RequestManagedInstanceRoleToken`
- `ssm:UpdateInstanceAssociationStatus`
- `ssm:UpdateInstanceInformation`
- `ssm:UpdateManagedInstancePublicKey`

Dies sind spezielle Operationen, die von verwendet werden AWS Systems Manager, wie im Rest dieses Themas beschrieben.

API-Vorgänge (**ssmmessages**- und **ec2messages**-Endpunkte) im Zusammenhang mit Agenten

ssmmessages-API-Operationen

Systems Manager verwendet den `ssmmessages`-Endpunkt für die folgenden Arten von API-Vorgängen:

- Operationen vom Systems Manager Agent (SSM Agent) zum Systems-Manager-Service in der Cloud.
- Operationen von SSM Agent bisSession Manager, ein Tool in AWS Systems Manager, in der Cloud. Dieser Endpunkt ist zum Erstellen und Löschen von Sitzungskanälen mit dem Session Manager-Service in der Cloud erforderlich. Zusätzlich empfängt SSM Agent, wenn Konnektivität erlaubt ist, `Command`-Dokumente über dieses Amazon Message Gateway Service. Wenn Konnektivität nicht zulässig ist, empfängt SSM Agent `Command`-Dokumente über den Amazon Message Delivery Service. Weitere Informationen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Message Gateway Service.

> **Note**
>
> Wenn die `ssmmessages:OpenControlChannel` Berechtigung aus den Richtlinien entfernt wird, die mit Ihrem IAM-Instanzprofil oder Ihrer IAM-Dienstrolle verknüpft sind, SSM Agent verliert der verwaltete Knoten die Konnektivität zum Systems Manager Manager-Dienst in der Cloud. Es kann jedoch bis zu 1 Stunde dauern, bis eine Verbindung beendet wird, nachdem die Berechtigung entfernt wurde. Dies ist dasselbe Verhalten wie beim Löschen der IAM-Instanzrolle oder der IAM-Servicerolle.
> Beachten Sie, dass die `ssmmessages:OpenControlChannel` Berechtigung in der verwalteten Richtlinie Amazon enthalten ist SSMManagedInstanceCore, die in den

Anweisungen zum Erstellen eines IAM-Instance-Profils für EC2 Instances und zum Erstellen einer IAM-Servicerolle für EC2 Nicht-Instances verwendet wird.

- Operationen von Run Command.

ec2messages-API-Operationen

`ec2messages:*`-API-Operationen werden an den Amazon Message Delivery Service-Endpunkt gemacht. Systems Manager verwendet diesen Endpunkt für API-Operationen vom Systems-Manager-Agent (SSM Agent) an den Systems-Manager-Service in der Cloud.

Important

`ec2messages:*`-API-Vorgänge werden nur für Operationen in AWS-Regionen unterstützt, die vor 2024 gestartet wurden. In Regionen, die 2024 und später eingeführt wurden, werden nur `ssmmessages:*` API-Vorgänge unterstützt.

Rangfolge der Endpunktverbindungen

Ab Version 3.3.40.0 von SSM Agent begann Systems Manager, den `ssmmessages:*`-Endpunkt (Amazon Message Gateway Service), wann immer verfügbar, anstelle des `ec2messages:*`-Endpunkts (Amazon Message Delivery Service) zu verwenden.

Wenn Sie `ssmmessages:*` in Ihren AWS Identity and Access Management (IAM-) Berechtigungsrichtlinien Zugriff auf gewähren, SSM Agent stellt eine Verbindung zum `ssmmessages:*` Endpunkt her, auch wenn Ihr IAM-Instance-Profil so konfiguriert ist, dass beide Endpunkte zugelassen sind. Dazu gehören Richtlinien für IAM-Instance-Profile und IAM-Servicerollen, die Sie selbst erstellt haben, sowie für IAM-Instance-Profile, die mit der Host-Management-Konfiguration und der Quick Setup-Standard-Host-Management-Konfiguration erstellt wurden.

Wenn Sie Berechtigungen für beide Endpunkte bereitgestellt haben und API-Operationen beispielsweise mithilfe von CloudWatch Metrics überwachen, werden Ihnen keine Aufrufe von angezeigt. `ec2messages:*`

Für AWS-Regionen Produkte, die vor 2024 veröffentlicht wurden: Sie können zu diesem Zeitpunkt problemlos `ec2messages:*` Berechtigungen aus Ihren Richtlinien entfernen.

Failover der Endpunktverbindung

Nur für vor 2024 AWS-Regionen gestartete Versionen: Wenn Ihr IAM-Instanzprofil zum `ssmmessages:*` Zeitpunkt des Starts des Agenten keine Berechtigungen bereitstellt, sondern lediglich `ec2messages:*` SSM Agent eine Verbindung zum `ec2messages:*` Endpunkt herstellt. Wenn Sie `ssmmessages:*` und `ec2messages:*` zum Zeitpunkt des SSM Agent-Starts haben, aber `ssmmessages:*` entfernen, nachdem der Agent gestartet wurde, wechselt SSM Agent bald die Verbindung zum `ec2messages:*`-Endpunkt. Für Regionen, die 2024 und später eingeführt wurden, wird nur der `ssmmessages:*`-Endpunkt unterstützt.

Weitere Informationen zu den `ssmmessages`- und `ec2messages:*`-Endpunkten finden Sie in den folgenden Themen im AWS -Service-Authorization-Benutzerhandbuch.

- Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Message Gateway Service (`ssmmessages`).
- Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Message Delivery Service (`ec2messages:*`)

`ssm:*`-Namespace-Instance-bezogene API-Vorgänge

`DescribeDocumentParameters`

Systems Manager führt diesen API-Vorgang aus, um bestimmte Knoten in der EC2 Amazon-Konsole zu rendern. Ergebnisse der `DescribeDocumentParameters`-Operation werden im Knoten Dokumente angezeigt.

`DescribeInstanceProperties`

Systems Manager führt diese API-Operationen aus, um bestimmte Knoten in der EC2 Amazon-Konsole zu rendern. Ergebnisse der `DescribeInstanceProperties`-Operation werden im Knoten Fleet Manager angezeigt.

`GetCalendar`

Systems Manager führt diesen API-Vorgang aus, um Change Calendar-Typdokumente in der Change Calendar-Konsole zu rendern.

`GetManifest`

SSM Agent führt diesen API-Vorgang aus, um die Systemanforderungen für die Installation oder Aktualisierung einer bestimmten Version eines AWS Systems Manager Distributor-Pakets zu

ermitteln. Dies ist ein älterer API-Vorgang, der nicht verfügbar ist, wenn er nach 2017 AWS-Regionen gestartet wurde.

`ListInstanceAssociations`

SSM Agent ruft diesen API-Vorgang auf, um festzustellen, ob eine neue State Manager-Zuordnung verfügbar ist. Dieser API-Vorgang ist für die Funktion von State Manager erforderlich.

`PutCalendar`

Systems Manager führt diesen API-Vorgang aus, um Change Calendar-Typdokumente in der Change Calendar-Konsole zu aktualisieren.

`PutConfigurePackageResult`

SSM Agent führt diesen API-Vorgang aus, um Installationsfehler- und Latenzmetriken für öffentliche Distributor-Pakete im Konto des Paketbesitzers zu veröffentlichen.

`RegisterManagedInstance`

SSM Agent führt diesen API-Vorgang für die folgenden Szenarien aus:

- So registrieren Sie einen On-Premises-Server oder eine virtuelle Maschine (VM) mithilfe eines Aktivierungscodes und einer ID als verwaltete Instance bei Systems Manager.
- Um AWS IoT Greengrass Version 2 Anmeldeinformationen zu registrieren.

Dieser Vorgang wird auch von EC2 Amazon-Instances aufgerufen, auf denen SSM Agent Version 3.1.x oder höher ausgeführt wird.

`RequestManagedInstanceRoleToken`

SSM Agent führt diesen API-Vorgang aus, um temporäre Anmeldeinformationen für den Zugriff auf den verwalteten Knoten abzurufen.

`UpdateInstanceAssociationStatus`

SSM Agent führt diesen API-Vorgang aus, um eine Zuordnung zu aktualisieren. Dieser API-Vorgang ist erforderlichState Manager, damit ein Tool in AWS Systems Manager, funktioniert.

`UpdateInstanceInformation`

SSM Agent ruft den Systems-Manager-Service in der Cloud alle fünf Minuten auf, um Heartbeat-Informationen bereitzustellen. Dieser Aufruf ist erforderlich, um einen Heartbeat mit dem Agent beizubehalten, damit der Service erkennt, dass der Agent erwartungsgemäß funktioniert.

`UpdateManagedInstancePublicKey`

SSM Agent führt diesen API-Vorgang aus, um den öffentlichen Schlüssel bereitzustellen, nachdem das key pair auf dem verwalteten Knoten rotiert wurde. Der öffentliche Schlüssel wird zur Authentifizierung der mit dem privaten Schlüssel signierten Anfragen verwendet, um temporäre Anmeldeinformationen von Systems Manager zu erhalten.

ssm:* Namespace andere API-Vorgänge

`ExecuteApi`

Von Systems Manager delegierte Administratoren, die OpsItems in OpsCenter verwalten benötigen Zugriff auf diese API-Aktion, damit sie zugehörige Ressourcendetails zu OpsItems über mehrere AWS-Konten anzeigen können. Insbesondere gewährt diese API einem delegierten Administrator die Erlaubnis, die folgenden OpsItem Details in der OpsItem Beschreibung, den AWS-Managementkonsole Tags, der CloudFormation Vorlage, den AWS Config Änderungen, den CloudWatch Protokollen, Alarmen und AWS CloudTrail Ereignissen einzusehen. Weitere Informationen zum kontoübergreifenden OpsItems-Arbeiten finden Sie unter (Optional) Manuelle Einrichtung von OpsCenter für die zentrale kontenübergreifende Verwaltung von OpsItems. Weitere Informationen zu zugehörigen Ressourcendetails fürOpsItems finden Sie unter Hinzufügen zugehöriger Ressourcen zu einem OpsItem.

Referenz: Datums- und Uhrzeitzeichenfolgen für Systems Manager

AWS Systems Manager-API-Vorgänge akzeptieren Filter, um die Anzahl der von einer Anforderung zurückgegebenen Ergebnisse zu begrenzen. Einige dieser API-Vorgänge akzeptieren Filter, die zur Darstellung eines bestimmten Datums und einer bestimmten Uhrzeit eine formatierte Zeichenfolge erfordern. Beispielsweise akzeptiert der API-Vorgang `DescribeSessions` die Schlüssel `InvokedAfter` und `InvokedBefore` als gültige Werte für ein `SessionFilter`-Objekt. Ein weiteres Beispiel ist der API-Vorgang `DescribeAutomationExecutions`. Dieser akzeptiert die Schlüssel `StartTimeBefore` und `StartTimeAfter` als gültige Werte für ein `AutomationExecutionFilter`-Objekt. Die Werte, die Sie für diese Schlüssel beim Filtern Ihrer Anforderungen angeben, müssen dem ISO 8601-Standard entsprechen. Weitere Informationen zu ISO 8601 finden Sie unter ISO 8601.

Diese formatierten Datums- und Uhrzeitzeichenfolgen sind nicht auf Filter beschränkt. Es gibt auch API-Vorgänge, die eine im ISO 8601-Format formatierte Zeichenfolge erfordern,

um ein bestimmtes Datum und eine bestimmte Uhrzeit darzustellen, wenn ein Wert für einen Anforderungsparameter angegeben wird. Ein Beispiel ist der Anforderungsparameter `AtTime` für den Vorgang `GetCalendarState`. Das Erstellen dieser Zeichenfolgen ist schwierig. Die Beispiele in diesem Thema helfen Ihnen, formatierte Datums- und Uhrzeitzeichenfolgen für die Verwendung mit Systems Manager-API-Vorgängen zu erstellen.

Formatieren von Datums- und Uhrzeitzeichenfolgen für Systems Manager

Im Folgenden finden Sie ein Beispiel für eine im ISO 8601-Format formatierte Datums- und Uhrzeitzeichenfolge.

```
2024-05-08T15:16:43Z
```

Dies entspricht dem 8. Mai 2024 um 15:16 Uhr koordinierter Weltzeit (UTC). Der Kalenderdatumsbereich der Zeichenfolge wird durch ein vierstelliges Jahr, einen zweistelligen Monat und einen zweistelligen Tag dargestellt, getrennt durch Bindestriche. Dies kann im folgenden Format dargestellt werden.

```
YYYY-MM-DD
```

Der Zeitbereich der Zeichenfolge beginnt mit dem Buchstaben „T“ als Trennzeichen. Er wird durch eine zweistellige Stunde, eine zweistellige Minute und eine zweistellige Sekunde dargestellt, getrennt durch Doppelpunkte. Dies kann im folgenden Format dargestellt werden.

```
hh:mm:ss
```

Der Zeitbereich der Zeichenfolge endet mit dem Buchstaben „Z“, der den UTC-Standard angibt.

Erstellen benutzerdefinierter Datums- und Uhrzeitzeichenfolgen für Systems Manager

Sie können benutzerdefinierte Datums- und Uhrzeitzeichenfolgen auf Ihrem lokalen Computer mit Ihrem bevorzugten Befehlszeilen-Tool erstellen. Die Syntax, die Sie zum Erstellen einer im ISO 8601-Format formatierten Datums- und Uhrzeitzeichenfolge verwenden, ist vom Betriebssystem Ihres lokalen Computers abhängig. Im Folgenden finden Sie Beispiele dafür, wie Sie `date` aus coreutils von GNU unter Linux oder PowerShell unter Windows verwenden, um eine im ISO 8601-Format formatierte Datums- und Uhrzeitzeichenfolge zu erstellen.

coreutils

```
date '+%Y-%m-%dT%H:%M:%SZ'
```

PowerShell

```
(Get-Date).ToString("yyyy-MM-ddTH:mm:ssZ")
```

Wenn Sie mit Systems Manager-API-Vorgängen arbeiten, müssen Sie möglicherweise zu Berichts- oder Fehlerbehebungszwecken historische Datums- und Uhrzeitzeichenfolgen erstellen. Im Folgenden finden Sie Beispiele dafür, wie Sie benutzerdefinierte, im ISO 8601-Format formatierte historische Datums- und Uhrzeitzeichenfolgen für den AWS -Tools für PowerShell und die AWS Command Line Interface (AWS CLI) erstellen und verwenden.

AWS CLI

- Rufen Sie die letzte Woche des Befehlsverlaufs für ein SSM-Dokument ab.

```
lastWeekStamp=$(date '+%Y-%m-%dT%H:%M:%SZ' -d '7 days ago')

docFilter='{"key":"DocumentName","value":"AWS-RunPatchBaseline"}'
timeFilter='{"key":"InvokedAfter","value":'\"$lastWeekStamp\"'}'

commandFilters=[$docFilter,$timeFilter]

aws ssm list-commands \
    --filters $commandFilters
```

- Rufen Sie die letzte Woche des Automatisierungsausführungsverlaufs ab.

```
lastWeekStamp=$(date '+%Y-%m-%dT%H:%M:%SZ' -d '7 days ago')

aws ssm describe-automation-executions \
    --filters Key=StartTimeAfter,Values=$lastWeekStamp
```

- Rufen Sie den letzten Monat des Sitzungsverlaufs ab.

```
lastWeekStamp=$(date '+%Y-%m-%dT%H:%M:%SZ' -d '30 days ago')

aws ssm describe-sessions \
```

```
    --state History \
    --filters key=InvokedAfter,value=$lastWeekStamp
```

AWS -Tools für PowerShell

- Rufen Sie die letzte Woche des Befehlsverlaufs für ein SSM-Dokument ab.

```
$lastWeekStamp = (Get-Date).AddDays(-7).ToString("yyyy-MM-ddTH:mm:ssZ")

$docFilter = @{
    Key="DocumentName"
    Value="AWS-InstallWindowsUpdates"
}
$timeFilter = @{
    Key="InvokedAfter"
    Value=$lastWeekStamp
}

$commandFilters = $docFilter,$timeFilter

Get-SSMCommand `
    -Filters $commandFilters
```

- Rufen Sie die letzte Woche des Automatisierungsausführungsverlaufs ab.

```
$lastWeekStamp = (Get-Date).AddDays(-7).ToString("yyyy-MM-ddTH:mm:ssZ")

Get-SSMAutomationExecutionList `
    -Filters @{Key="StartTimeAfter";Values=$lastWeekStamp}
```

- Rufen Sie den letzten Monat des Sitzungsverlaufs ab.

```
$lastWeekStamp = (Get-Date).AddDays(-30).ToString("yyyy-MM-ddTH:mm:ssZ")

Get-SSMSession `
    -State History `
    -Filters @{Key="InvokedAfter";Value=$lastWeekStamp}
```

Anwendungsfälle und bewährte Methoden

In diesem Thema werden allgemeine Anwendungsfälle und bewährte Methoden für AWS Systems Manager Tools aufgeführt. Wo verfügbar, sind in diesem Thema auch Links zu relevanten Blogbeiträgen und technischer Dokumentation enthalten.

> **Note**
>
> Die Abschnittstitel sind aktive Links zum entsprechenden Abschnitt in der technischen Dokumentation.

Automation

- Erstellen Sie Self-Service-Automation-Runbooks für Infrastruktur.
- Verwenden Sie Automation, ein Tool in AWS Systems Manager, um das Erstellen von Amazon Machine Images (AMIs) aus AWS Marketplace oder benutzerdefinierten Dokumenten zu vereinfachenAMIs, indem Sie öffentliche Systems Manager Manager-Dokumente (SSM-Dokumente) verwenden oder Ihre eigenen Workflows erstellen.
- Erstellen und verwalten Sie AMIs mithilfe der `AWS-UpdateLinuxAmi` und `AWS-UpdateWindowsAmi` Automation-Runbooks oder mithilfe benutzerdefinierter Automation-Runbooks, die Sie erstellen.

Compliance

- Aus Sicherheitsgründen empfehlen wir, dass Sie die von Ihren verwalteten Knoten verwendete AWS Identity and Access Management (IAM-) Rolle aktualisieren, um die Fähigkeit des Knotens, die API-Aktion zu verwenden, einzuschränken. PutComplianceItems Diese API-Aktion registriert einen Compliance-Typ und andere Compliance-Details auf einer bestimmten Ressource, z. B. einer EC2 Amazon-Instance oder einem verwalteten Knoten. Weitere Informationen finden Sie unter Konfigurieren von Berechtigungen für die Compliance.

Inventory

- Verwenden Sie Inventory, ein Tool in AWS Systems Manager, mit, AWS Config um Ihre Anwendungskonfigurationen im Laufe der Zeit zu überprüfen.

Maintenance Windows

- Definieren Sie einen Zeitplan zur Ausführung potenziell störender Aktionen auf Ihren Knoten, wie z. B. Betriebssystem-Patches, Treiber-Updates oder Software-Installationen.
- Informationen zu den Unterschieden zwischen State Manager und Maintenance Windows, Tools von AWS Systems Manager, finden Sie unter Auswahl zwischen State Manager und Maintenance Windows.

Parameter Store

- Verwenden SieParameter Store, ein Tool in AWS Systems Manager, um globale Konfigurationseinstellungen zentral zu verwalten.
- Wie AWS Systems ManagerParameter Store verwendet AWS KMS.
- Verweisen Sie auf AWS Secrets Manager Geheimnisse aus Parameter Store Parametern.

Patch Manager

- Verwenden SiePatch Manager, ein Tool in AWS Systems Manager, um Patches in großem Umfang bereitzustellen und die Transparenz der Flottenkonformität auf Ihren Knoten zu erhöhen.
- Integrieren Sie Patch Manager in AWS Security Hub CSPM, um Warnungen zu erhalten, wenn Knoten in Ihrer Flotte nicht konform sind, und überwachen Sie den Patching-Status Ihrer Flotten hinsichtlich der Sicherheit. Die Nutzung von Security Hub CSPM ist kostenpflichtig. Weitere Informationen finden Sie unter – Preise.
- Verwenden Sie jeweils nur eine Methode zum Scannen verwalteter Knoten auf Patch-Compliance, um unbeabsichtigtes Überschreiben von Compliance-Daten zu vermeiden.

Run Command

- Verwalten Sie Instanzen in großem Umfang ohne SSH-Zugriff mit EC2 Run Command.
- Überwachen Sie alle API-Aufrufe, die von oder im Namen eines Tools getätigt wurden Run Command AWS Systems Manager, in oder mithilfe von AWS CloudTrail.
- Wenn Sie einen Befehl mit Run Command senden, schließen Sie keine vertraulichen Informationen ein, die als Klartext formatiert sind, z. B. Passwörter, Konfigurationsdaten oder andere Geheimnisse. Alle Systems Manager Manager-API-Aktivitäten in Ihrem Konto werden in einem S3-Bucket für AWS CloudTrail Protokolle protokolliert. Dies bedeutet, dass jeder Benutzer mit

Zugriff auf den S3-Bucket die Klartextwerte dieser Geheimnisse anzeigen kann. Aus diesem Grund empfehlen wir, `SecureString`-Parameter zu erstellen und zu verwenden, um die sensiblen Daten zu verschlüsseln, die Sie in Ihren Systems-Manager-Operationen verwenden.

Weitere Informationen finden Sie unter Einschränken des Zugriffs auf Parameter Store-Parameter mithilfe von IAM-Richtlinien.

> **Note**
>
> Standardmäßig werden die von an Ihren Bucket übermittelten Protokolldateien durch CloudTrail serverseitige Amazon-Verschlüsselung mit von Amazon S3 verwalteten Verschlüsselungsschlüsseln (SSE-S3) verschlüsselt. Um eine Sicherheitsebene bereitzustellen, die direkt verwaltet werden kann, können Sie stattdessen serverseitige Verschlüsselung mit AWS KMS verwalteten Schlüsseln (SSE-KMS) für Ihre Protokolldateien verwenden. CloudTrail
> Weitere Informationen finden Sie im Benutzerhandbuch unter Verschlüsseln von CloudTrail Protokolldateien mit AWS KMS verwalteten Schlüsseln (SSE-KMS).AWS CloudTrail

- Verwenden Sie die Ziel- und Tempo-Steuerungsfunktionen in Run Command zum Ausführen zwischengespeicherter Befehle.
- Verwenden Sie mithilfe von (IAM-) Richtlinien detaillierte Zugriffsberechtigungen für Run Command (und alle Systems Manager Manager-Tools). AWS Identity and Access Management

Session Manager

- Protokollierung von Sitzungsaktivitäten in Ihrem AWS-Konto mithilfe von AWS CloudTrail.
- Protokollieren Sie Sitzungsdaten in Ihrer AWS-Konto Nutzung von Amazon CloudWatch Logs oder Amazon S3.
- Steuerung des Benutzer-Sitzungszugriffs auf Instances.
- Beschränken des Zugriffs auf Befehle in einer Sitzung.
- Deaktivieren oder Aktivieren der Administratorberechtigungen für das SSM-Benutzerkonto.

State Manager

- Aktualisieren Sie den SSM Agent mindestens einmal pro Monat mit dem vorkonfigurierten `AWS-UpdateSSMAgent`-Dokument.

- (Windows) Laden Sie das PowerShell oder DSC-Modul auf Amazon Simple Storage Service (Amazon S3) hoch und verwenden Sie `AWS-InstallPowerShellModule` es.
- Erstellen Sie Anwendungsgruppen für Ihre Knoten mit Tags. Und dann verwenden Sie den `Targets` Parameter als Zielknoten, anstatt einzelne Knoten IDs anzugeben.
- Beseitigen Sie die von Amazon Inspector erzeugten Ergebnisse mit Systems Manager automatisch.
- Verwenden Sie ein zentrales Konfigurations-Repository für Ihre SSM-Dokumente und geben Sie Dokumente in Ihrer Organisation frei.
- Informationen zu den Unterschieden zwischen State Manager und Maintenance Windows finden Sie unterAuswahl zwischen State Manager und Maintenance Windows.

Verwaltete Knoten

- Systems Manager erfordert genaue Zeitreferenzen, um seine Operationen auszuführen. Wenn in Ihrem Knoten das Datum und die Uhrzeit nicht korrekt festgelegt wurden, stimmen sie möglicherweise nicht mit dem Signaturdatum Ihrer API-Anforderungen überein. Dies kann zu Fehlern oder unvollständiger Funktionalität führen. Beispiel: Knoten mit falschen Zeiteinstellungen werden nicht in die Liste der verwalteten Knoten aufgenommen.

  Informationen zum Einstellen der Uhrzeit auf Ihren Knoten finden Sie unter Zeit für Ihre EC2 Amazon-Instance festlegen.
- Überprüfen Sie auf verwalteten Linux-Knoten die Signatur von SSM Agent.

Weitere Informationen

- Bewährte Methoden für die Sicherheit für Systems Manager

Löschen von Systems Manager Ressourcen und Artefakten

Als bewährte Methode wird empfohlen, Systems Manager Ressourcen und Artefakte zu löschen, wenn Sie keine Daten zu diesen Ressourcen mehr anzeigen oder die Artefakte in irgendeiner Weise verwenden müssen. In der folgenden Tabelle werden die einzelnen Systems-Manager-Tools oder Artefakte sowie ein Link zu weiteren Informationen zum Löschen der von Systems Manager erstellten Ressourcen oder Artefakte aufgeführt.

Funktion oder Artefakt	Details
Application Manager	Sie können keine Anwendung im Applicati on Manager löschen, Sie können aber eine Anwendung aus dem Dienst entfernen, indem Sie die zugrunde liegenden Tags, -Ressourc e Groups, oder AWS CloudFormationStacks löschen.
Automatisierung	Wenn Sie AWS-Ressourcen mithilfe von Systems Manager Automation erstellen , müssen Sie diese Ressourcen manuell löschen, indem Sie die entsprechende AWS-Managementkonsole verwenden. Wenn Sie ein benutzerdefiniertes Runbook erstellt haben, können Sie das zugrunde liegende SSM-Dokument löschen. Weitere Informationen finden Sie unter Löschen benutzerdefinierter SSM-Dokumente.
Change Calendar	Sie können einen Änderungskalender und ein Änderungskalenderereignis löschen. Weitere Informationen erhalten Sie unter Einen Änderungskalender löschen und Löschen eines Change Calendar-Ereignisses.
Change Manager	Sie können eine Änderungsvorlage löschen. Weitere Informationen finden Sie unter Löschen von Änderungsvorlagen.
Compliance	Systems Manager Compliance zeigt automatis ch Konformitätsdaten zu Patch Manager-P atching und State Manager-Verknüpfungen an. Sie können diese Daten nicht löschen. Wenn Sie eine Ressourcendaten-Synchronisierung konfiguriert haben, um Compliance-Daten in einem S3-Bucket zu zentralisieren, können

Funktion oder Artefakt	Details
	Sie die Synchronisierung löschen. Weitere Informationen finden Sie unter Löschen einer Ressource Data Sync für Compliance.
Distributor	Sie können Pakete in Distributor löschen. Weitere Informationen finden Sie unter Löschen eines Distributor-Pakets.
Explorer	Sie können die Verbindung zu den Quellen trennen, von denen Explorer OpsData sammelt. Weitere Informationen finden Sie unter Bearbeiten von Systems-Manager-Explorer-Datenquellen. Sie können auch eine Ressourcendatensynchronisierung löschen, die von Explorer zum Aggregieren von OpsData und OpsItems von mehreren AWS-Regionen und Konten in einen einzigen Amazon Simple Storage Service (Amazon S3)-Bucket verwendet wird. Weitere Informationen finden Sie unter Löschen einer Systems-Manager-Explorer-Ressourcendatensynchronisierung. Informationen zum Löschen eines S3-Buckets finden Sie unter Löschen eines Buckets im Entwicklerhandbuch für Amazon Simple Email Service.
Fleet Manager	Sie können einen verwalteten Knoten nicht mit Fleet Manager löschen. Sie müssen Amazon Elastic Compute Cloud (Amazon EC2) verwenden. Weitere Informationen finden Sie unter Beenden Ihrer Instance (Linux) und Beenden Ihrer Instance (Windows).

Funktion oder Artefakt	Details
Bestand	Sie können die Inventory-Datensammlung stoppen, indem Sie die State Manager-V erknüpfungen löschen, die den Zeitplan und die Ressourcen definieren, aus denen Metadaten gesammelt werden sollen. Weitere Informati onen finden Sie unter Anhalten der Datenerfa ssung und Löschen von Bestandsdaten. Wenn Sie AWS Systems Manager-Inventory nicht mehr zum Anzeigen von Metadaten Ihrer AWS-Ressourcen verwenden möchten, empfehlen wir auch, Ressourcendatensyn chronisierungen zu löschen, die für die Inventardatenerfassung verwendet werden. Weitere Informationen finden Sie unter Löschen einer Inventory Resource Data Sync.
Maintenance Windows	Sie können ein Wartungsfenster, ein Wartungsf ensterziel und eine Aufgabe im Wartungsf enster löschen. Weitere Informationen finden Sie unter Ressourcen für das Wartungsfenster mithilfe der Konsole aktualisieren oder löschen.
OpsCenter	Sie können ein einzelnes OpsItem löschen, indem Sie die API-Operation OpsItem löschen mit dem AWS Command Line Interface- oder dem AWS-SDK aufrufen. Sie können ein OpsItem nicht in der AWS-Managementkonsole löschen. Weitere Informationen finden Sie unter Löschen Sie OpsItems.
Parameter Store	Sie können einen Parameter löschen, den Sie erstellt haben. Weitere Informationen finden Sie unter Löschen von Parametern in Parameter Store.

Funktion oder Artefakt	Details
Patch Manager	Sie können eine benutzerdefinierte Patch-Baseline löschen. Weitere Informationen finden Sie unter Aktualisieren oder Löschen einer benutzerdefinierten Patch-Baseline.
Quick Setup	Sie können mit Quick Setup erstellte Zuordnungen löschen. Die Zuordnungen werden gespeichert und von State Manager verarbeitet. Weitere Informationen finden Sie unter Löschen von Zuordnungen.
Run Command	Nachdem die Verarbeitung eines Befehls abgeschlossen ist, werden Informationen darüber in Befehls-Verlauf-Registerkarte gespeichert. Sie können keine Informationen aus der Befehls-Verlauf-Registerkarte löschen.
Automatisierung	Nachdem die Verarbeitung einer Automatisierung abgeschlossen ist, werden Informationen darüber in der Registerkarte Ausführungen gespeichert. Sie können keine Informationen aus der Registerkarte Ausführungen löschen.
Serviceverknüpfte Rolle	Systems Manager erstellt automatisch dienstverknüpfte Rollen für einige Tools. Sie können diese Rollen löschen. Weitere Informationen finden Sie unter Löschen einer `AWSServiceRoleForAmazonSSM` -serviceverknüpften Rolle für Systems Manager.
Session Manager	Session Manager speichert keine Daten zu Ihren Ressourcen, nachdem Sie eine Sitzung beendet haben. Weitere Informationen zum Beenden einer Sitzung finden Sie unter Eine Sitzung beenden.

Funktion oder Artefakt	Details
SSM Agent	Sie können SSM Agent manuell von Ihren Knoten deinstallieren. Weitere Informationen finden Sie unter den folgenden Themen. • Linux: Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für Linux • macOS: Manuelles Installieren und Deinstallieren SSM Agent auf EC2 Instanzen für macOS • Windows Server: Öffnen Sie Bedienfeld und wählen Sie dann Programme hinzufügen/entfernen.
State Manager	Sie können eine Verknüpfung löschen. Weitere Informationen finden Sie unter Löschen von Zuordnungen.
Systems Manager-Dokumentenservice	Sie können keine Runbooks löschen, die von AWS oder dem AWS -Support bereitgestellt werden, aber Sie können benutzerdefinierte Runbooks löschen. Weitere Informationen finden Sie unter Löschen benutzerdefinierter SSM-Dokumente.

Auswahl zwischen State Manager und Maintenance Windows

State Managerund Maintenance Windows können mit beiden Tools einige ähnliche Arten von Updates auf Ihren verwalteten Knoten durchführen. AWS Systems Manager Welche Option Sie wählen, hängt davon ab, ob Sie die System-Compliance automatisieren oder zeitkritische Aufgaben mit hoher Priorität während der von Ihnen angegebenen Zeiträume ausführen müssen.

State Manager und Maintenance Windows: Hauptanwendungsfälle

State Manager, ein Tool in AWS Systems Manager, legt die Zielzustandskonfiguration für verwaltete Knoten und AWS Ressourcen in Ihrem fest und verwaltet sie AWS-Konto. Sie können Kombinationen

von Konfigurationen und Zielen als Zuordnungsobjekte definieren. State Managerist das empfohlene Tool, wenn Sie alle verwalteten Knoten in Ihrem Konto in einem konsistenten Zustand halten, Amazon EC2 Auto Scaling verwenden möchten, um neue Knoten zu generieren, oder wenn Sie strenge Compliance-Berichtsanforderungen für die verwalteten Knoten in Ihrem Konto haben möchten.

Die wichtigsten Anwendungsfälle für State Manager lauten wie folgt:

- Auto-Scaling-Szenarien: State Manager kann alle neuen Knoten, die innerhalb eines Kontos gestartet werden, entweder manuell oder über Auto-Scaling-Gruppen überwachen. Wenn im Konto Verknüpfungen vorhanden sind, die auf diesen neuen Knoten ausgerichtet sind (über Tags oder alle Knoten), wird diese bestimmte Zuordnung automatisch auf den neuen Knoten angewendet.
- Compliance-Berichtswesen: State Manager kann das Compliance-Berichtswesen der gewünschten Status für Ressourcen in Ihrem Konto steuern.
- Unterstützt alle Knoten: State Manager kann alle Knoten innerhalb eines bestimmten Kontos anvisieren.

Ein Wartungsfenster führt eine oder mehrere Aktionen auf AWS -Ressourcen innerhalb eines bestimmten Zeitfensters aus. Sie können ein einziges Wartungsfenster mit Start- und Endzeiten definieren. Sie können mehrere Aufgaben angeben, die in diesem Wartungsfenster ausgeführt werden sollen. Verwenden Sie Maintenance Windows ein Tool AWS Systems Manager, wenn Ihre Operationen mit hoher Priorität das Patchen Ihrer verwalteten Knoten, das Ausführen mehrerer Arten von Aufgaben auf Ihren Knoten während eines Aktualisierungszeitraums oder die Steuerung umfassen, wann Aktualisierungsvorgänge auf Ihren Knoten ausgeführt werden können.

Die wichtigsten Anwendungsfälle für Maintenance Windows lauten wie folgt:

- Ausführen mehrerer Dokumente: Wartungsfenster können mehrere Aufgaben ausführen. Jede Aufgabe kann einen anderen Dokumenttyp verwenden. Dadurch können Sie komplexe Workflows mit unterschiedlichen Aufgaben innerhalb eines einzigen Wartungsfensters erstellen.
- Patching: Ein Wartungsfenster kann Patching-Unterstützung für alle verwalteten Knoten in einer einzelnen Region bieten, die mit einem bestimmten Tag oder einer bestimmten Ressourcengruppe versehen sind. Da das Patchen normalerweise das Herunterfahren von Knoten (z. B. das Entfernen von Knoten aus einem Load Balancer), das Patchen und die Nachbearbeitung (das Zurücksetzen von Knoten in die Produktion) umfasst, kann das Patchen als eine Reihe von Aufgaben innerhalb eines bestimmten Patch-Zeitfensters durchgeführt werden.

> **Note**
>
> Wenn Sie ein Wartungsfenster verwenden, ist Ihr Patching-Vorgang auf eine einzige Region in einem einzigen Konto beschränkt. Mit einer in Quick Setup, einem Tool in Systems Manager, erstellten Patch-Richtlinie können Sie stattdessen das Patching für einige oder alle Konten und Regionen in einer in AWS Organizations erstellten Organisation konfigurieren. Weitere Informationen finden Sie unter Patch-Richtlinienkonfigurationen in Quick Setup.

- Fensteraktionen: Wartungsfenster können einen oder mehrere Aktionssätze innerhalb eines bestimmten Zeitfensters starten. Wartungsfenster werden nicht außerhalb dieses Fensters gestartet. Bereits gestartete Aktionen werden bis zum Abschluss fortgesetzt, auch wenn sie außerhalb des Zeitfensters abgeschlossen werden.

In der folgenden Tabelle werden die wichtigsten Features von State Manager und Maintenance Windows verglichen.

Feature	State Manager	Maintenance Windows
CloudFormation Integration	AWS CloudFormation Vorlagen unterstützen State Manager Assoziationen.	CloudFormation Vorlagen unterstützen Wartungsfenster, Fensterziele und Fensterau fgaben.
Compliance	Jede State Manager-Zuordnung meldet die Compliance in Bezug auf den erforderlichen Status der Zielressource. Sie können das Compliance-Dashboard verwenden, um die gemeldete Compliance zu aggregieren und anzuzeigen.	Nicht zutreffend.
Integration der Konfigura tionsverwaltung	State Managerunterstützt externe Targeted State-	Nicht zutreffend.

Feature	State Manager	Maintenance Windows
	Lösungen wie Microsoft PowerShell Desired State Configuration (DSC), Ansible Playbooks und Chef Rezepte. Sie können State Manager-Zuordnungen verwenden, um zu testen, ob die Konfigurationsverwaltungs-Lösungen funktionieren, und um deren Konfigurationsänderungen auf Ihre Knoten anzuwenden, wenn Sie bereit sind.	
Dokumente	State Manager-Konfigurationen können als Richtliniendokumente (zum Sammeln von Bestandsinformationen), Automation-Runbooks, für AWS -Ressourcen wie Amazon Simple Storage Service (Amazon S3)-Buckets oder Systems-Manager-Befehlsdokumente (SSM-Dokumente) für verwaltete Knoten definiert werden.	Maintenance Windows-Konfigurationen können als Automatisierungsdokumente (mehrstufige Aktionen mit optionalen Genehmigungs-Workflows) oder SSM-Dokumente (gewünschter Status für verwaltete Knoten) definiert werden.

Feature	State Manager	Maintenance Windows
Überwachung	State Manager überwacht Änderungen in der Konfiguration, der Zuordnung oder dem Zustand eines Knotens (z. B. neue Knoten, die online geschaltet werden). Wenn State Manager diese Änderungen erkennt, wird die gegebene Zuordnung erneut auf die Knoten angewendet, auf die diese Zuordnung ursprünglich abzielte.	Nicht zutreffend.
Prioritäten innerhalb von Aufgaben	Nicht zutreffend.	Aufgaben innerhalb eines Wartungsfensters können mit einer Priorität versehen werden. Alle Aufgaben mit derselben Priorität werden parallel ausgeführt. Aufgaben mit niedrigeren Prioritäten werden ausgeführt, nachdem Aufgaben mit höheren Prioritäten einen endgültigen Status erreicht haben. Es gibt keine Möglichkeit, Aufgaben bedingt auszuführen. Nachdem eine Aufgabe mit höherer Priorität den endgültigen Status erreicht hat, wird die nächste Prioritätsaufgabe unabhängig vom Status der vorherigen Aufgabe ausgeführt.

Feature	State Manager	Maintenance Windows
Sicherheitskontrollen	State Manager unterstützt zwei Sicherheitskontrollen bei der Bereitstellung von Konfigurationen in einer großen Flotte. Sie können die maximale Nebenläufigkeit verwenden, um festzulegen, auf wie viele nebenläufige Knoten oder Ressourcen die Konfiguration angewendet werden soll. Sie können eine maximale Fehlerrate festlegen, die verwendet werden kann, um die State Manager-Verknüpfung zu pausieren, wenn eine bestimmte Anzahl oder ein bestimmter Prozentsatz von Fehlern in der gesamten Flotte auftritt.	Wartungsfenster unterstützen zwei Sicherheitskontrollen bei der Bereitstellung von Konfigurationen in einer großen Flotte. Sie können die maximale Nebenläufigkeit verwenden, um festzulegen, auf wie viele nebenläufige Knoten oder Ressourcen die Konfiguration angewendet werden soll. Sie können eine maximale Fehlerrate festlegen, die verwendet werden kann, um die Aktionen in einem Wartungsfenster zu pausieren, wenn eine bestimmte Anzahl oder ein Prozentsatz von Fehlern in der gesamten Flotte auftritt.

Feature	State Manager	Maintenance Windows
Planung	Sie können State Manager-Verknüpfungen bei Bedarf, in einem bestimmten Cron-Intervall, mit einer bestimmten Rate oder einmal nach der Erstellung ausführen. Dies ist nützlich, wenn Sie den gewünschten Status Ihrer Ressourcen konsistent und zeitnah aufrechterhalten möchten. **Important** Cron-Ausdrücke für State Manager-Zuordnungen unterstützen das Monatsfeld nicht, wie `03` oder `MAR` für den Monat März. Wenn Sie monatliche oder vierteljährliche Konfigurationsupdates benötigen, kann ein Wartungsfenster Ihre Anforderungen am besten erfüllen. Weitere Informationen finden Sie unter Referenz: Cron- und Rate-Ausdrücke für System Manager.	Wartungsfenster unterstützen mehrere Planungsoptionen, einschließlich `at`-Ausdrücken (z. B. `"at(2021-07-07T13:15:30)"`), Cron- und Rate-Ausdrücke, Cron mit Offsets und Start- und Endzeiten für die Ausführung von Wartungsfenstern sowie Grenzzeiten, um anzugeben, wann die Planung innerhalb eines bestimmten Zeitfensters beendet werden soll.

Feature	State Manager	Maintenance Windows
Targeting	State Manager-Zuordnungen können auf einen oder mehrere Knoten mithilfe der Knoten-ID, des Tags oder der Ressourcengruppe abzielen. State Manager kann auf alle verwalteten Knoten innerhalb eines bestimmten Kontos abzielen.	Wartungsfenster können mithilfe von Knoten IDs, Tags oder Ressourcengruppen auf einen oder mehrere Knoten abzielen.

Feature	State Manager	Maintenance Windows
Aufgaben innerhalb von Wartungsfenstern	Nicht zutreffend.	Wartungsfenster können eine oder mehrere Aufgaben unterstützen, bei denen jede Aufgabe auf ein bestimmte s Automation-Runbook oder eine Command-Dokumentat kion abzielt. Alle Aufgaben innerhalb eines Wartungsf ensters werden parallel ausgeführt, es sei denn, für unterschiedliche Aufgaben sind unterschiedliche Priorität en festgelegt. Insgesamt unterstützen die Wartungsfenster vier Aufgabentypen: • AWS Systems Manager Run Command-Befehle • AWS Systems Manager Workflows zur Automatis ierung • AWS Lambda Funktionen • AWS Step Functions Aufgaben

Ähnliche Informationen

Die folgenden verwandten Ressourcen bieten Ihnen nützliche Informationen für die Arbeit mit diesem Service.

Preise

Einige Systems-Manager-Tools sind gebührenpflichtig. Weitere Informationen finden Sie unter AWS Systems Manager Preise.

AWS Systems Manager-Dokumentationsbibliothek

AWS Systems Manager-Dokumentation – Zugriff auf die gesamte Benutzerdokumentation für Systems Manager, einschließlich AWS AppConfig, Vorfallmanager und AWS Systems Manager für SAP.

AWS re:Post

AWS re:Post – AWS-verwalteter Frage-und-Antwort-Dienst (F & A), der von Experten geprüfte Crowdsourcing-Antworten auf Ihre technischen Fragen bietet.

AWS-Blog und Podcast

Lesen Sie Blogbeiträge über Systems Manager in der Kategorie AWS-Management-Tools und andere Beiträge, die mit #Systems Manager getaggt sind.

Servicekontingente

Überprüfen Sie Systems Manager Service Quotas im Allgemeine Amazon Web Services-Referenz. Wenn nicht anders angegeben, gilt jedes Kontingent für eine Region in einem AWS-Konto.

Referenz zur Serviceautorisierung für Systems Manager

In der AWS Referenz zur Serviceautorisierung finden Sie Informationen zu den Aktionen, Ressourcen und Bedingungskontextschlüsseln, die Sie in AWS Identity and Access Management (IAM)-Richtlinien für Systems Manager verwenden können.

AWS Systems Manager Service Level Agreement

Das AWS Systems Manager Service Level Agreement (SLA) ist eine Richtlinie, die die Verwendung von Systems Manager regelt und für jedes AWS-Konto, das Systems Manager verwendet, separat gilt.

Allgemeine AWS-Ressourcen

Die folgenden allgemeinen Ressourcen bieten Ihnen nützliche Informationen für die Arbeit mit AWS.

- Kurse und Workshops – Links zu rollenbasierten und speziellen Kursen sowie Übungen im Selbststudium zur Verbesserung Ihrer AWS-Kompetenzen und Erweiterung Ihrer praktischen Erfahrung.
- AWS-Entwicklerzentrum – Entdecken Sie Tutorials, laden Sie Tools herunter und erfahren Sie mehr über Veranstaltungen für AWS-Entwickler.
- AWS-Entwickler-Tools – Links zu Entwickler-Tools, SDKs, IDE-Toolkits und Befehlszeilen-Tools für die Entwicklung und Verwaltung von AWS-Anwendungen.
- Ressourcenzentrum für die ersten Schritte – Erfahren Sie, wie Sie Ihr AWS-Konto einrichten, der AWS-Community beitreten und Ihre erste Anwendung starten.
- Praktische Tutorials – Schritt-für-Schritt-Anleitungen zum Starten Ihrer ersten Anwendung auf AWS.
- AWS Whitepaper – Links zu einer umfangreichen Liste technischer AWS-Whitepaper zu Themen wie Architektur, Sicherheit und Wirtschaftlichkeit. Diese Whitepaper wurden von AWS-Lösungsarchitekten und anderen technischen Experten verfasst.
- AWS -Support-Center – Hub für die Erstellung und Verwaltung Ihrer AWS -Support-Fälle. Stellt darüber hinaus Links zu weiteren nützlichen Ressourcen bereit, beispielsweise Foren, häufig gestellten technischen Fragen, Status der Service-Integrität und AWS Trusted Advisor.
- Support – Primäre Website für Informationen zu Support, einem persönlichen und reaktionsschnellen Support-Channel, der Sie bei der Erstellung und Ausführung von Anwendungen in der Cloud unterstützt.
- Kontakt – Zentraler Kontaktpunkt für Fragen zu AWS-Abrechnung, Konten, Ereignissen Missbrauch und anderen Problemen.
- Nutzungsbedingungen für die AWS-Website – Detaillierte Informationen zu unseren Copyright- und Markenbestimmungen, Ihrem Konto, den Lizenzen und anderen Themen.

Dokumentverlauf

In der folgenden Tabelle werden die wichtigen Änderungen an der Dokumentation seit der letzten Version von AWS Systems Manager beschrieben. Für Benachrichtigungen über Aktualisierungen dieser Dokumentation können Sie einen RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
AWS Systems ManagerChange Managersteht neuen Kunden nicht mehr offen.	AWS Systems ManagerChange Managersteht neuen Kunden nicht mehr offen. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.	7. November 2025
Neue Versionen der AWS Lambda-Erweiterung Parameters and Secrets	Neue Versionen der AWS -Parameter und -Secrets-Lambda-Erweiterung sind jetzt für alle zuvor unterstützten Regionen und Architekturen verfügbar. Darüber hinaus ARM64 wurde Mac with Apple silicon Architekturunterstützung für die Region Asien-Pazifik (Neuseeland) (ap-southeast-6) hinzugefügt.	22. Oktober 2025
Aktualisierte Python- und PowerShell Runtime-Unterstützung	Wir haben die Listen und Beispiele der „Unterstützten“ mit den Laufzeiten aktualisiert, die jetzt offiziell von Systems Manager unterstützt werden. Dazu gehören Python 3.10, Python 3.11 und PowerShel	16. Oktober 2025

	l 7.4 (dotnet8). Frühere Versionen werden nicht mehr unterstützt.	
Patch ManagerUnterstützung für IPv6 Umgebungen	Patch Managerbietet jetzt Unterstützung für Internet Protocol Version 6 (IPv6) -Umgebungen. Weitere Informationen finden Sie unter Verbessern der Sicherheit von EC2 Instances mithilfe von VPC-Endpunkten für Systems Manager und Patchen eines Servers in einer IPv6 reinen Umgebung.	15. Oktober 2025
AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein.	AWS Systems ManagerChange Managerwird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie es nutzen möchtenChange Manager, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter Änderung der AWS Systems ManagerChange Manager Verfügbarkeit.	7. Oktober 2025

Neue Versionen der AWS Lambda-Erweiterung Parameters and Secrets	Neue Versionen der AWS -Parameter und -Secrets-Lambda-Erweiterung sind jetzt für alle zuvor unterstützten Regionen und Architekturen verfügbar. Darüber hinaus wurde Support für die Region Asien-Pazifik (Neuseeland) (ap-southeast-6) hinzugefügt, und zwar nur für die x86_64- und x86-Architekturen.	30. September 2025
Verwaltete Richtlinie aktualisiert AWSSystemsManagerJustInTimeAccessTokenPolicy	Systems Manager hat die verwaltete Richtlinie aktualisiertAWSSystemsManagerJustInTimeAccessTokenPolicy , um Datenkanalberechtigungen für die Sitzungsverwaltung bereitzustellen. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	25. September 2025

Aktualisierte verwaltete Richtlinie: AWSQuickSetupStartStopInstancesExecutionPolicy	Systems Manager hat die verwaltete Richtlinie aktualisiert `AWSQuickSetupStartStopInstancesExecutionPolicy` , um die Berechtigungen für die Quick Setup Scheduler-Konfiguration zu verfeinern. Die Richtlinie bietet jetzt spezifischere Berechtigungen für das Starten und Stoppen von EC2 Amazon-Instances, den Zugriff auf Änderungskalender und die Ausführung von Automatisierungsdokumenten mit verbesserten Sicherheitsbedingungen. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	12. September 2025

Aktualisierte verwaltete Richtlinie: AWSQuickSetupStartSSMAssociationsExecutionPolicy	Systems Manager hat die verwaltete Richtlinie aktualisiertAWSQuickSetupStartSSMAssociationsExecutionPolicy , um das Automatisierungsdokument von AWSQuickSetupType-StartSSMAssociations zu zu zu ändernAWSQuickSetupType-Scheduler-ChangeCalendarState . Mit diesem Update ändert sich der Zweck der Richtlinie von der Einrichtung von SSM-Verknüpfungen zur Verwaltung von Kalenderänderungen bei geplanten Vorgängen. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	12. September 2025

Drei aktualisierte verwaltete Richtlinien	Systems Manager hat drei verwaltete Richtlinien aktualisiert, damit Automatisierungsausführungen auf zusätzlichen Systems Manager Manager-Ressourcen gestartet werden können: `AWSQuickSetupSSMLifecycleManagementExecutionPolicy` `AWSQuickSetupPatchPolicyPermissionsBoundary` , und`AWSQuickSetupManagedInstanceProfileExecutionPolicy` . Quick Setup Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	12. September 2025
Aktualisierte verwaltete Richtlinie: Amazon SSMAutomation Role	Systems Manager hat die verwaltete Richtlinie `AmazonSSMAutomationRole` aktualisiert, um die Berechtigung `ssmmessages:OpenDataChannel` hinzuzufügen. Mit dieser Berechtigung können Automation-Runbooks Kommunikationskanäle für sitzungsbasierte Vorgänge einrichten. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	11. September 2025

Neues Thema: Grundlegendes zum Ruhezustand von SSM Agent	In diesem neuen Thema wird erklärt, dass der Ruhezustand ein Betriebsmodus ist, der auftritt, wenn der Agent nicht ordnungsgemäß mit dem Systems Manager-Service kommunizieren kann. Während des Ruhezustands reduziert der Agent seine Kommunikationshäufigkeit und wechselt in einen Bereitschaftszustand. Einzelheiten zum Ruhezustand von Agenten finden Sie unter Grundlegendes zum Ruhezustand von SSM Agent.	2. September 2025
Aktualisierte verwaltete Richtlinie: AWSSystems ManagerJustInTimeAccessServicePolicy	Systems Manager hat die verwaltete Richtlinie `AWSSystemsManagerJustInTimeAccessServicePolicy` aktualisiert, um Tagging-Berechtigungen für die Automatisierungsausführung hinzuzufügen, sodass Kunden die Bedienerberechtigungen auf bestimmte Tags einschränken können. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	25. August 2025

Die Beschreibungen der Registerkarten mit Knotendetails wurden aktualisiert (vereinheitlichte Konsole)	Wir haben die Inhaltsbeschreibungen auf den Registerkarten auf der Seite mit den Knotendetails in der vereinheitlichten Konsole aktualisiert (Registerkarten: Tags, Inventar, Zuordnung, Patches, Konfigurationskonformität). Weitere Informationen finden Sie unter Was ist die vereinheitlichte Konsole?.	20. August 2025
Neues Thema mit bewährten Sicherheitsmethoden für die Installation von SSM Agent	Das neue Thema Bewährte Methoden zur SSM Agent Installation erklärt, wie wichtig es ist, je nach Maschinentyp, EC2 Instanz oder hybridaktivierter Maschine die richtige Methode für die Installation des Agenten zu verwenden.	18. August 2025

Neue konsolenbasierte Automation-Features	AWS Systems Manager Die Automatisierung hat erweiterte Ausführungsfunktionen mit automatischen Wiederholungen bei gedrosselten Vorgängen und der Möglichkeit, Automatisierungsausführungen erneut auszuführen, eingeführt. Das Feature zur automatischen Drosselung von Wiederholungsversuchen trägt dazu bei, eine zuverlässige Ausführung der Automatisierung in großem Umfang sicherzustellen, indem API-Aufrufe, bei denen Drosselungsgrenzen auftreten, automatisch wiederholt werden. Dabei können Zeitlimits von bis zu 3.600 Sekunden konfiguriert werden. (Alle Automation-Aktionen außer aws:executeScript, automatische Drosselungswiederholung). Darüber hinaus können Sie abgeschlossene Automatisierungsausführungen jetzt problemlos mit identischen oder geänderten Parametern erneut ausführen, um Wiederholungen zu vereinfachen und den Betriebsaufwand zu reduzieren, ohne Konfigurationen manuell neu erstellen zu müssen. Weitere Informati	14. August 2025

onen finden Sie unter den folgenden Themen:

- Konfigurieren der automatischen Wiederholung für gedrosselte Vorgänge
- Wiederholtes Ausführen von Automatisierungsausführungen
- Ausführung von Automatisierungen in mehreren Endkonten AWS-Regionen

Aktualisierungen für die AWSQuick SetupDeployment RolePolicy verwaltete Richtlinie	Systems Manager hat die verwaltete Richtlinie `AWSQuickSetupDeployment RolePolicy` aktualisiert, um weiteren `AWSQuickSetupType-*` Automation-Runbooks zuordnungsbezogene Berechtigungen zu gewähren und Quick Setup das Hinzufügen und Entfernen zusätzlicher IAM-Richtlinien zu ermöglichen. Darüber hinaus haben wir die Dokumentation für `AWSQuickSetupDeployment RolePolicy` aktualisiert, um detailliertere Beschreibungen der für Konfigurationsverwaltungsvorgänge in Quick Setup gewährten Berechtigungen bereitzustellen. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	12. August 2025

Neues Thema: Datenperimeter in AWS Systems Manager	Sie können Datenperimeterrichtlinien für AWS diensteigene Ressourcen konfigurieren, auf die Systems Manager zugreift. Datenperimeter in AWS Systems Manager beschreibt den Zweck des von Systems Manager verwalteten `ssm-document-categories-` *`region`* S3-Buckets. Dieser Bucket enthält Metadaten zu Dokumentkategorien, mit deren Hilfe SSM-Dokumente in der Konsole organisiert und klassifiziert werden können.	12. August 2025
Neue verwaltete Richtlinie für Scheduler: Quick Setup AWSQuick SetupStartStopInstancesExecutionPolicy	Systems Manager hat eine neue verwaltete Richtlinie veröffentlicht`AWSQuickSetupStartStopInstancesExecutionPolicy` , die Berechtigungen Quick Setup zum planmäßigen Starten und Stoppen von EC2 Amazon-Instances bietet. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	12. August 2025

Neue AWS verwaltete Richtlinie fürQuick Setup: AWSQuickSetupStart SSMAssociationsExecutionPolicy	Systems Manager hat eine neue verwaltete Richtlinie `AWSQuickSetupStartSSMAssociationsExecutionPolicy` veröffentlicht, mit der das Quick Setup das Automation-Runbook `AWSQuickSetupType-StartSSMAssociations` ausführen kann. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	12. August 2025
Neue Anleitung zur Behebung von Fehlern bei der zypper-Paketsperrabhängigkeit auf SLES-verwalteten Knoten	Es wurde eine umfassende Anleitung zur Behebung von Fehlern bei der zypper-Paketsperrabhängigkeit hinzugefügt, die bei `Install`-Vorgängen in Patch Manager auf SUSE Linux Enterprise Server-verwalteten Knoten auftreten. Der neue Abschnitt Problem: Fehler bei der zypper-Paketsperrabhängigkeit auf SLES-verwalteten Knoten enthält detaillierte Informationen zur Identifizierung, Diagnose und Lösung von Paketsperrkonflikten, die eine erfolgreiche Patch-Installation verhindern.	28. Juli 2025

Verbesserte Anleitung zur Fehlerbehebung bei übersprungenen Inventarzuordnungen	Der Abschnitt zur Fehlerbehebung bei Problemen mit Überspringen wurde aktualisiert und bietet nun umfassende Hinweise zur prioritären Behandlung von Inventarzuordnungen. Die erweiterte Dokumentation enthält jetzt detaillierte Informationen zu Quick Setup Inventarzuordnungen, expliziten Zielgruppenszenarien und step-by-step Lösungsverfahren für den Umgang mit widersprüchlichen Inventarzuordnungen. Zu den Verbesserungen gehören: • Erläuterung der Prioritätsreihenfolge von Inventarzuordnungen (Quick Setup, explizite und globale Zuordnungen) • Typische Szenarien mit spezifischen Beispielen und erwarteten Ergebnissen • Detaillierte Lösungsschritte für das Entfernen von Quick Setup-Konfigurationen bei der Verwendung benutzerdefinierter Inventarzuordnungen • Wichtige Hinweise zum Verhalten bei der Datenerfa	24. Juli 2025

	ssung und zur Rangfolge von Zuordnungen	
Aktualisierte verwaltete Richtlinie: AWS-SSM-RemediationAutomation-ExecutionRolePolicy	Systems Manager hat die verwaltete Richtlinie aktualisiertAWS-SSM-RemediationAutomation-ExecutionRolePolicy , um die Sicherheitslage der SSM: StartAutomationExecution API zu verbessern, indem Berechtigungen sowohl für die Ressourcentypen „Dokument“ als auch für „Automationsausführung“ erforderlich sind. Mit dieser Aktualisierung soll die Sicherheitslage der SSM: StartAutomationExecution API verbessert werden, indem Berechtigungen sowohl für die Ressourcentypen „Dokument“ als auch für „Automatisierungsausführung“ erforderlich sind, wodurch umfassendere und detailliertere Berechtigungen für die automatische Ausführung von Problembehebungen bereitgestellt werden. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	16. Juli 2025

Aktualisierte verwaltete Richtlinie: AWS-SSM-RemediationAutomation-AdministrationRolePolicy	Systems Manager hat die verwaltete Richtlinie `AWS-SSM-RemediationAutomation-AdministrationRolePolicy` aktualisiert, um Verbesserungen der API-Autorisierung für automatische Problembehebungen zu unterstützen. Die aktualisierte Richtlinie erweitert die Berechtigungen für die Ausführung von Aktivitäten, die in Automatisierungsdokumenten definiert sind, und bietet verbesserte Sicherheitskontrollen und Ressourcenzugriffsmuster für Korrektur-Workflows. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	16. Juli 2025

Aktualisierte verwaltete Richtlinie: AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy	Systems Manager hat die verwaltete Richtlinie `AWS-SSM-DiagnosisAutomation-ExecutionRolePolicy` aktualisiert, um detailliertere und genauere Berechtigungen für die Ausführung automatisierter Diagnosen bereitzustellen. Die aktualisierte Richtlinie umfasst erweiterte Beschreibungen für den Zugriff auf Amazon EC2 - und Amazon VPC-Ressourcen, spezifische SSM-Automatisierungsberechtigungen AWS KMS sowie verbesserte IAM-Berechtigungsbeschreibungen mit entsprechenden Ressourceneinschränkungen. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	16. Juli 2025

Aktualisierte verwaltete Richtlinie: AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy	Systems Manager hat die verwaltete Richtlinie `AWS-SSM-DiagnosisAutomation-AdministrationRolePolicy` aktualisiert und enthält nun spezifischere Berechtigungen und Sicherheitsbedingungen für Diagnoseautomatisierungsvorgänge. Die aktualisierte Richtlinie bietet erweiterte Sicherheitskontrollen für die AWS KMS Schlüsselnutzung, den Zugriff auf Amazon S3 S3-Buckets und Rollenannahmen mit strengeren ressourcenbasierten Bedingungen und Einschränkungen auf Kontoebene. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	16. Juli 2025
Dokumentation für Amazon SSMAutomation Role Managed Policy hinzugefügt	Systems Manager hat die Dokumentation für die bestehende verwaltete Richtlinie `AmazonSSMAutomationRole` aktualisiert. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	15. Juli 2025

Verbesserte Sicherheit für SSM-Dokumente mit Interpolation von Umgebungsvariablen	AWS Systems Manager Befehlsdokumente unterstützen jetzt die Interpolation von Umgebungsvariablen bei der Verarbeitung von Parametern. Diese Funktion, die in Schemaversion 2.2 und in SSM Agent Version 3.3.2746.0 oder höher verfügbar ist, trägt dazu bei, Befehlseinschleusungsangriffe zu verhindern, indem Parameterwerte als wörtliche Zeichenketten und nicht als potenziell ausführbare Befehle behandelt werden. Wenn diese Option auf `ENV_VAR` gesetzt ist, erstellt der Agent eine Umgebungsvariable, die im Format `SSM_`*`parameter-name`* benannt ist und den Wert des Parameters enthält. Weitere Informationen finden Sie unter den folgenden Themen: • Dokument-Komponenten • Beispiele für die sichere Parameterverarbeitung • Datenelemente und Parameter • Referenz für das Command-Dokument-Plugin	14. Juli 2025

ExplorerDie Synchronisierung von Ressourcendaten ist in vier Opt-in-Optionen verfügbar AWS-Regionen	Standardmäßig unterstützen konten-/regionsübergreifende Ressourcendaten-Synchronisierungen in Explorer keine Datenaggregation in Opt-in-Regionen. Am 30. Juni 2025 wurde Support für die folgenden Opt-In-Regionen hinzugefügt. • Europa (Milan) • Afrika (Kapstadt) • Middle East (Bahrain) • Asien-Pazifik (Hongkong) Beachten Sie, dass Sie ein delegiertes Administratorkonto nicht verwenden können, um eine Ressourcendatensynchronisierung in Opt-in-Regionen zu erstellen. Sie müssen ein AWS Organizations Verwaltungskonto verwenden. Weitere Informationen finden Sie unter Ressourcendatensynchronisierung zwischen mehreren Konten und Regionen.	30. Juni 2025

Neue Versionen der AWS Lambda-Erweiterung Parameters and Secrets	Neue Versionen der AWS -Parameter und -Secrets-Lambda-Erweiterung sind jetzt für alle unterstützten Regionen und Architekturen verfügbar . Darüber hinaus wurde die Unterstützung für die Region Asien-Pazifik (Taipeh) (ap-east-2) hinzugefügt.	26. Juni 2025
Neues Thema: Change Calendar Integration mit Amazon EventBridge	Das Thema Change CalendarIntegration mit Amazon EventBridge bietet Informationen zu dieser Integration in den folgenden Kategorien: • Timing und Zuverlässigkeit von Ereignissen • Änderungen von Ereignissen und Hinweise zum Timing • Aufeinanderfolgende und sich überschneidende Ereignisse • Best Practices	25. Juni 2025
Update zur Laufzeitunterstützung für die Automation-Aktion 'aws:ExecuteScript'	Der `Runtime` Parameter in der Automatisierungsaktion aws:ExecuteScript unterstützt PowerShell jetzt 7.4 () - Skripte. `dotnet8`	24. Juni 2025

SSM Agent und Patch Manager unterstützen jetzt RHEL 10 und Ubuntu Server 25.04	Sie können jetzt Red Hat Enterprise Linux 10 und Ubuntu Server 25.04 Maschinen mit Systems Manager verwalten, einschließlich Patches für verwaltete Knoten mithilfe von Patch Manager. Eine vollständige Liste der unterstützten Versionen OSs und Versionen finden Sie in den folgenden Themen: • Unterstützte Betriebssysteme für Systems Manager • Unterstützte Betriebssysteme für Patch Manager	23. Juni 2025

Aktualisierungen der Themen zum just-in-time Knotenzugriff	Wir haben die folgenden Inhaltsaktualisierungen im Kapitel Just-in-time Knotenzugriff mithilfe von Systems Manager vorgenommen: • Die Authentifizierungsunterstützung für den just-in-time Knotenzugriff wurde klargestellt. Weitere Informationen finden Sie unter just-in-timeZugriff mit Systems Manager einrichten. • Es wurde klargestellt, dass die AWS KMS Schlüssel, die für die Session Manager Verschlüsselung und RDP-Aufzeichnung beim just-in-time Knotenzugriff verwendet werden, mit dem Tag-Schlüssel `SystemsManagerJustInTimeNodeAccessManaged` und dem Tag-Wert `true` gekennzeichnet werden müssen. Weitere Informationen finden Sie unter Sitzungseinstellungen für den just-in-time Knotenzugriff aktualisieren und S3-Bucket-Verschlüsselung für RDP-Aufzeichnungen konfigurieren. • Die Konfigurationsbenachrichtigungen für just-in-t	17. Juni 2025

	ime Zugriffsanfragen wurden mit zusätzlichen Konfigura tionsdetails aktualisiert.	
Verbesserte RDP-Aufze ichnung für den just-in-time Knotenzugriff	AWS Systems Manager just-in-time Der Knotenzug riff bietet jetzt zusätzliche Informationen zu RDP-Aufze ichnungskonfigurationen. Wenn Sie die AWS KMS Verschlüsselung für S3-Buckets verwenden, in denen Aufzeichnungen gespeiche rt werden, müssen Sie dem `ssm-guiconnect` Dienstpri nzipal Zugriff auf den KMS-Schlüssel gewähren, der für die Bucket-Verschlüsse lung verwendet wird. In dieser Version wird auch ein neuer Aufzeichnungsstatu s `ProcessingError` eingeführt, der auf temporäre Probleme bei der Videovera rbeitung hinweist, die auf Fehler bei der Serviceab hängigkeit oder fehlende Berechtigungen zurückzuf ühren sein könnten. Weitere Informationen finden Sie unter Aufzeichnen von RDP-Verbi ndungen.	13. Juni 2025

Informationen zur Patch-Auswahl und Installation in Patch Manager aktualisiert	Es wurden Informationen hinzugefügt, die erläutern, wie Patch Manager bei der Auswahl und Installation von Patches mit veraltete n Paketen umgeht. Weitere Informationen finden Sie unter So werden Sicherheitspatches ausgewählt und So werden Patches installiert.	12. Juni 2025
Aktualisierte verwaltete Richtlinie fürQuick Setup: AWSQuick SetupDeployomentRolePolicy	Systems Manager hat die verwaltete Richtlinie `AWSQuickSetupDeployomentRolePolicy` aktualisiert, sodass Quick Setup Zuordnungen mithilfe dieser verwalteten Richtlinie anstelle von Inline-Richtlinien erstellen kann. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	10. Juni 2025

Einführende Informationen zu Patch Manager erweitert	Um potenziellen Kunden und Benutzern eine bessere Vorstellung von Patch Manager zu geben, haben wir das Kapitel zu Patch Manager um die folgenden drei Abschnitte erweitert: • Welche Vorteile bietet Patch Manager meiner Organisation? • An wen richtet sich Patch Manager? • Was sind die Hauptfeatures von Patch Manager?	21. Mai 2025
Die verwaltete Richtlinie wurde aktualisiert fürQuick Setup: AWSQuick Setup SSMDeployment RolePolicy	Systems Manager hat die verwaltete Richtlinie `AWSQuickSetupSSMDeploymentRolePolicy` aktualisiert, um Berechtigungen zum Kennzeichnen von IAM-Rollen und Lambda-Ressourcen bereitzustellen, die für die vereinheitlichte Konsole erstellt wurden. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	7. Mai 2025

Optionen zur Featurekonfiguration in der vereinheitlichten Konsole	Systems Manager bietet jetzt Optionen zur Konfiguration von Features bei der Einrichtung der vereinheitlichten Konsole. Bei diesen Featurekonfigurationen können Sie wählen, ob Sie die Standard-Host-Management-Konfiguration (DHMC), die Erfassung von Bestandsmetadaten und automatische SSM Agent-Updates einrichten möchten. Weitere Informationen finden Sie unter Einrichten der vereinheitlichten Systems Manager-Konsole für eine Organisation und Einrichten der vereinheitlichten Systems Manager-Konsole für ein einzelnes Konto und eine Region.	7. Mai 2025

Aktualisierte verwaltete Richtlinie fürQuick Setup: AWSSystems ManagerJustInTimeNodeAccessRolePropagationPolicy	Systems Manager hat die verwaltete Richtlinie aktualisiertAWSSystemsManagerJustInTimeNodeAccessRolePropagationPolicy , um Berechtigungen zum Markieren einer Ressource bereitzustellen, die AWS Resource Access Manager für den just-in-time Knotenzugriff gemeinsam genutzt wird. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	29. April 2025
Aktualisierte verwaltete Richtlinie fürQuick Setup: AWSQuick SetupManageJITNAResources ExecutionPolicy	Systems Manager hat die verwaltete Richtlinie aktualisiert, um Berechtigungen AWSQuickSetupManageJITNAResourcesExecutionPolicy zum Markieren von IAM-Rollen bereitzustellen, die für den just-in-time Knotenzugriff erstellt wurden. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	29. April 2025

Systems Manager führt just-in-time Knotenzugriff ein	AWS Systems Manager unterstützt jetzt eine just-in-time Zugriffsmethode für die Verbindung zu verwalteten Knoten. Diese neue Methode zur Verbindung mit verwaltet en Knoten trägt dazu bei, die Sicherheit Ihrer Knoten zu verbessern, indem temporäre r, zeitgebundener Zugriff auf Knoten nur bei Bedarf gewährt wird, sodass keine permanent en Berechtigungen erforderl ich sind. Darüber hinaus bietet Systems Manager Sitzungsaufzeichnungen für RDP-Sitzungen zu Windows Server Knoten, damit Sie Compliance-Anforderungen erfüllen, Ursachenanalysen durchführen und vieles mehr können. Weitere Informationen finden Sie unter Just-in-time Knotenzugriff mit Systems Manager.	29. April 2025
Neue verwaltete Richtlinien für den just-in-time Knotenzugriff	Systems Manager hat mehrere neue verwaltete Richtlinien zur Unterstützung des just-in-time Knotenzugriffs veröffentlicht. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	29. April 2025

Neue Versionen der AWS Lambda-Erweiterung Parameters and Secrets	Neue Versionen der AWS -Parameter und -Secrets-Lambda-Erweiterung sind jetzt für alle unterstützten Regionen und Architekturen verfügbar.	23. April 2025
Neues Thema: „Worin besteht Compliance in Patch Manager?“	Die Patch-Konformität für verwaltete Knoten in Ihrer Flotte wird weder von AWS noch von Dritten festgeleg t. Das neue Thema Was bedeutet Compliance in Patch Manager? stellt klar, dass Patch-Konformität für einen von Systems Manager verwalteten Knoten bedeutet, dass alle Patches auf dem Knoten installiert wurden, die die in der zugehörigen Patch-Baseline angegeben en Genehmigungskriterien erfüllen. Die Einhaltung der Vorschriften wird von Ihnen in den Regeln, die Sie in einer Patch-Baseline konfigurieren, oder in einer vordefinierten Patch-Baseline AWS , die Sie für Ihre Patching-Operationen auswählen, festgelegt.	11. April 2025

Neue Versionen der AWS Lambda-Erweiterung Parameters and Secrets	Neue Versionen der AWS -Parameter und -Secrets-Lambda-Erweiterung sind jetzt verfügbar. Außerdem wurde die Unterstützung der Erweiterung für die folgenden Regionen hinzugefügt: • Asien-Pazifik (Thailand) (ap-southeast-7) • Mexiko (Zentral) (mx-central-1)	3. April 2025
Neues Thema: Konfigurieren von Berechtigungen für die Compliance	Aus Sicherheitsgründen empfehlen wir, dass Sie die von Ihren verwalteten Knoten verwendete AWS Identity and Access Management (IAM-) Rolle aktualisieren, um die Fähigkeit des Knotens zur Nutzung der PutComplianceItemsAPI-Aktion einzuschränken. Diese API-Aktion registriert einen Compliance-Typ und andere Compliance-Details auf einer bestimmten Ressource, z. B. einer EC2 Amazon-Instance oder einem verwalteten Knoten. Weitere Informationen finden Sie unter Konfigurieren von Berechtigungen für Compliance.	11. März 2025

Thema aktualisiert: Anzeigen von Knoteninformationen	In der vereinheitlichten Konsole für Systems Manager wird auf der Seite „Knoteninformationen überprüfen“ die Anzahl der verwalteten und nicht verwalteten EC2 Instanzen in Ihrer Organisation oder Ihrem Konto angezeigt. Wir haben klargestellt, dass Systems Manager eine EC2 Instanz in die Zählung „Nicht verwaltet“ aufnimmt, wenn sie sich in einem `Stopped` Status befindet. Weitere Informationen finden Sie unter Was ist eine nicht verwaltete Instance?.	25. Februar 2025

Thema aktualisiert: Überprüfen der Signatur des SSM Agent	Die AWS Systems Manager Agent (SSM Agent) RPM- und Debian-Installationspakete für Linux-Instances sind kryptografisch signiert. Sie können einen öffentlichen Schlüssel verwenden, um sicherzustellen, dass das Paket des Agenten original und unverändert ist. Wenn die Dateien beschädigt oder verändert werden, schlägt die Überprüfung fehl. Sie können die Signatur des Installer-Pakets entweder mit RPM oder GPG überprüfen. Das Thema beinhaltet einen neuen öffentlichen Schlüssel für SSM Agent. Weitere Informationen finden Sie unter Überprüfen der Signatur von SSM Agent.	14. Februar 2025

Neues Thema: Überprüfen der Signatur des Session Manager-Plugins	Die RPM- und Debian-Installationspakete des Session Manager-Plugins für Linux-Instances sind kryptografisch signiert. Sie können einen öffentlichen Schlüssel verwenden, um zu verifizieren, dass die Binärdatei und das Paket des Plugins original und unverändert sind. Wenn die Dateien beschädigt sind oder verändert wurden, schlägt die Überprüfung fehl. Sie können die Signatur des Installationspakets mithilfe des GNU Privacy Guard (GPG)-Tools überprüfen. Weitere Informationen finden Sie unter Überprüfen der Signatur des Session Manager-Plugins.	13. Februar 2025
Session Manager-Plugin aktualisiert	Das Session Manager-Plugin wurde kürzlich mit den folgenden Verbesserungen aktualisiert: Die Go-Version wurde im Dockerfile auf 1.23 aktualisiert. Der Schritt zur Versionskonfiguration in der README-Datei wurde aktualisiert. Weitere Informationen zur Verwendung des Session Manager-Plugins finden Sie unter Installieren des Session Manager-Plugins für die AWS CLI.	6. Februar 2025

TPS-Kontingente für Automation-API-Aktionen veröffentlicht	Wir haben die TPS-Kontingente (Transaction per Second), die früher als limitiert bezeichnet wurden, für mehrere API-Aktionen im Zusammenhang mit Systems Manager Automation veröffentlicht. Informationen finden Sie unter Systems Manager Service Quotas im Allgemeine Amazon Web Services-Referenz.	21. Januar 2025
Änderung der Terminologie: Die Funktionen von Systems Manager wurden in „Tools“ umbenannt	Bisher wurden die Funktionseinheiten, aus denen Systems Manager besteht, als Funktionen bezeichnet, z. B. als „Automation-Funktion“, als Parameter Store-Funktion usw. Diese Funktionseinheiten werden jetzt als Tools bezeichnet, wie z. B. das Automation-Tool und das Parameter Store-Tool. Das Benutzerhandbuch wurde aktualisiert, um diese Änderung widerzuspiegeln.	10. Januar 2025

Patch ManagerUnterstützung für zusätzliche macOS Versionen (Amazon EC2 AMIs)	Patch Manager unterstützt jetzt die macOS-Versionen 12.7, 13.6 und 13.7 sowie 15.x. Vollständige Listen OSs und Versionen, die von unterstützt werdenPatch Manager, finden Sie unter Unterstützte Betriebssysteme für Patch Manager.	8. Januar 2025
SSM Agent- und Patch Manager-Unterstützung für zusätzliche Versionen : CentOS Stream, Ubuntu Server und Windows Server	Patch Manager unterstützt jetzt CentOS Stream 9, Ubuntu Server 24.10 und Windows Server 2025. SSM Agent unterstützt jetzt Ubuntu Server 24.10 und Windows Server 2025. (Agentenunterstützung für CentOS Stream 9 wurde bereits veröffentlicht.) Vollständige Listen der unterstützten Versionen OSs und Versionen finden Sie in den folgenden Themen: • Unterstützte Betriebssysteme für Systems Manager • Unterstützte Betriebssysteme für Patch Manager	22. November 2024

Neues Thema: AWS KMS Verschlüsselung für Parameter Store SecureString Parameter	Im Parameter Store folgenden Thema erfahren Sie AWS Key Management Service , wie AWS Systems ManagerPa rameter Store Sie die Werte von `SecureString` Parametern verschlüsseln können: • AWS KMS Verschlüsselung für SecureString Parameter in Parameter Store	22. November 2024
Neue und aktualisierte verwaltete Richtlinien für Systems Manager	Um neue Features für Systems Manager zu unterstüt zen, veröffentlichen wir mehrere neue verwaltete Richtlinien zur Unterstützung neuer Systems-Manager-Ko nfigurationen und -Vorgänge und aktualisieren andere verwaltete Richtlinien. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	21. November 2024

Eine neue, vereinfachte Knoten-Management-Erfahrung für Systems Manager	AWS Systems Manager hat eine neue einheitliche Konsolenoberfläche für die Verwaltung von Knoten in großem Umfang für Konten und Regionen veröffentlicht. Sie können jetzt alle verwaltet en und nicht verwalteten Knoten in Ihren Organisat ionen AWS-Konten und Regionen von einem einzigen Ort aus einsehen. Sie können auch nicht verwaltete Knoten identifizieren, diagnostizieren und korrigieren. Systems Manager ist jetzt auch in Amazon Q Developer (Amazon Q) integriert, wodurch Sie Ihre Nodes von überall aus sehen und steuern können, AWS-Manag ementkonsole indem Sie Eingabeaufforderungen in natürlicher Sprache eingeben. Mit dieser Version können Sie nun auch einem delegierten Administratorkonto ermöglich en, Knoten im gesamten Unternehmen von einem zentralen Standpunkt aus zu verwalten. AWS Organizat ions Weitere Informationen finden Sie unter den folgenden Themen:	21. November 2024

	• Was ist? AWS Systems Manager • Einrichtung AWS Systems Manager • Erledigen Sie Knotenaufgaben mit AWS Systems Manager	
Session Manager-Behebung eines Plugin-Fehlers	Das Session Manager Plugin wurde kürzlich mit dem folgenden Bugfix aktualisiert: Änderung rückgängig gemacht, durch die Anmeldeinformationen zu OpenDataChannel Anfragen hinzugefügt wurden.	20. November 2024
Erweiterungen des Session Manager-Plugins	Diese Version ist am 20.11.2024 veraltet. Das Session Manager-Plugin wurde kürzlich mit den folgenden Verbesserungen aktualisiert. • Anmeldeinformationen zu OpenDataChannel Anfragen hinzugefügt. • Die von `testify` und `objx` abhängigen Pakete wurden aktualisiert.	6. November 2024

Zusätzliche Unterstützung für Betriebssystemversionen für macOS	Systems Manager unterstützt jetzt Version 15. x (Sequoia) des macOS Betriebssystems (nur EC2 Instanzen). Eine Liste aller unterstützten OSs Versionen finden Sie unter Unterstützte Betriebssysteme für Systems Manager.	6. November 2024
:Patch Manager: Aktualisierungen der unterstützten Paketnamenformate für Listen mit genehmigten und abgelehnten Paketen	Für mehrere Betriebssysteme haben wir die Formatlisten für Paketnamen aktualisiert und erweitert, die Sie in Ihren Patch-Baselines in den Listen Zulässige Patches und Abgelehnte Patches angeben können. Weitere Informationen finden Sie unter Paketnamensformate für Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022, Amazon Linux 2023, CentOS, Oracle Linux und Red Hat Enterprise Linux (RHEL).	1. November 2024

Unterstützung zusätzlicher Betriebsversionen für Patch Manager	Patch Manager unterstützt jetzt zusätzliche Versionen von Oracle Linux (8.10 und 9.4) und Ubuntu Server (23.10 und 24.04). Listen aller unterstützten Betriebssysteme und Versionen finden Sie in den folgenden Themen: • Unterstützte Betriebssysteme für Systems Manager • Unterstützte Betriebssysteme für Patch Manager	1. November 2024
SSM Agent-Unterstützung für zusätzliche Versionen: CentOS Stream, Oracle Linux und Ubuntu Server	SSM Agent unterstützt jetzt zusätzlich zu früheren unterstützten Versionen CentOS Stream 9, Oracle Linux 8.10 und 9.4 sowie Ubuntu Server 24.04 LTS. Eine vollständige Liste der unterstützten Betriebssysteme OSs und Versionen für Systems Manager finden Sie unter Unterstützte Betriebssysteme für Systems Manager.	25. Oktober 2024
Erweiterung des Session Manager-Plugins	Unterstützung für die Weitergabe der Plugin-Version mit OpenDataChannel Anfragen hinzugefügt.	10. Oktober 2024

Neu: Zeigt Details zu RDP-Verbindungen an, die hergestellt wurden mit Fleet Manager	Sie können jetzt Informationen über Remote Desktop Protocol-Verbindungen anzeigen, die von Benutzern in Ihrem Computer hergestellt wurden AWS-Konto. Weitere Informationen finden Sie unter Informationen zu aktuellen und abgeschlossenen Verbindungen anzeigen.	10. Oktober 2024
Patch Manager unterstützt jetzt SLES Version 15.6	Patching-Unterstützung für SUSE Linux Enterprise Server (SLES) 15.6 wurde veröffentlicht. Sie können jetzt SLES 15.6-Maschinen patchen mit Patch Manager. Eine vollständige Liste der von Patch Manager unterstützten Betriebssysteme und Versionen finden Sie unter Unterstützte Betriebssysteme für Patch Manager.	29. September 2024

Neue Versionen der AWS Lambda-Erweiterung Parameters and Secrets	Neue Versionen der AWS -Parameter und -Secrets-Lambda-Erweiterung sind jetzt verfügbar. Support für alle Architekturen wurde für die Region Asien-Pazifik (Malaysia) (ap-southeast-5) eingeführt. Darüber hinaus wurde Unterstützung für ARM64- und Mac with Apple silicon-Architekturerweiterungen für die folgenden Regionen hinzugefügt: • Asien-Pazifik (Hyderabad) (ap-south-2) • Asien-Pazifik (Melbourne) (ap-southeast-4) • Kanada West (Calgary) (ca-west-1) • Europa (Zürich) (eu-central-2) • Europa (Spanien) (eu-south-2) • Naher Osten (VAE) (me-central-) • China (Peking) (cn-north-1) • China (Ningxia) (cn-northwest-1) • Israel (Tel Aviv) (il-central-1) • AWS GovCloud (US-Ost) (us-gov-east-1) • AWS GovCloud (US-West) (us-gov-west-1)	19. September 2024

Neues Thema: Konfiguration von Berechtigungen für Wartungsfenster mit AWS CLI	Das Thema Konfigurieren von Berechtigungen für Wartungsfenster mithilfe von AWS CLI enthält Anweisungen zum Erstellen einer benutzerdefinierten Servicerolle (und der zugehörigen Richtlinien) zur Ausführung von Wartungsfensteraufgaben im Namen eines Benutzers.	19. August 2024
SSM Agentund Patch Manager Unterstützung für zusätzliche Versionen: AlmaLinuxOracle Linux, und Rocky Linux	SSM Agentund unterstützt Patch Manager jetzt zusätzlich zu früheren unterstützten Versionen die Versionen 8.10Rocky Linux, 9.3 und 9.4 von AlmaLinux und sowie Version 9.3 vonOracle Linux. Eine vollständige Liste der unterstützten Versionen OSs und Versionen finden Sie in den folgenden Themen: • Unterstützte Betriebssysteme für Systems Manager • Unterstützte Betriebssysteme für Patch Manager	14. August 2024

Neue IAM-Richtlinienbedingung für den Parameter Store-Support: ssm:Policies	Mithilfe von `ssm:Policies` , einer neu unterstützten Bedingung für IAM-Richtlinien, können Sie verhindern, dass Entitäten Parameter erstellen oder aktualisieren, die eine Parameterrichtlinie enthalten. Weitere Informationen finden Sie unter den folgenden Themen: • ssm:Policies: Die Erstellung oder Aktualisierung von Parametern verhindern, die Parameterrichtlinien verwenden • Bedingungsschlüssel für AWS Systems Manager in der Service Authorization Reference.	14. August 2024
Aktualisierte verwaltete Richtlinie fürQuick Setup: SSMQuick SetupRolePolicy	Systems Manager hat die verwaltete Richtlinie aktualisiertSSMQuickSetupRolePolicy , um Zugriff auf zusätzliche CloudFormation Stack-Sets zu ermöglichen. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	13. August 2024

Support für die Bereitstellung und Verwaltung von Systems-Manager-Ressourcen mithilfe von Terraform	Wir haben HashiCorp Terraform zur Liste der unterstützten Integrationen von Drittanbietern mit Systems Manager hinzugefügt. Terraform ist ein Open-Source-Softwaretool für Infrastructure as Code (IaC), das einen Befehlszeilenschnittstellen (CLI)-Workflow zur Verwaltung verschiedener Cloud-Services bereitstellt. Sie können Terraform verwenden, um eine Reihe häufig verwendeter Systems-Manager-Ressourcen und Datenquellen bereitzustellen und zu verwalten. Informationen zu dieser und anderen Integrationen von Drittanbietern mit Systems Manager finden Sie unter Integration mit anderen Produkten und Services.	1. August 2024

Neue Quick Setup-Konsolenerfahrung und API	Systems Manager Quick Setup hat ein neues Konsolenerlebnis und eine neue API veröffentlicht. Jetzt können Sie mit dieser API über die Konsole, AWS CLI, CloudFormation und interagieren. SDKs Sie können sich über die Quick Setup-Konsole für die neue Konsole anmelden. Weitere Informationen zur Einführung in das neue Quick Setup-Erlebnis finden Sie unter Erste Schritte mit Quick Setup. Weitere Informationen zu den API-Vorgängen, die über die Quick Setup-API verfügbar sind, finden Sie in der Quick Setup-API-Referenz.	1. August 2024

Neues Thema: Optionen für abgelehnte Patch-Listen in benutzerdefinierten Patch-Baselines	Für Patch-Vorgänge, die eine benutzerdefinierte Patch-Baseline in Patch Manager verwenden, haben wir das Verhalten klarer definiert, wenn einem zur Liste Abgelehnte Patches hinzugefügten Patch die Aktion `Allow as dependency` zugewiesen wird. Da Windows Server das Konzept der Patch-Abhängigkeiten nicht unterstützt, werden Patches, die noch nicht auf einem verwalteten Knoten installiert sind, übersprungen. Patches, die bereits auf dem Knoten installiert sind, erhalten den Status `INSTALLED_REJECTED`. Weitere Informationen finden Sie unter Listenoptionen für abgelehnte Patches in benutzerdefinierten Patch-Baselines und unter Werte zur Patch-Konformität für andere Betriebssysteme.	23. Juli 2024

Neues Thema: Konfigurieren von SSM Agent für die Verwendung mit dem Federal Information Processing Standard (FIPS)	Wir haben Anweisungen für die Konfiguration SSM Agent für die Verwendung mit dem Federal Information Processing Standard (FIPS) bereitgestellt. Weitere Informationen finden Sie unter Konfiguration von SSM Agent für die Verwendung mit dem Federal Information Processing Standard (FIPS).	22. Juli 2024
Update: Die Unterstützung für @-Symbole in Fleet Manager-Benutzernamen wurde klargestellt	Wenn ein IAM-Identity-Center-Benutzername ein oder mehrere @-Symbole enthält, ignoriert Fleet Manager RDP das erste @-Symbol und alle darauf folgenden Zeichen, unabhängig davon, ob das @ den Domainteil einer E-Mail-Adresse einleitet oder nicht. Weitere Informationen zu den unterstützten Zeichen für Benutzernamen in Fleet Manager-RDP-Verbindungen finden Sie unter Authentifizieren von Remotedesktopverbindungen.	21. Juli 2024

Aktualisierte verwaltete Richtlinie: Amazon SSMManaged EC2 InstanceDefaultPolicy	Systems Manager hat die verwaltete Richtlinie `AmazonSSMManagedEC2InstanceDefaultPolicy` aktualisiert und eine Inline-Erklärung IDs (SIDs) bereitgestellt, um den Zweck der einzelnen Richtlinienanweisungen zu verdeutlichen. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	18. Juli 2024

Namensänderungen an AWS verwalteten Buckets für Patch Manager Patch-Operationen	AWS besitzt und verwaltet eine Reihe von Amazon S3 S3-Buckets, auf die im Zuge der Durchführung verschied ener Patch Manager Patch-Operationen SSM Agent zugegriffen wird. Diese S3-Buckets sind öffentlich zugänglich und SSM Agent verbindet sich standardm äßig über HTTP-Aufrufe mit ihnen. Wenn Sie jedoch einen Virtual Private Cloud (VPC) -Endpunkt in Ihren Systems Manager-Vorgängen verwenden, müssen Sie in einem EC2 Amazon-Instanzprofil für Systems Manager oder in einer Servicerolle für EC2 Nicht-Mas chinen in einer Hybrid- und Multi-Cloud-Umgebung eine ausdrückliche Genehmigung erteilen. Andernfalls haben Ihre Ressourcen keinen Zugriff auf diese öffentlichen Buckets. In den meisten Fällen ändern wir die Namen dieser Buckets. Bei Patch-Vorgängen `aws-patchmanager-macos-us-east-2` wird beispiels weise der Bucket durch `aws-patchmanager-macos-us-east-2-552881074` ersetzt und der Bucket `aws-ssm-us-east-2` wird durch	18. Juli 2024

	`aws-patch-manager-us-east-2-552881074` ersetzt. Weitere Informationen finden Sie unter den folgenden Themen: • SSM AgentKommunikation mit AWS verwalteten S3-Buckets • Referenz: Amazon-S3-Buckets für Patch-Vorgänge	
Neue Serviceverknüpfte Rolle für Quick Setup	Systems Manager hat eine neue serviceverknüpfte Rolle, `AWSServiceRoleForSSMQuickSetup` , veröffentlicht. Systems Manager verwendet diese Rolle, um die Konfigurationsintegrität von mit Quick Setup eingerichteten Ressourcen zu prüfen, eine konsistente Verwendung von Parametern und bereitgestellten Ressourcen sicherzustellen und Ressourcen zu korrigieren, wenn Abweichungen erkannt werden. Die verwaltete Richtlinie, die dieser Rolle zugeordnet werden soll, `SSMQuickSetupRolePolicy` lautet. Weitere Informationen finden Sie unter AWSServiceRoleForSSMQuickEinrichten von dienstbezogenen Rollenberechtigungen für Systems Manager.	3. Juli 2024

Neue verwaltete Richtlinien für Quick Setup-Konfigurationstypen	Systems Manager hat weitere 12 neue verwaltete Richtlinien zur Unterstützung verschiedener Quick Setup-Konfigurationstypen und Prozesse veröffentlicht. Weitere Informationen finden Sie unter Systems Manager Manager-Updates für AWS verwaltete Richtlinien.	3. Juli 2024
Support für RHEL 8.10 und 9.4	Systems Manager und Patch Manager unterstützen jetzt die Red Hat Enterprise Linux-Versionen 8.10 und 9.4. Weitere Informationen finden Sie unter Unterstützte Betriebssysteme und Regionen und unter Unterstützte Betriebssysteme für Patch Manager.	26. Juni 2024
Patch ManagerUnterstützung für die Versionen 8.8 und 8.9: AlmaLinuxOracle Linux, und Rocky Linux	Patch Managerunterstützt jetzt zusätzlich zu früheren 8.x-Versionen die Versionen 8.8 und 8.9 von AlmaLinuxRocky Linux, und. Oracle Linux Eine vollständige Liste der unterstützten Betriebssysteme OSs und der Versionen für Patch Manager finden Sie unter Unterstützte Betriebssysteme für Patch Manager.	17. Juni 2024

Neue öffentliche Parameter für macOS Amazon EC2 AMIs	Öffentliche Parameter wurden Amazon Machine Images zur Unterstützung von macOS Instances von Amazon Elastic Compute Cloud veröffent licht. Weitere Informationen finden Sie unter den folgenden Themen. • Auffinden von öffentlichen Parametern • Aufrufen öffentlicher AMI-Parameter für macOS • Starten Sie eine Mac-Insta nce im EC2 Amazon-Be nutzerhandbuch • Amazon EC2 Mac-Insta nces im EC2 Amazon-Be nutzerhandbuch • Verwenden Sie AWS Systems Manager Parameter statt AMI IDs in Startvorlagen im Amazon EC2 Auto Scaling Scaling-B enutzerhandbuch	17. Juni 2024

Update: Regionale Verfügbarkeit des `/aws/service/global-infrastructure` -Parameterpfads	Wir haben geklärt, aus welchen kommerziellen Regionen der `/aws/service/global-infrastructure` öffentliche Parameterpfad abgefragt werden kann und wie eine Abfrage für den Pfad ausgeführt wird, wenn Sie in einem anderen kommerziellen AWS-Region arbeiten. Weitere Informationen finden Sie unter Aufrufen öffentlicher Parameter für AWS Dienste, Regionen, Endpunkte, Availability Zones, Local Zones und Wavelength Zones.	12. Juni 2024
Neu: Kapitel mit Codebeispielen	Ein neues Kapitel, Codebeispiele für die Verwendung von Systems Manager AWS SDKs, enthält Beispiele in verschiedenen SDK-Sprachen für die Arbeit mit dem Systems Manager Manager-Dienst.	8. Mai 2024

Änderungen an der `ec2messages:*` -Endpunktunterstützung	Für den AWS-Regionen Start im Jahr 2024 oder später werden die `ec2messages:*` Endpunkte nicht unterstützt, um SSM Agent Status- und Ausführungsinformationen zurück an den Systems Manager Manager-Dienst zu senden. Konten in diesen Regionen müssen `ssmmessages:*` verwenden. In Regionen, die vor 2024 gestartet wurden, werden `ssmmessages:*` und `ec2messages:*` weiterhin unterstützt. Wir empfehlen jedoch, jetzt nur den `ssmmessages:*` -Endpunkt (Amazon Message Gateway Service) zu verwenden. Sie können derzeit problemlos `ec2messages:*` -Berechtigungen aus Ihren Richtlinien entfernen. Weitere Informationen finden Sie unter Arbeiten mit SSM Agent und agentenbezogene API-Vorgänge (ssmmessages- und ec2messages-Endpunkte).	3. Mai 2024

Zusätzliche Laufzeiten sind für die Ausführung von Skripten in Automation-Runbooks verfügbar	Die `aws:executeScript` -Aktion unterstützt jetzt die Python-Laufzeiten 3.9, 3.10 und 3.11. Weitere Informationen zur Verwendung dieser Aktion finden Sie unter `aws:executeScript` .	23. April 2024
Support für die Versionen 8.8 und 8.9: AlmaLinuxOracle Linux, und Rocky Linux	Systems Manager unterstützt jetzt zusätzlich zu früheren 8.x-Versionen die Versionen 8.8 und 8.9 von AlmaLinux Rocky Linux, und. Oracle Linux Eine vollständige Liste der unterstützten Betriebssysteme OSs und Versionen finden Sie unter Unterstützte Betriebssysteme für Systems Manager.	22. April 2024

Patch Manager: Zum Patch-Status 'INSTALLED_PENDING_REBOOT' wechseln	Bisher konnten nur Patches, die von Patch Manager installiert wurden, als `INSTALLED_PENDING_REBOOT` markiert werden. Patches, die außerhalb von Patch Manager installiert werden, erhalten niemals den Status. `INSTALLED_PENDING_REBOOT` kann jetzt auf jeden Patch angewendet werden, der seit dem letzten Neustart auf einen verwalteten Knoten angewendet wurde. Dazu gehören Patches, die Patch Manager mit der ausgewählten `NoReboot`-Option installiert wurden, und Patches, die außerhalb oder Patch Manager nach dem letzten Neustart des Knotens installiert wurden. Eine Beschreibung aller Werte für den Patch Manager-Patchstatus finden Sie unter Grundlegendes zu den Werten für den Status der Patch-Konformität.	16. April 2024
Support für RHEL 8.9 und 9.3	Systems Manager einschließlich Patch Manager unterstützt jetzt die Versionen 8.8 und 9.3 von Red Hat Enterprise Linux (RHEL) zusätzlich zu früheren 8.x-und 9.x-Versionen.	26. März 2024

Themen-Update: AWS verwaltete Richtlinien für AWS Systems Manager	Das Thema Von AWS verwaltete Richtlinien für AWS Systems Manager enthält Informationen zu den vier verwalteten Richtlinien für Systems Manager, die seit dem 12. März 2021 eingeführt oder aktualisiert wurden. Wir haben diesem Thema einen Abschnitt mit Informationen zu den 12 anderen verwalteten Richtlinien zur Verwendung mit Systems Manager hinzugefügt, die vor diesem Datum erstellt oder zuletzt aktualisiert wurden. Weitere Informationen finden Sie unter Zusätzliche verwaltete Richtlinien für Systems Manager.	1. März 2024

Parameter Store-Unterstützung der kontoübergreifenden Freigabe	Sie können jetzt erweiterte Parameter sicher und effizient innerhalb Ihrer Organisation AWS-Konten oder innerhalb Ihrer AWS Organisation teilen, indem Sie die gemeinsame Nutzung von Ressourcen einrichten. Die gemeinsame Nutzung von Ressourcen ermöglicht es Ihnen, das Anwendungskonfigurationsmanagement zu zentralisieren und den betrieblichen Aufwand zu reduzieren, der durch die gemeinsame Nutzung der Parameter mit jedem einzelnen Konto, das Sie besitzen, entsteht. Parameter können über die Parameter Store Konsole, die Konsole oder die AWS RAM von Konten gemeinsam genutzt AWS CLI werden. Weitere Informationen finden Sie unter Arbeiten mit geteilten Parametern.	21. Februar 2024

Verbesserung der Automation-Aktionen	Sie können jetzt die Eigenschaften `onFailure` und `isCritical` mit der `aws:approve` -Aktion verwenden. Weitere Informationen zur `aws:approve` -Aktion finden Sie unter aws:approve – Eine Automatisierung für die manuelle Genehmigung pausieren.	12. Februar 2024
Unterstützung zusätzlicher Betriebsversionen für Patch Manager	Wir haben die Liste der unterstützten Betriebssystemversionen für Patch Manager erweitert. Außerdem wurde Unterstützung für Folgendes hinzugefügt: • Debian Server 11.x und 12.x • macOS 14.x (Sonoma) • SUSE Linux Enterprise Server (SLES) 15.5 • Ubuntu Server23.04	04. Januar 2024
Automatische SSM Agent-Updates mithilfe der Application Manager-Konsole konfigurieren	Sie können jetzt die Application Manager-Konsole verwenden, um SSM Agent-Updates für Ihre Anwendungs-Instances zu automatisieren. Weitere Informationen finden Sie unter Arbeiten mit Ihren Anwendungs-Instances.	21. Dezember 2023

Aktualisierter Prozess für die Registrierung von EC2 Maschinen, die nicht von Amazon stammen, in Hybrid- und Multi-Cloud-Umgebungen	Systems Manager unterstützt Sie jetzt bei der Registrierung von Maschinen `ssm-setup-cli` , die nicht von Amazon Elastic Compute Cloud (Amazon EC2) stammen, in Hybrid- und Multi-Clo ud-Umgebungen. Weitere Informationen finden Sie unter So installieren Sie den SSM Agent auf Hybrid-Linux-Knote n und So installieren Sie den SSM Agent auf Hybrid-Wi ndows-Knoten.	20. Dezember 2023
Verwalten von Amazon-EBS-Volumes mit Fleet Manager	Sie können jetztFlee t Manager, ein Tool in, verwenden AWS Systems Manager, um Amazon Elastic Block Store-Volumes auf Ihren verwalteten Instances zu verwalten. Sie können beispielsweise ein EBS-Volume initialisieren, eine Partition formatieren und das Volume mounten, um es für die Nutzung verfügbar zu machen. Weitere Informati onen finden Sie unter EBS-Volumeverwaltung.	14. Dezember 2023

Erweiterung des Session Manager-Plugins	Unterstützung für die Übergabe einer StartSession API-Antwort als Umgebungsvariable an hinzugefügt session-manager-plugin.	4. Dezember 2023
Neue visuelle Designerfahrung für Automation-Runbooks	Sie können Runbooks jetzt mithilfe einer visuellen Designerfahrung erstellen und bearbeiten, die von Systems Manager Automation entwickelt wurde. Das visuelle Designerlebnis bietet eine drag-and-drop Low-Code-Oberfläche, sodass Sie Runbooks einfacher erstellen und bearbeiten können. Weitere Informationen finden Sie unter Visuelle Designerfahrung für Automation-Runbooks.	26. November 2023

Neue Systems-Manager-Automation-Aktionen, Datenelement- und Funktionserweiterungen für Runbooks	Mit der `aws:loop`-Aktion können Sie jetzt mehrere Aktionen in einem Runbook wiederholen. Diese neue Aktion unterstützt Loops im Stil `do while` und `for each`. Darüber hinaus können Sie mithilfe des neuen Variablen-Datenelements Werte dynamisch im Kontext eines Runbooks definieren, referenzieren und aktualisieren. Verwenden Sie die neue `aws:updateVariable` -Aktion, um den Wert einer Variablen in Ihrem Runbook zu aktualisieren. Automation hat auch Unterstützung für dynamische Datentypkonvertierungen für Ausgaben hinzugefügt. Das heißt, wenn der Wert einer Ausgabe nicht dem von Ihnen angegebenen Datentyp entspricht, versucht Automation, den Datentyp zu konvertieren. Wenn der zurückgegebene Wert beispielsweise ein `Integer` ist, der angegebene `Type` jedoch ein `String` ist, ist der endgültige Ausgabewert ein `String`-Wert. Schließlich unterstützt Automation jetzt JSONPath Filterausdrücke für Selektoren. Weitere Informati	17. November 2023

	onen finden Sie unter den folgenden Themen: • aws:loop – Über Schritte in einer Automatisierung interieren • aws:UpdateVariable – Aktualisiert einen Wert für eine Runbook-Variable • Datenelemente und Parameter – Datenelemente der obersten Ebene • Verwenden von Aktionsausgaben als Eingaben. • Verwendung JSONPath in Runbooks.	
Die Regionsunterstützung für Remote Desktop Protocol (RDP)-Verbindungen wurde aktualisiert	Fleet Manager Remote Desktop, das von Amazon DCV unterstützt wird, bietet Ihnen eine sichere Verbindung zu Ihren Windows Server-Instances direkt von der Systems-Manager-Konsole aus. Die folgenden drei zusätzlichen Regionen wurden für Fleet Manager-Remotedesktopverbindungen aktiviert: • Afrika (Kapstadt) (af-south-1) • Asien-Pazifik (Jakarta) (ap-southeast-3) • Israel (Tel Aviv) (il-central-1)	15. November 2023

Patch Manager: Erweiterte Unterstützung von Betriebssystemversionen für RHEL und macOS	Patch Manager unterstützt jetzt die folgenden zusätzlichen Betriebssystemversionen: • Red Hat Enterprise Linux: Version 8.8 • macOS: 11.5–11.7 (Big Sur) • macOS: 12.0–12.6 (Monterey) • macOS: 13.0–13.5 (Ventura)	23. Oktober 2023
Neue OpsCenter-API – OpsItem löschen	OpsCenter bietet jetzt die API „OpsItem löschen“ zum Löschen einzelner OpsItems. Weitere Informationen finden Sie unter DeleteOpsItem in der AWS Systems Manager -API-Referenz.	20. Oktober 2023
Neuer Quick Setup-Konfigurationstyp: SSM Agent-Updates für die gesamte Organisation	Der neue Konfigurationstyp Standard-Host-Management-Konfiguration ermöglicht es einem Organisationsadministrator, wie unter definiert AWS Organizations, automatische Prüfungen und SSM Agent Aktualisierungen aller EC2 Instanzen in den Konten und Regionen der Organisation zu veranlassen. Weitere Informationen finden Sie unter Standard-Host-Verwaltung für eine Organisation.	16. Oktober 2023

Neues Titel- und Beschreibungsformat für „OpsItems Erstellt von CloudWatch Application Insights“	Der Titel und die Beschreibung von OpsItems Created by CloudWatch Application Insights werden am 16. Oktober 2023 in ein verbessertes Format geändert. Das neue Format finden Sie unter Amazon CloudWatch Application Insights.	29. September 2023

Support für mehrere Bildschirmauflösungen in Fleet Manager-RDP-Verbindungen	Wenn Sie sich über die Option Remote-Desktop-Protokoll (RDP) in Fleet Manager mit Windows Server verwalteten Knoten verbinden, können Sie jetzt die Bildschirmauflösung wählen. Bisher wurde für alle Verbindungen eine feste Auflösung von 720P (1366 x 768) verwendet. Sie können jetzt für jede Verbindung aus den folgenden Optionen wählen: • Automatisch anpassen (bestimmt anhand der erkannten Bildschirmgröße die optimale Auflösung) • 1 920 x 1 080 • 1 400 x 900 • 1 366 x 768 • 800 x 600 Weitere Informationen finden Sie unter Über Remote Desktop mit einem verwalteten Knoten verbinden.	22. September 2023

Neues Thema: Zufällige Patch-Baseline IDs bei Patch-Policy-Vorgängen	Wir haben Inhalte hinzugefügt, die beschreiben, wie Quick Setup Patch-Richtlinien den `BaselineOverride` Parameter im `AWS-RunPatchBaseline` SSM-Befehlsdokument verwenden, um bei jeder Ausführung eines Patch-Policy-Vorgangs zufällige IDs Patch-Baselines zu generieren. Weitere Informationen finden Sie unter Zufällige Patch-Baseline IDs bei Vorgängen mit Patch-Richtlinien.	22. September 2023

Ein neuer betrieblicher Einblick in die Verwaltung von OpsItems	OpsCenterenthält jetzt einen Einblick in betriebliche Abläufe mit der Bezeichnung Ressourcen, die am meisten generieren OpsItems. Ein solcher Einblick wird generiert, wenn für eine AWS Ressource mehr als 10 geöffnet sindOpsItems. Verwenden Sie diesen Einblick, um problematische Ressourcen zu lokalisieren. Verwenden Sie das `AWS-BulkResolveOpsItems` -Runbook aus einem Einblick heraus, um OpsItems im Zusammenhang mit einer Ressource schnell zu lösen. Weitere Informationen finden Sie unter Analysieren betrieblicher Einblicke zur Reduzierung von OpsItems.	22. September 2023
Öffentlicher GPG-Schlüssel aktualisiert	Ein neuer öffentlicher Schlüssel wurde erstellt, um die Signatur von SSM Agent zu verifizieren. Weitere Informationen finden Sie unter Signatur von SSM Agent verifizieren.	5. September 2023

Support für zusätzliche Versionen von AlmaLinux , Oracle LinuxRHEL, und hinzugefügt Rocky Linux	Die Listen der unterstützten Betriebssysteme für AWS Systems Manager und Patch Manager wurden aktualisiert, um die Unterstützung der folgenden zusätzlichen Betriebssystemversionen widerzuspiegeln: • AlmaLinux: 9.2 • Oracle Linux: 8.7 und 9.2 • Red Hat Enterprise Linux(RHEL): 8.7, 9.1 und 9.2 • Rocky Linux: 8.6 und 8.7, 9.0–9.2	30. August 2023

OpsCenter unterstützt nun die Markdown-Formatierung im OpsItem-Beschreibungsfeld.	OpsCenter unterstützt jetzt die Markdown-Formatierung im OpsItem-Beschreibungsfeld. Die folgenden Typen der Markdown-Formatierung werden unterstützt: • Paragraphen • Zeilenabstand • Horizontale Linien • Überschriften • Textformatierung • Links • Listen Weitere Informationen finden Sie unter Verwenden von Markdown in der Konsole im Handbuch Erste Schritte mit dem Handbuch AWS-Managementkonsole Erste Schritte.	18. August 2023

Neue Versionen der AWS Lambda-Erweiterung Parameters and Secrets	Neue Versionen der AWS Parameters and Secrets Lambda Extension sind jetzt verfügbar. Darüber hinaus wurde Erweiterungsunterstützung für die Regionen Asien-Pazifik (Melbourne) (ap-southeast-4) und Israel (Tel Aviv) (il-central-1) hinzugefügt (nur für x86_64- und x86-Architekturen). Weitere Informationen finden Sie unter Verwenden von Parameter Store Parametern in AWS Lambda Funktionen.	16. August 2023

Update: Es wurden Informationen zu den erforderlichen Berechtigungen für Patch-Richtlinien-Buckets von Quick Setup hinzugefügt	Wenn Sie eine Patch-Richtlinie erstellen, erstellt Quick Setup einen Amazon-S3 Bucket, der eine Datei mit dem Namen `baseline_overrides.json` enthält. In dieser Datei werden Informationen zu den Patch-Baselines gespeichert, die Sie für Ihre Patch-Richtlinie angegeben haben. Bei der Konfiguration der Patch-Richtlinie haben Sie die Möglichkeit, das Kontrollkästchen Erforderliche IAM-Richtlinien zu vorhandenen Instance-Profilen hinzufügen, die an Ihre Instances angehängt sind, zu aktivieren. Wenn Sie diese Option nicht auswählen, müssen Sie bestimmten Ressourcen manuell Berechtigungen für den Zugriff auf diesen Bucket gewähren. Andernfalls schlagen Ihre Richtlinienoperationen möglicherweise fehl. Weitere Informationen finden Sie unter den folgenden Themen: • Berechtigungen für den S3-Bucket mit der Patch-Richtlinie • Problem: Fehler „InvokePatchBaselineOperation : Zugriff verweigert“ oder	6. Juli 2023

	Fehler „Datei kann nicht von S3 heruntergeladen werden“ für `baseline_overrides.json`	
Verwenden Sie Quick Setup, um OpsCenter für die Mehrkontenverwaltung von OpsItem zu konfigurieren	Quick Setup für OpsCenter hilft Ihnen dabei, die folgenden Aufgaben für die Verwaltung von OpsItems über mehrere Konten hinweg zu erledigen: • Angeben des delegierten Administratorkontos • Erforderliche AWS Identity and Access Management (IAM-) Richtlinien und Rollen erstellen • Angabe einer AWS Organizations Organisation oder einer Teilmenge von Mitgliedskonten, die ein delegierter Administrator kontenübergreifend verwalten kann OpsItems Weitere Informationen finden Sie unter (Optional) Konfigurieren von OpsCenter für die kontenübergreifende Verwaltung von OpsItems mithilfe vonQuick Setup.	19. Juni 2023

Aktualisieren Sie Amazon EC2 Launch Agents mit Quick Setup	Sie können jetzt zulassen, dass Systems Manager alle 30 Tage nach einer neuen Version des auf Ihrer Instance installierten Startagenten sucht. Wenn eine neue Version verfügbar ist, aktualisiert Systems Manager den Agenten auf Ihrer Instance. Weitere Informationen finden Sie unter Quick Setup-Host-Verwaltung.	19. Juni 2023
Patch Manager unterstützt jetzt Ubuntu Server 22.04 LTS	Sie können jetzt Patch Manager verwenden, um Ubuntu Server 22.04. LTS-Knoten zu patchen. Wie andere unterstützte Versionen von verwendet Ubuntu Server auch Version 22.04 LTS die AWS verwaltete `AWS-UbuntuDefaultPatchBaseline` Patch-Baseline.	15. Mai 2023

Systems Manager unterstützt jetzt AlmaLinux, einschließlich Patch Manager	Sie können jetzt Systems Manager verwenden, um Knoten der Versionen AlmaLinux 8.3-8.7; 9.0-9.1 zu verwalten. Viele der Regeln, die für RHEL 8 beim Patchen gelten, gelten auch für. AlmaLinux AlmaLinuxverwendet das neue`AWS-DefaultAlmaLinuxPatchBaseline` . Weitere Informationen finden Sie unter den folgenden Themen: • Manuell SSM Agent auf AlmaLinux Instanzen installieren • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • So funktionieren Patch-Basisregeln für AlmaLinux RHEL, und Rocky Linux.	8. Mai 2023
Stellen Sie den EC2 Launch v2-Agenten bereit mit Quick Setup	Sie können den EC2 Launch v2-Agenten jetzt mithilfe von bereitstellenQuick Setup. Weitere Informationen finden Sie unter Bereitstellen von Distributor-Paketen mit Quick Setup.	13. April 2023

Systems Manager unterstützt jetzt Amazon Linux 2023	Systems Manager unterstützt jetzt den neuen EC2 Instance-Typ Amazon Linux 2023 (AL2023), einschließlich der Unterstützung von Patch Manager Vorgängen. Viele der Regeln für Patches, die für Amazon Linux 2 gelten, gelten auch für Amazon Linux 2023. (Patch Manager unterstützt auch weiterhin die Vorabversion Amazon Linux 2022.) Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln in Amazon Linux 1, Amazon Linux 2, Amazon Linux 2022 und Amazon Linux 2023	23. März 2023

Inhalt zur Einrichtung von EC2 Amazon-Instances überarbeitet	Wir haben den Inhalt zur Einrichtung von EC2 Amazon-Instances überarbeitet. Es wird nun empfohlen, die neu veröffentlichte Standardkonfiguration für die Host-Verwaltung für Instance-Berechtigungen zu verwenden. Weitere Informationen finden Sie unter Erforderliche Instance-Berechtigungen für Systems Manager konfigurieren.	15. Februar 2023
Automatische Instance-Verwaltung mit der Standardkonfiguration für die Host-Verwaltung	Mit Systems Manager können Sie jetzt automatisch gesamte EC2 AWS-Region Amazon-Instances verwalten. Weitere Informationen finden Sie unter Standardkonfiguration für die Host-Verwaltung.	15. Februar 2023

Hinzufügen von SSM-Dokumenten zu Ihren Favoriten	Um Ihnen das Auffinden häufig genutzter SSM-Dokumente zu erleichtern, können Sie jetzt Dokumente zu Ihren Favoriten hinzufügen. Sie können bis zu 20 Dokumente pro Dokumenttyp, pro AWS-Konto und als Favorit markieren AWS-Region. Sie können Ihre Favoriten über die Dokumenten-Konsole von Systems Manager auswählen, ändern und anzeigen. Weitere Informationen finden Sie unter Hinzufügen von Dokumenten zu Ihren Favoriten.	07. Februar 2023
Implementieren von Änderungskontrollen für die Automatisierung mit Change Calendar	Durch die Integration von Automation mit Change Calendar können Sie jetzt Änderungskontrollen für alle Automatisierungen in Ihrem AWS-Konto implementieren. Weitere Informationen finden Sie unter Implementierung von Änderungskontrollen für die Automatisierung.	24. Januar 2023

Neuer Change Manager-Genehmigungs-Workflow	Der Change Manager-Genehmigungs-Workflow unterstützt jetzt Genehmigungen pro Ebene anstatt Genehmigungen pro Zeile. Bisher musste jeder Genehmiger, den Sie einer Genehmigungsebene hinzugefügt haben, eine Änderungsanfrage genehmigen. Andernfalls wurde das Level nicht genehmigt. Nun können Sie festlegen, wie viele Genehmigungen pro Ebene erforderlich sind, und können so viele oder mehr Genehmiger hinzufügen. Beispielsweise können Sie drei Genehmigungen für eine Ebene anfordern, aber bis zu fünf Genehmiger angeben. Die Genehmigungen von drei dieser Genehmiger sind ausreichend, um die Ebene zu genehmigen. Weitere Informationen finden Sie unter Über Genehmigungen in Ihren Änderungsvorlagen.	23. Januar 2023

Neu: Konfigurieren von Patching für eine gesamte Organisation mithilfe einer Patch-Richtlinie in Quick Setup	Mit Quick Setup, einem Tool von Systems Manager, können Sie jetzt Patch-Richtlinien erstellen, die von Patch Manager unterstützt werden. Eine Patch-Richtlinie definiert den Zeitplan und die Patch-Baseline, die beim automatischen Patching Ihrer verwalteten Knoten verwendet werden sollen. Mit einer einzelnen Patch-Richtlinienkonfiguration können Sie Patches für alle Konten in allen Regionen in Ihrer Organisation, nur für die von Ihnen ausgewählten Konten und Regionen oder für ein einzelnes Konto-Region-Paar definieren. Weitere Informationen finden Sie unter den folgenden Themen. • Verwenden von Quick Setup Patch-Richtlinien • Automatisieren von unternehmensweitem Patching mithilfe einer Quick Setup-Patch-Richtlinie	22. Dezember 2022

Application Managerintegriert sich in Amazon EC2 , um Informationen zu Ihren Instances im Kontext einer Anwendung anzuzeigen.	Application Managerzeigt den Instance-Status, den Status und den Zustand von Amazon EC2 Auto Scaling für eine ausgewählte Anwendung in einem grafischen Format an. Die Registerkarte Instances enthält auch eine Tabelle mit den folgenden Informationen für jede Instance in Ihrer Anwendung.
	• Instance-Status (Ausstehend, Angehalten, Wird ausgeführt, Beendet)
	• Ping-Status für SSM Agent
	• Status und Name des letzten Systems-Manager-Automation-Runbooks, das auf der Instance verarbeitet wurde
	• Eine Anzahl von Amazon CloudWatch Logs-Alarmen pro Bundesstaat.
	  • `ALARM` – Die Metrik oder der Ausdruck liegt außerhalb des festgelegten Schwellenwerts.
	  • `OK` – Die Metrik oder der Ausdruck liegt innerhalb des festgelegten Schwellenwerts.
	  • `INSUFFICIENT_DATA` – Der Alarm wurde soeben gestartet; die Metrik ist
		22. Dezember 2022

	nicht verfügbar oder es sind nicht genügend Daten verfügbar, damit die Metrik den Alarmstatus bestimmen kann. • Zustand der Auto-Scaling-Gruppe für die übergeordneten und einzelnen Auto-Scaling-Gruppen	
Planen Sie das Starten und Stoppen Ihrer EC2 Amazon-Instances mit Quick Setup	Sie können jetzt die Resource Scheduler-Lösung einsetzen, um das Starten und Stoppen Ihrer EC2 Amazon-Instances mithilfe Quick Setup von zu automatisieren. Weitere Informationen finden Sie unter Resource Scheduler.	19. Dezember 2022

OpsCenter unterstützt jetzt das kontenübergreifende Arbeiten mit OpsItems	OpsCenter unterstützt das Arbeiten mit OpsItems von einem Verwaltungskonto (entweder einem AWS Organizations -Verwaltungskonto oder einem von Systems Manager delegierten Administratorkonto) und einem Mitgliedskonto während einer Sitzung. Nach der Konfiguration können Benutzer die folgenden Arten von Aktionen ausführen: • Erstellen, Anzeigen und Aktualisieren von OpsItems in einem Mitgliedskonto • Sehen Sie sich detaillierte Informationen zu den AWS Ressourcen an, die OpsItems in einem Mitgliedskonto angegeben sind • Starten von Systems-Manager-Automation-Runbooks zur Behebung von Problemen mit AWS -Ressourcen in einem Mitgliedskonto Weitere Informationen finden Sie unter Einrichten von OpsCenter für das kontoübergreifende Arbeiten mit OpsItems.	16. November 2022

Verfolgen Sie die Details von Change Manager Änderungsanfragen mithilfe von AWS CloudTrail Lake	Sie können jetzt einen Ereignisdatenspeicher in AWS CloudTrail Lake verwenden, um Details zu den Änderungsanforderungen zu erfassen und zu überprüfen, die Change Manager für Ihre Organisation oder Ihr Konto eingereicht wurden. Zu diesen Informationen gehören überprüfbare Details über die Benutzeridentität, die die Änderungsanforderung erstellt hat, die IP-Adresse, von der aus die Anfrage gestellt wurde, den AWS-Regionen Ort, an dem die Änderungen vorgenommen wurden, die Zielressourcen und vieles mehr. Weitere Informationen finden Sie unter Überwachung der Ereignisse Ihrer Änderungsanfragen und Überprüfen von Details, Aufgaben und Zeitplänen für Änderungsanfragen.	11. November 2022

Zusätzliche Aufgabensteuerungen von Systems Manager Automation mithilfe von CloudWatch Alarmen	Sie können jetzt mithilfe von CloudWatch Alarmen zusätzliche Steuerung implementieren, wenn Sie Automatisierungen über mehrere Konten und Regionen hinweg ausführen. Indem Sie eine Metrik oder einen zusammengesetzten CloudWatch Alarm auf eine Automatisierung anwenden, können Sie anhand der von Ihnen definierten Metriken steuern, wann eine Automatisierung beendet wird. Weitere Informationen zum Anwenden eines CloudWatch Alarms auf eine Automatisierung, die über mehrere Konten und Regionen läuft, finden Sie unter Ausführen einer Automatisierung in mehreren Regionen und Konten (Konsole)	9. November 2022

Aktualisiert: Parameter Store 'Parameter in AWS Lambda Funktionen verwenden'	Wir haben zusätzliche Informationen bereitgestellt, die Ihnen helfen sollen, die Lambda-Erweiterung Parameters and Secrets zu verwenden, um Parameterwerte abzurufen und sie für die future Verwendung in Lambda-Funktionen AWS zwischenzuspeichern. Durch die Verwendung der Lambda-Erweiterung können Sie Ihre Kosten senken, indem Sie die Anzahl der API-Aufrufe auf Parameter Store reduzieren. Weitere Informationen finden Sie unter Verwenden von Parameter Store Parametern in AWS Lambda Funktionen.	25. Oktober 2022

Zusätzliche Systems Manager Manager-Aufgabenst euerungen mithilfe von CloudWatch Alarmen	Sie können jetzt mithilfe von CloudWatch Alarmen zusätzliche Steuerelemente bei der Ausführung von Automatisierungen und Befehlen implementieren. Ein CloudWatch Alarm kann auch zu einer Automatisierung oder einem Befehl hinzugefü gt werden, wenn er für eine State Manager Zuordnung s- oder Wartungsfensterauf gabe registriert ist. Indem Sie einen zusammengesetzten CloudWatch Alarm auf eine Automatisierung oder einen Befehl anwenden, können Sie anhand der von Ihnen definiert en Metrik steuern, wann eine Automatisierung oder ein Befehl beendet wird. Weitere Informationen zum Anwenden eines CloudWatch Alarms auf eine Automatisierung oder einen Befehl finden Sie in den folgenden Verfahren: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln in Amazon Linux 1, Amazon Linux 2 und Amazon Linux 2022.	26. September 2022

Zusätzliche Systems Manager Manager-Aufgabenst euerungen mithilfe von CloudWatch Alarmen	Sie können jetzt mithilfe von CloudWatch Alarmen zusätzliche Steuerelemente bei der Ausführung von Automatisierungen und Befehlen implementieren. Ein CloudWatch Alarm kann auch zu einer Automatisierung oder einem Befehl hinzugefü gt werden, wenn er für eine State Manager Zuordnung s- oder Wartungsfensterauf gabe registriert ist. Indem Sie einen zusammengesetzten CloudWatch Alarm auf eine Automatisierung oder einen Befehl anwenden, können Sie anhand der von Ihnen definiert en Metrik steuern, wann eine Automatisierung oder ein Befehl beendet wird. Weitere Informationen zum Anwenden eines CloudWatch Alarms auf eine Automatisierung oder einen Befehl finden Sie in den folgenden Verfahren: • Ausführen einer einfachen Automatisierung • Ausführen von Befehlen über die Konsole • Erstellen einer Zuordnung • Einem Wartungsfenster Aufgaben zuweisen	26. September 2022

Klärung der Anforderungen für Advanced-Instances-Kontingente	Basierend auf Kundenfeedback haben wir die Szenarien geklärt, in denen Sie die Advanced-Instances-Kontingente in Instance-Kontingente konfigurieren aktivieren müssen.	21. September 2022
Stellen Sie den CloudWatch Amazon-Agenten bereit mit Quick Setup	Sie können den CloudWatch Amazon-Agenten jetzt mithilfe von bereitstellenQuick Setup. Weitere Informationen finden Sie unter Bereitstellen von Distributor-Paketen mit Quick Setup.	20. September 2022

Der Schlüssel PatchGroup " wird jetzt für Patch-Gruppen unterstützt, wenn EC2 Instance-Metadaten zulässig sind	Wenn Sie Tags in EC2 Instanzmetadaten zulassen, dürfen die von Ihnen erstellten Tag-Schlüssel keine Leerzeichen enthalten. Bisher konnten Kunden dadurch einige ihrer EC2 Instances nicht zu Patchgruppen hinzufügen, Patch Manager da der Tag-Schlüssel auf die Instanzen angewendet werden `Patch Group` musste. Patch Managerunterstützt jetzt sowohl `Patch Group` (mit einem Leerzeichen) als auch `PatchGroup` (ohne Leerzeichen) als Tag-Schlüssel zur Identifizierung von Instanzen für eine Patchgruppe. EC2Instanzen, in denen Tags in Instanzmetadaten zulässig sind, können jetzt zu Patchgruppen in hinzugefügt werdenPatch Manager. Informationen zu Patch-Gruppen finden Sie unter Über Patch-Gruppen.	31. August 2022

Neues Thema: „So werden Veröffentlichungs- und Aktualisierungsdaten von Paketen berechnet“	In Patch-Baselines, die von verwaltet werden AWS, werden neue Patches 7 Tage nach ihrer Veröffentlichung oder Aktualisierung automatisch genehmigt. In benutzerdefinierten Patch-Baselines, die Sie erstellen, können Sie optional angeben, wie viele Tage nach ihrer Veröffentlichung oder Aktualisierung gewartet werden sollen, um ihre Installation automatisch zu genehmigen. Bei Amazon Linux 1 und Amazon Linux 2 beeinflussen verschiedene Faktoren, wie die neuesten Veröffentlichungs- und Aktualisierungsdaten berechnet werden. Um unerwartete Ergebnisse bei der Auswahl von Verzögerungen bei der automatischen Genehmigung zu vermeiden, werden diese Faktoren im Thema So werden Veröffentlichungs- und Aktualisierungsdaten von Paketen berechnet erläutert.	24. August 2022

Aktualisierter Inhalt: Patchen eines AMI und Aktualisieren einer Auto-Scaling-Gruppe	Wir haben die exemplarische Vorgehensweise zum Aktualisieren von AMIs für Auto-Scaling-Gruppen aktualisiert, um Startvorlagen anstelle von Startkonfigurationen zu verwenden. Darüber hinaus haben wir die neuesten Automatisierungsaktionen und Runtimes in den Runbook-Inhalten implementiert.	22. Juni 2022

Change Manager: Verhindern Sie, dass Benutzer automatisch genehmigbare Anfragen erstellen	Sie können Änderungsvorlagen in Change Manager so konfigurieren, dass automatische Genehmigungen unterstützt werden. Dies bedeutet, dass Benutzer mit den erforderlichen IAM-Berechtigungen die Änderungsanforderung starten können, ohne dass eine zusätzliche Genehmigung erforderlich ist. Sie können jetzt auch einzelne Benutzer, Gruppen oder IAM-Rollen daran hindern, automatisch genehmigbare Anforderungen zu übermitteln, selbst wenn sie von einer Änderungsvorlage unterstützt werden. Dies wird durch die Verwendung eines neuen IAM-Bedingungsschlüssels, `ssm:AutoApprove`, erreicht. Weitere Informationen finden Sie unter Controlling access to auto-approval runbook workflows (Steuern des Zugriffs auf Runbook-Workflows für automatische Genehmigung)	15. Juni 2022

Aktualisierte Anleitung für Wartungsfenster-Aufgaben	Zuvor bot Ihnen die Systems-Manager-Konsole die Möglichkeit, die von AWS verwaltete serviceverknüpfte IAM-Rolle `AWSServiceRoleForAmazonSSM` als Wartungsrolle für Ihre Aufgaben zu verwenden. Die Verwendung dieser Rolle und der zugehörigen Richtlinie, `AmazonSSMServiceRolePolicy` , für Wartungsfenster-Aufgaben wird nicht mehr empfohlen. Erstellen Sie stattdessen eine benutzerdefinierte Richtlinie und Rolle für Wartungsfenster-Aufgaben. Weitere Informationen erhalten Sie unter Einrichten von Maintenance Windows.	9. Juni 2022
Unterstützung für Portweiterleitung an Remote-Hosts für Session Manager	Session Manager unterstützt jetzt Port-Weiterleitungs-Sitzungen für Remote-Hosts. Der Remote-Host muss nicht von Systems Manager verwaltet werden. Weitere Informationen finden Sie unter Starting a session (port forwarding to remote host)Starten einer Sitzung (Port-Weiterleitung zum Remote-Host).	25. Mai 2022

Aktualisierter Inhalt: Anweisungen für die manuelle Installation SSM Agent auf Amazon EC2 Linux-Instances	Als Reaktion auf Kundenfeedback haben wir die Themen überarbeitet, die Anweisungen für die manuelle Installation SSM Agent auf EC2 Amazon-Instances enthalten. Diese Themen enthalten jetzt Befehle mit global verfügbaren Dateien, die Sie zur schnellen Installation auf beliebigen EC2 Instances kopieren und einfügen können. AWS-Region Diese Themen enthalten auch Informationen, die Ihnen beim Erstellen von Installationsbefehlen helfen, die Dateien verwenden, die in Ihrer eigenen Arbeitsregion verfügbar sind. Der letztere Ansatz wird empfohlen, wenn Sie den Agenten mit einem Skript oder einer Vorlage auf mehreren Instances installieren. Weitere Informationen finden Sie in den Anweisungen für Ihr Linux-Betriebssystem im Abschnitt Manuelle Installation SSM Agent auf EC2 Instanzen für Linux.	9. Mai 2022

Neues Thema: Amazon Machine Images (AMIs) mit SSM Agent vorinstalliert	Als Reaktion auf Kundenfeedback haben wir zentralisierte Informationen darüber, welche von AWS verwaltete AMIs SSM Agent vorinstalliert haben. Dieses Thema enthält auch Anweisungen, wie Sie überprüfen können, ob eine EC2 daraus erstellte Amazon-Instance erfolgreich installiert AMIs wurde und läuft. Für seltene Fälle, in denen der Agent möglicherweise nicht erfolgreich installiert wird bzw. installiert, aber nicht gestartet wird, stellen wir auch Informationen zum Starten oder manuellen Installieren des Agenten auf diesen Instances bereit. Details dazu finden Sie unter .Amazon Machine Images (AMIs) mit SSM Agent vorinstalliert.	8. Mai 2022
Neuer State Manager-Abschnitt	Es wurde ein neuer Abschnitt hinzugefügt, der die Details beschreibt, wann State Manager Zuordnugnen ausführt. Weitere Informationen finden Sie unter Über Zuordnungsplanung.	27. April 2022

Patch Manager unterstützt jetzt Rocky Linux	Sie können jetzt Patch Manager verwenden, um Rocky Linux-Knoten zu patchen. Viele der Regeln, die für RHEL 8 zum Patchen gelten, gelten auch für Rocky Linux. Rocky Linux 8 nutzt die neue `AWS-DefaultRockyLinuxPatchBaseline` . Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf RHEL, CentOS Stream und Rocky Linux.	14. April 2022

Patch Manager unterstützt jetzt CentOS Stream 8	Sie können jetzt Patch Manager verwenden, um CentOS Stream 8-Instances und Red Hat Enterprise Linux (RHEL) 4.4–4.5-Instances zu patchen. Viele der Regeln, die für RHEL 8 zum Patchen gelten, gelten auch für CentOS Stream 8. CentOS Stream 8 verwendet die `AWS-DefaultCentOSPatchBaseline` . Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf RHEL und CentOS Stream	4. April 2022

Erstellen einer Übernahmerolle für Change Manager	In einem neuen Abschnitt werden die Anforderungen für das Erstellen und Implementieren einer Übernahmerolle für Change Manager erläutert. Eine Übernahmerolle ist eine AWS Identity and Access Management (IAM)-Servicerolle, die es Change Manager ermöglicht, die in einer genehmigten Änderungsanforderung angegebenen Runbook-Workflows in Ihrem Namen sicher auszuführen. Die Rolle gewährt AWS Systems Manager (AWS STS) `AssumeRole` Vertrauen anChange Manager. Weitere Informationen finden Sie unter Konfigurieren von Rollen und Berechtigungen für Change Manager.	18. März 2022
Genehmigen oder Ablehnen von Change Manager-Änderungsanträgen auf einmal	In der Systems-Manager-Konsole können Sie jetzt mehrere Änderungsanträge auswählen, die in einem einzigen Vorgang genehmigt oder abgelehnt werden sollen. Weitere Informationen finden Sie unter Überprüfen und Genehmigen oder Ablehnen von Änderungsanforderungen (Konsole).	8. März 2022

Support für von 2022 verwaltete Knoten Rocky Linux und Windows Server	Systems Manager unterstützt von 2022 verwaltete Knoten Rocky Linux und Windows Server, einschließlich Edge-Geräte und Hybrid-Maschinen, die sich On-Premises oder bei anderen Cloud-Anbietern befinden. Um Systems Manager mit diesen Betriebssystemen verwenden zu können, müssen Sie alle erforderlichen System-Manager-Einrichtungsverfahren ausführen, einschließlich Verfahren für Hybrid-Umgebungen oder Edge-Geräte, falls zutreffend. Weitere Informationen erhalten Sie unter Einrichten von Systems Manager. Für Rocky Linux-Maschinen müssen Sie auch SSM Agent manuell installieren. Weitere Informationen finden Sie unter Manuelles Installieren eines SSM Agent auf Rocky Linux-Instances. Für Windows Server 2022 SSM Agent ist Amazon Elastic Compute Cloud (Amazon EC2) -Instances standardmäßig installiert.	1. März 2022

Automation die Anpassung an Ihre Nebenläufigkeitsanforderungen erlauben und Metriken zur Automation-Nutzung anzeigen	Sie können Automation jetzt erlauben, Ihr Kontingent für gleichzeitige Automatisierung automatisch anzupassen und die Automatisierungsnutzungsmetriken einzusehen, die veröffentlicht wurden. CloudWatch Weitere Informationen zur adaptiven Nebenläufigkeit finden Sie unter Zulassen, dass sich Automation an Ihre Nebenläufigkeitsanforderungen anpasst. Weitere Informationen zum Anzeigen von Automation-Nutzungsmetriken finden Sie unter Automation-Metriken mit Amazon überwachen CloudWatch.	27. Januar 2022

Automation die Anpassung an Ihre Nebenläufigkeitsanforderungen erlauben und Metriken zur Automation-Nutzung anzeigen	Sie können Automation jetzt erlauben, Ihr Kontingent für gleichzeitige Automatisierung automatisch anzupassen und die auf veröffentlichten Automation-Nutzungsmetriken einzusehen. CloudWatch Weitere Informationen zur adaptiven Nebenläufigkeit finden Sie unter Zulassen, dass sich Automation an Ihre Nebenläufigkeitsanforderungen anpasst. Weitere Informationen zum Anzeigen von Automation-Nutzungsmetriken finden Sie unter Automation-Metriken mit Amazon überwachen CloudWatch.	27. Januar 2022
Nach Kategorien geordnete Systems-Manager-Dokumente	Amazon-eigene Systems-Manager-Dokumente sind jetzt nach Typ und Kategorien geordnet, damit Sie die benötigten Dokumente leichter finden können.	13. Januar 2022

Integrationen für Automation erstellen und aufrufen	Ab sofort können Sie während einer Automatisierung Nachrichten über Webhooks senden, indem Sie eine Integration erstellen. Integrationen können während einer Automatisierung mit der neuen Aktion `aws:invokeWebhook` in Ihrem Runbook aufgerufen werden. Weitere Informationen zum Erstellen von Integrationen finden Sie unter Erstellen von Webhook-Integrationen für Automation. Weitere Informationen zur Aktion `aws:invokeWebhook` finden Sie unter aws:invokeWebhook – Automation-Webhook-Integration aufrufen.	13. Januar 2022
Funktionen, die in der neuen Version nicht verfügbar sind AWS-Region	Die folgenden Tools des Systems Manager sind derzeit in der neuen Region Asien-Pazifik (Jakarta) nicht verfügbar. • Application Manager • Change Calendar • Change Manager • Explorer • Fleet Manager • Incident Manager • Quick Setup	13. Dezember 2021

Anzeigen von Ressourcen-Preisdetails für eine Anwendung	Application Managerist AWS Fakturierung und Kostenmanagement über das Cost Explorer Explorer-Widget in integriert. Nachdem Sie den Cost Explorer in der Fakturierungs- und Kostenmanagement-Konsole aktiviert haben, zeigt das Cost-Explorer-Widget imApplication Manager Preisdaten für eine bestimmte Anwendung oder Anwendungskomponente ohne Container an. Sie können Filter im Widget verwenden, um Preisdaten nach verschiedenen Zeiträumen, Details und Preisarten in einem Balken- oder Liniendiagramm anzuzeigen. Weitere Informationen finden Sie unter Anzeigen von Übersichtsinformationen über eine Anwendung.	7. Dezember 2021
Prozesse verwalten mit Fleet Manager	Sie können jetzt Fleet Manager verwenden, um Prozesse auf Ihren Knoten zu verwalten. Weitere Informationen finden Sie unter Arbeiten mit Prozessen.	6. Dezember 2021

Terminologieänderung: verwaltete Instances sind jetzt verwaltete Knoten	Aufgrund der Unterstützung für AWS IoT Greengras s Kerngeräte wurde der Begriff verwaltete Instanz in den meisten Dokumente n von Systems Manager in Manager-Knoten geändert. Die Systems-Manager-Ko nsole, API-Aufrufe, Fehlermel dungen und SSM-Dokum ente verwenden weiterhin den Begriff Instance.	29. November 2021

Support für Edge-Geräte	Systems Manager unterstützt die folgenden Edge-Gerätekonfigurationen. • AWS IoT Greengrass: Systems Manager unterstützt jetzt jedes Gerät, das für die AWS IoT Greengrass Core-Software konfiguriert ist AWS IoT Greengrass und auf denen sie ausgeführt wird. Um Ihre AWS IoT Greengrass Kerngeräte zu integrieren, müssen Sie eine AWS Identity and Access Management (IAM-) Servicerolle erstellen. Sie müssen die AWS IoT Greengrass Konsole auch für die Bereitstellung SSM Agent als AWS IoT Greengrass Komponente auf Ihren Geräten verwenden. Weitere Informationen finden Sie unter Einrichtung AWS Systems Manager für Edge-Geräte. • Edge-Geräte in einer Hybridumgebung: Systems Manager unterstützt auch AWS IoT Core-Geräte und AWS Nicht-IoT-Geräte, nachdem Sie sie als lokale Maschinen konfiguriert haben. Für das	29. November 2021

	Onboarding Ihrer Geräte müssen Sie eine IAM-Servicerolle erstellen, eine Aktivierung für verwaltete Knoten für eine Hybrid-Umgebung erstellen und SSM Agent manuell auf Ihren Geräten installieren. Weitere Informationen finden Sie unter Einrichtung AWS Systems Manager für Hybridumgebungen	
Verbindung mit verwalteten Instances über Remote Desktop	Sie können jetzt Fleet Manager verwenden, um eine Verbindung zu verwalteten Windows-Instances mithilfe des Remote Desktop Protocol (RDP) herzustellen. Diese Remote-Desktop-Sitzungen, die von Amazon DCV unterstützt werden, bieten sichere Verbindungen zu Ihren Instances direkt von Ihrem Browser aus. Weitere Informationen finden Sie unter Verbinden über Remote Desktop.	23. November 2021

Angeben einer maximalen Sitzungsdauer und Lieferun von Gründen für Sitzungen	Sie können jetzt eine maximale Sitzungsdauer für alle Session Manager-S itzungen in einer AWS-Region in Ihrem AWS-Konto festlegen . Wenn eine Sitzung die von Ihnen angegebene Dauer erreicht, wird sie beendet. Sie können jetzt auch optional einen Grund hinzufüge n, wenn Sie eine Sitzung starten. Weitere Informationen finden Sie unter Angeben der maximalen Sitzungsdauer.	16. November 2021

Patch Manager unterstützt jetzt das Raspberry Pi OS-Betriebssystem	Sie können jetzt Patch Manager verwenden, um Raspberry Pi OS-Instances zu patchen. Patch Manager unterstützt das Patchen von Raspberry Pi OS 9 (Stretch) und 10 (Buster). Da es sich bei dem Raspberry Pi OS um ein Debian-basiertes Betriebssystem handelt, gelten dafür viele der gleichen Patch-Regeln wie für Debian Server. Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf Debian Server	16. November 2021
Zugreifen auf das Red-Hat-Knowledgebase-Portal	Verwenden Sie Fleet Manager, um auf das RHEL-Knowledgebase-Portal zuzugreifen, um Lösungen, Artikel, Dokumentationen und Videos zur Verwendung von Red-Hat-Produkten zu finden. Weitere Informationen finden Sie unter Zugriff auf das Red-Hat-Wissensdatenbank-Portal.	3. November 2021

Massenbearbeitung von OpsItems	OpsCenter unterstützt jetzt Massenbearbeitung von OpsItems. Sie können mehrere OpsItems auswählen und eines der folgenden Felder bearbeiten: Status, Priorität, Schweregrad, Kategorie. Weitere Informationen finden Sie unter Bearbeiten von OpsItems.	15. Oktober 2021
Erstellen Sie Eingabeparameter, mit denen Ressourcen aufgefüllt werden AWS	Sie können jetzt Eingabeparameter in Automation-Runbooks erstellen, die AWS-Ressourcen in AWS-Managementkonsole ausfüllen. Weitere Informationen finden Sie unter Eingabeparameter erstellen, mit denen Ressourcen aufgefüllt werden. AWS	14. Oktober 2021
Neue Abschalt-Option für Aufgabenaufrufe für Wartungsfenster	Sie können jetzt verhindern, dass neue Aufgabenaufrufe starten, nachdem die für ein Wartungsfenster angegebene Abschaltzeit erreicht wurde. Weitere Informationen finden Sie unter Zuweisen von Aufgaben zu einem Wartungsfenster (Konsole).	13. Oktober 2021

Patch Manager-Support für macOS 11.3.1 und 11.4 (Big Sur)	Amazon Elastic Compute Cloud (Amazon EC2) - Instances für macOS 11.3.1 und 11.4 (Big Sur) können jetzt mit gepatcht werden. Patch Manager Dies ist zusätzlich zur bestehenden Support für macOS 10.14.x (Mojave) und 10.15.x (Catalina). Informationen zum Arbeiten mit Patch Manager finden Sie unter AWS Systems ManagerPatch Manager.	1. Oktober 2021

Anwendungseinblicke in Application Manager	Application Managerintegriert sich in Amazon CloudWatch Application Insights. Application Insights identifiziert Schlüsselmetriken, Protokolle und Alarme und richtet diese für Ihre Anwendungsressourcen und Ihren Technologie-Stack ein. Application Insights überwacht fortlaufend Metriken und Protokolle, um Anomalien und Fehler zu erkennen und zu korrelieren. Wenn das System Fehler oder Anomalien erkennt, generiert Application Insights CloudWatch Ereignisse, anhand derer Sie Benachrichtigungen einrichten oder Maßnahmen ergreifen können. Sie können Application Insights auf den Tabs Übersicht und Überwachungin Application Manager aktivieren und ansehen. Weitere Informationen zu Application Insights finden Sie unter Was ist Amazon CloudWatch Application Insights im CloudWatch Amazon-Benutzerhandbuch.	21. September 2021

Importieren von Ereignissen aus anderen Kalendern in Change Calendar	Sie können nun die Ereignisse aus einem Drittanbieter-Kalender in einen Kalender in Change Calendar importieren. Zuvor musste jedes Ereignis manuell in einen Kalender eingegeben werden. Nachdem Sie einen Kalender von einem unterstützten Drittanbieter-Kalenderanbieter in eine iCalendar (`.ics`)-Datei exportiert haben, importieren Sie ihn in Change Calendar. Die Ereignisse sind dann in den Regeln für den offenen oder geschlossenen Kalender in Systems Manager enthalten. Zu den unterstützten Anbietern zählen iCloud-Kalender, Google-Kalender und Microsoft Outlook. Weitere Informationen finden Sie unter Importieren und Verwalten von Ereignissen aus Drittanbieter-Kalendern.	8. September 2021

Neue Tagging- und Runbook-Features in Application Manager	Tagging-Erweiterungen umfassen die Möglichkeit, Tags zu einer bestimmte n Ressource oder allen Ressourcen in einer Applicati on Manager-Anwendung hinzuzufügen oder sie zu entfernen. Runbook-E rweiterungen umfassen die Möglichkeit, eine gefilterte Liste von Runbooks für einen bestimmten Ressourcentyp anzuzeigen oder ein Runbook für alle Ressourcen desselben Typs zu initiieren. Weitere Informationen finden Sie unter Arbeiten mit Tags in Applicati on Manager und Arbeiten mit Runbooks in Application Manager.	31. August 2021
Neues Beispiel: Erstellen Sie eine Änderungsanforderung mit dem AWS CLI	Ein Beispiel für die Erstellun g einer Änderungsanforderu ng mit dem AWS CLI wurde dem Change Manager Kapitel hinzugefügt. Im Beispiel wird die `AWS-HelloWorldChan geTemplate` -Änderung svorlage und Folgendes `AWS-HelloWorld runbook` verwendet: • Erstellen von Änderungs anforderungen (AWS CLI)	20. August 2021

Neuer Abschnitt: Verwenden von Parametern in Amazon EKS	Einer neuer Abschnitt wurde dem Kapitel Parameter Store hinzugefügt. Dieses Thema enthält eine exemplarische Vorgehensweise zur Verwendung Ihrer Parameter in Amazon EKS-Clustern. Weitere Informationen finden Sie unter Verwenden von Parameter Store-Parametern im Amazon Elastic Kubernetes Service.	19. August 2021
Aktualisierte Patch Manager Lebenszyklus-Hooks	Patch Manager bietet jetzt einen Lebenszyklus-Hook – die Möglichkeit, ein Systems Manager-Befehlsdokument auszuführen – für einen zusätzlichen Punkt während der Patch-Operation Jetzt patchen. Wenn Sie einen Neustart der Instance planen, nachdem Sie Jetzt patchen ausgeführt haben, können Sie einen Lebenszyklus-Hook angeben, der nach Abschluss des Neustarts ausgeführt werden soll. Weitere Informationen finden Sie unter Lebenszyklus-Hooks „Patch now“ (Jetzt patchen) verwenden und unter Informationen über das `AWS-RunPatchBaselineWithHooks` SSM-Dokument.	9. August 2021

Automatische Genehmigungen jetzt für Change Manager-Anforderungen unterstützt	Sie können jetzt Änderungsvorlagen in Change Manager so konfigurieren, dass automatische Genehmigungen unterstützt werden. Dies bedeutet, dass Benutzer mit den erforderlichen IAM-Berechtigungen die Änderungsanforderung starten können, ohne dass eine zusätzliche Genehmigung erforderlich ist. Benutzer, die Zugriff auf Vorlagen für automatische Genehmigungen haben, können weiterhin Genehmiger angeben, wenn sie dies wünschen. Um Ihnen bei der Kontrolle Ihrer Change Manager-Prozesse zu helfen, sind Genehmigungen weiterhin für alle Anforderungen während der Change–Freeze-Perioden erforderlich. Weitere Informationen finden Sie unter den folgenden Themen: • Erstellen von Änderungsvorlagen • Erstellen von Änderungsanforderungen • Probieren Sie die Vorlage für AWS verwaltete `Hello World` Änderungen aus	30. Juli 2021

Betriebliche OpsCenter-Einblicke	OpsCenter analysiert automatisch OpsItems in Ihrem Konto und generiert Insights. Ein Insight enthält Informationen, die Ihnen vermitteln, wie viele Duplikat-OpsItems sich in Ihrem Konto befinden und welche Quellen sie erstellen. Insights bieten auch empfohlene bewährte Methoden und Automatisierungs-Runbooks, um Duplikat-OpsItems zu beheben. Weitere Informationen finden Sie unter Arbeiten mit operativen Einblicken.	13. Juli 2021
Angehaltene Instances in Fleet Manager aufrufen	Sie können jetzt anzeigen, welche Instances in der Fleet Manager-Konsole `running` und welche Instances `stopped` sind. Weitere Informationen finden Sie unter AWS Systems ManagerFleet Manager.	12. Juli 2021
Neues Thema: Verfassen von Automation-Runbooks	Das neue Thema Verfassen von Automatisierungs-Runbooksenthält Anleitungen und erläuternde Beispiele für die Erstellung von Inhalten für benutzerdefinierte Automatisierungs-Runbooks.	8. Juli 2021

AWS CloudFormation Erstellung von Stapel und Vorlagen in Application Manager	Application Managerhilft Ihnen bei der Bereitstellung und Verwaltung von Ressource n für Ihre Anwendungen durch die Integration mit CloudFormation. Sie können CloudFormation Vorlagen und Stacks in Application Manager erstellen, bearbeite n und löschen. Application Managerenthält auch eine Vorlagenbibliothek, in der Sie Vorlagen klonen, erstellen und speichern können. Application Managerund CloudFormation zeigt dieselben Informationen über den aktuellen Status eines Stacks an. Vorlagen und Vorlagenaktualisie rungen werden in Systems Manager gespeichert, bis Sie den Stack bereitstellen. Zu diesem Zeitpunkt werden die Änderungen auch angezeigt CloudFormation. Weitere Informationen finden Sie unter Arbeiten mit CloudFormation Stacks in Application Manager.	8. Juli 2021

Neues Thema: Automatisches Drehen privater Schlüssel für SSM Agent auf Hybrid-Instances	Das neue Thema Automatisches Drehen privater Schlüssel einrichten enthält Anweisungen zum Verbessern Ihres Sicherheitsstatus, indem Sie den SSM Agent so konfigurieren, dass private Schlüssel der hybriden Umgebung automatisch gedreht werden.	15. Juni 2021
Session ManagerPlugin für die AWS CLI Version 1.2.205.0	Eine neue Version des Session Manager Plugins für AWS CLI wurde veröffentlicht. Weitere Informationen finden Sie unter Aktuelle Version und Versionsverlauf des Session Manager-Plugins.	10. Juni 2021
Neue serviceverknüpfte IAM-Rolle	Wenn Sie OpsCenter betriebliche Erkenntnisse aktivieren, erstellt Systems Manager eine neue serviceverknüpfte AWS Identity and Access Management (IAM)-Rolle namens `AWSSSMOps InsightsServiceRolePolicy` . Weitere Informationen zu dieser Rolle finden Sie unter Verwenden von Rollen, um betriebliche Einblicke OpsItems in Systems Manager zu gewinnenOpsCenter: `AWSSSMOps InsightsServiceRolePolicy` .	9. Juni 2021

Neuer Patch Manager-Fehlerbehebungsinhalt von Linux	Ein neues Thema Fehler beim Ausführen von `AWS-RunPatchBaseline` unter Linux enthält Beschreibungen und Lösungen für verschiedene Probleme, die beim Patchen verwalteter Instances mit Linux-Betriebssystemen auftreten können.	8. Juni 2021

Verbesserte Unterstützung für Wartungsfenster-Aufgaben, für die keine festgelegten Ziele erforderlich sind (Konsole)	Sie können jetzt Wartungsfenster-Aufgaben in der Konsole erstellen, ohne ein Ziel in der Aufgabe angeben zu müssen, falls dies nicht erforderlich ist. Bisher war diese Option nur verfügbar, wenn die API AWS CLI oder verwendet wurde. Diese Option gilt für die AWS Step Functions Aufgabentypen Automatisierung und. AWS Lambda Wenn Sie beispielsweise eine Automatisierungsaufgabe erstellen und die zu aktualisierenden Ressourcen in den Dokument-Parametern für Automatisierung angegeben sind, müssen Sie kein Ziel mehr in der Aufgabe selbst angeben. Weitere Informationen finden Sie unter Registrieren von Wartungsfensteraufgaben ohne Ziele, Zuweisen von Aufgaben zu einem Wartungsfenster (Konsole) und Planen von Automatisierungen mit Wartungsfenstern.	28. Mai 2021

Referenz zu Automation-Runbooks verschoben	Die Referenz zum Automation-Runbook wurde an einen neuen Speicherort verschoben. Weitere Informationen finden Sie unter Referenz zum Systems Manager Automation-Runbook.	10. Mai 2021
AWS Systems Manager Incident Manager starten	Incident Manager ist eine Incident-Management-Konsole, die Benutzern hilft, Vorfälle, die sich auf ihre AWS gehosteten Anwendungen auswirken, zu minimieren und diese zu beheben. Weitere Informationen finden Sie im AWS Systems Manager Incident Manager -Benutzerhandbuch.	10. Mai 2021
State Manager unterstützt Change Calendar	Sie können jetzt Change Calendar Namen oder Amazon-Ressourcennamen (ARNs) angeben, wenn Sie eine State Manager Zuordnung erstellen oder aktualisieren. State Managerwendet Verknüpfungen nur an, wenn der Änderungskalender geöffnet ist, nicht, wenn er geschlossen ist. Weitere Informationen finden Sie unter Erstellen von Zuordnungen und Bearbeiten und Erstellen einer neuen Version einer Zuordnung.	6. Mai 2021

Klonen von Systems Manager-Dokumenten	Mit der Systems Manager-Dokumentenkonsole können Sie nun Inhalte aus einem vorhandenen Dokument in ein neues Dokument kopieren, das Sie ändern können. Weitere Informationen hierzu finden Sie unter Klonen eines SSM-Dokuments.	4. Mai 2021
Integrieren Sie Security Hub CSPM mit und ExplorerOpsCenter	Sie können jetzt Explorer und OpsCenter mit integrieren. AWS Security Hub CSPM Security Hub CSPM bietet einen umfassenden Überblick über Ihren Sicherheitsstatus AWS und hilft Ihnen dabei, Ihre Umgebung anhand von Industriestandards und Best Practices zu überprüfen. Wenn es integriert istExplorer, können Sie die Sicherheitsergebnisse im Security Hub CSPM-Widget auf dem Explorer Dashboard einsehen. Wenn es integriert istOpsCenter, können Sie OpsItems für Security Hub CSPM-Ergebnisse erstellen. Weitere Informationen finden Sie unter Empfangen von Ergebnissen von AWS Security Hub CSPM in Explorer und Empfangen von Ergebnissen von AWS Security Hub CSPM in. OpsCenter	27. April 2021

Neues Thema: Dokumentkonventionen	Wir haben ein neues Thema hinzugefügt, um den Benutzern die allgemeinen typografischen Konventionen für das AWS Systems Manager -Benutzerhandbuch zu vermitteln. Weitere Informationen finden Sie unter Document Conventions.	21. April 2021
Aktualisiertes Thema: Informationen zum Patchen von Anwendungen, die von Microsoft unter veröffentlicht wurdenWindows Server	Aus dem Thema About patching applications released by Microsoft on Windows Server (Informationen zum Patchen von Anwendungen, die von Microsoft unter Patch Manager veröffentlicht werden) geht hervor, dass die Windows-Update-Option Give me updates for other Microsoft products when I update Windows (Updates für andere Microsoft-Produkte bereitstellen, wenn ich ein Windows-Update ausführe) auf der Instance aktiviert sein muss, damit Microsoft-Anwendungen auf Ihren von Windows Server verwalteten Instances patchen kann.	12. April 2021

Neuorganisation der Automation-Runbook-Referenz	Um Ihnen dabei zu helfen, die benötigten Runbooks zu finden und effizienter durch die Referenz zu navigieren, haben wir den Inhalt in der Automation-Runbook-Referenz nach dem jeweiligen AWS-Service umorganisiert. Diese Änderungen finden Sie unter Referenz zum Systems Manager Automation-Runbook.	12. April 2021

Patch Manager: Generieren von CSV-Patch-Compliance-Berichten	Patch Manager unterstützt jetzt die Möglichkeit, Patch-Compliance-Berichte für Ihre Instances zu erstellen und den Bericht in einem S3-Bucket Ihrer Wahl im CSV-Format zu speichern. Anschließend können Sie mit einem Tool wie Amazon Quick Suite die Daten des Patch-Compliance-Berichts analysieren. Sie können einen Patch-Compliance-Bericht für eine einzelne Instance oder für alle Instances in Ihrem AWS-Konto generieren. Sie können bei Bedarf einen einmaligen Bericht generieren oder einen Zeitplan für die automatische Erstellung von Berichten einrichten. Sie können auch ein Thema zum Amazon Simple Notification Service angeben, um Benachrichtigungen zu erhalten, wenn ein Bericht erstellt wird. Weitere Informationen finden Sie unter Erstellen von CSV-Patch-Compliance-Berichten.	9. April 2021

Löschen von Parameter Store-Parameterbezeichnungen	Sie können jetzt Parameter Store-Parameterbeschriftungen mithilfe der Systems Manager Konsole oder der AWS CLI löschen. Weitere Informationen finden Sie im Artikel zum Arbeiten mit Parameterbezeichnungen.	6. April 2021
Planen eines Neustarts der Instance bei Verwendung von Patch Now (Jetzt patchen)	Patch Manager unterstützt jetzt die Planung einer Zeit für den Neustart Ihrer Instances nach der Installation von Patches mit dem „Patch now“ (Jetzt patchen)-Feature. Dies gilt zusätzlich zu den vorhandenen Optionen, Instances nur neu zu starten, wenn dies erforderlich ist, um eine Patch-Installation abzuschließen oder den Neustart nach dem Patch-Vorgang zu überspringen. Informationen finden Sie unter Instances auf Abruf patchen .	1. April 2021
Neues Thema: Entdecken von öffentlichen Parametern	Parameter StoreÖffentliche Parameter können jetzt über die AWS CLI oder Systems Manager Manager-Konsole gefunden werden. Weitere Informationen finden Sie unter Auffinden von öffentlichen Parametern.	1. April 2021

„Patch now“ (Jetzt patchen)-Updates: Protokolle in S3 speichern und Lebenszyklus-Hooks ausführen	Wenn Sie die Patch Manager Patch now (Jetzt patchen)-Operation ausführen, können Sie einen S3-Bucket auswählen, in dem Patchprotokolle automatisch gespeichert werden sollen. Darüber hinaus können Sie Systems Manager Command-Dokumente (SSM-Dokumente) an drei Punkten während des Vorgangs als Lebenszyklus-Hooks ausführen: Vor der Installation, Nach der Installation und Beim Verlassen. Weitere Informationen finden Sie unter Instances auf Abruf patchen .	31. März 2021
Systems Manager meldet jetzt Änderungen an seinen AWS verwalteten Richtlinien	Ab dem 24. März 2021 werden Änderungen an verwalteten Richtlinien im Thema Systems ManagerAktualisierungen AWS verwalteter Richtlinien veröffentlicht. Bei der ersten aufgeführten Änderung handelt es sich um die Hinzufügung der Unterstützung für das Explorer Tool zur Berichterstattung OpsData und OpsItems für mehrere Konten und Regionen.	24. März 2021

Explorerermöglicht automatisch alle OpsData Quellen für die Synchronisierung von Ressourcendaten auf der Grundlage von Konten in AWS Organizations	Wenn Sie eine Ressourcendatensynchronisierung erstellen und eine der AWS Organizations Optionen wählen, lässt Systems Manager automatisch alle OpsData Quellen in der ausgewählten AWS-Regionen für alle AWS-Konten in Ihrer Organisation (oder in den ausgewählten Organisationseinheiten) zu. Dies bedeutet beispielsweise, dass Systems Manager automatisch Daten aus dieser Region sammelt AWS-Region OpsData , auch wenn Sie keine zugelassen Explorer haben, wenn Sie eine AWS Organizations Option für Ihre Ressourcendatensynchronisierung auswählen. Weitere Informationen finden Sie unter Synchronisierung mehrerer Konto- und Regions-Ressourcendaten.	24. März 2021
Systems Manager Automation bietet eine neue Systemvariable für Ihre Runbooks	Mit der neuen `global:AWS_PARTITION` Systemvariablen können Sie bei der Erstellung Ihrer AWS Runbooks die Partition angeben, in der sich eine Ressource befindet. Weitere Informationen finden Sie unter Automation-Systemvariablen.	18. März 2021

Zulassen mehrerer Genehmigungsebenen für Change Manager-Änderungsanforderungen	Wenn Sie eine Change Manager-Änderungsvorlage erstellen, können Sie jetzt verlangen, dass mehr als eine Genehmigerebene die Berechtigung für die Ausführung einer Änderungsanforderung erteilen. Sie können beispielsweise verlangen, dass technische Prüfer eine Änderungsanforderung, die aus einer Änderungsvorlage erstellt wurde, zuerst genehmigen und dann eine zweite Genehmigungsebene von einem oder mehreren Managern anfordern. Weitere Informationen finden Sie unter Erstellen von Änderungsvorlagen.	4. März 2021
Patch Manager unterstützt jetzt Oracle Linux 8.x	Sie können jetzt Patch Manager verwenden, um Oracle Linux 8.x-Instances bis Version 8.3 zu patchen. Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf Oracle Linux	1. März 2021

OpsCenter zeigt andere OpsItems für eine ausgewählte Ressource an	Um Ihnen bei der Untersuchung von Problemen zu helfen und den Kontext für ein Problem bereitzustellen, können Sie sich eine Liste von Ressourcen OpsItems für eine bestimmte AWS Ressource anzeigen lassen. In der Liste werden Status, Schweregrad und Titel der einzelnen OpsItem angezeigt. Die Liste enthält auch Deep-Links zu jedem OpsItem. Weitere Informationen finden Sie unter Andere OpsItems für eine bestimmte Ressource anzeigen.	1. März 2021
Definieren von Patching-Voreinstellungen zur Laufzeit	Sie können jetzt mit dem Baseline-Override-Feature Patching-Voreinstellungen zur Laufzeit definieren. Weitere Informationen finden Sie unter Verwenden des BaselineOverride Parameters.	25. Februar 2021

Neuer Systems Manager-Dokumenttyp	AWS CloudFormation Vorlagen können jetzt als Systems Manager Manager-Dokumente gespeichert werden. Durch das Speichern von CloudFormation Vorlagen als Systems Manager Manager-Dokumente können Sie von den Funktionen von Systems Manager Manager-Dokumenten wie Versionierung, Vergleich von Versionsinhalten und Teilen mit Konten profitieren. Weitere Informationen finden Sie unter AWS Systems Manager-Dokumente.	9. Februar 2021

Patch-Instances mit optionalen Hooks	Das neue SSM-Dokument `AWS-RunPatchBaselineWithHooks` bietet Hooks, mit denen Sie SSM-Dokumente an drei Punkten während des Instance-Patching-Zyklus ausführen können. Weitere Informationen zu `AWS-RunPatchBaselineWithHooks` finden Sie unter Informationen über das SSM-Dokument `AWS-RunPatchBaselineWithHooks`. Eine exemplarische Vorgehensweise für einen Patching-Vorgang, der alle drei Hooks verwendet, finden Sie unter Exemplarische Vorgehensweise: Aktualisieren von Anwendungsabhängigkeiten, Patchen einer Instance und Durchführen einer anwendungsspezifischen Zustandsprüfung.	2. Februar 2021

Neues Thema:Überprüfen von On-Premises-Servern und virtuellen Maschinen mit einem Hardware-Fingerabdruck	SSM Agentüberprüft anhand eines berechneten Fingerabdrucks die Identität von lokalen Servern und virtuellen Maschinen und stellt sicher VMs , dass Sie sich beim Dienst registrieren. Der Fingerabdruck ist eine undurchsichtige Zeichenfolge, die im Tresor gespeichert ist und die der Agent an bestimmte Systems Manager APIs weiterleitet. Informationen zum Hardware-Fingerabdruck und Anweisungen zum Konfigurieren eines Ähnlichkeitsschwellenwerts zur Unterstützung der Maschinenverifizierung finden Sie unter Überprüfen On-Premises-Server und virtueller Computer mithilfe eines Hardware-Fingerabdrucks.	25. Januar 2021
Neues Thema: Technische SSM Agent-Referenz	Das Thema SSM Agenttechnische Referenz enthält Informationen, die Ihnen helfen sollen, den Agenten zu implementieren AWS Systems Manager SSM Agent und zu verstehen, wie er funktioniert. Dieses Thema enthält einen brandneuen Abschnitt mit SSM Agentfortlaufenden Updates von AWS-Regionen.	21. Januar 2021

SSM Agent auf Windows Server 2008	Seit dem 14. Januar 2020 wird Windows Server 2008 für Feature- oder Sicherheitsupdates von Microsoft nicht mehr unterstützt. Windows Server 2008 AMIs enthalten zwar den SSM Agent, aber der Agent wird für dieses Betriebssystem nicht mehr aktualisiert.	5. Januar 2021
Verbesserte Unterstützung für Aufgaben im Wartungsfenster, für die keine bestimmten Ziele (AWS CLI und nur API) erforderlich sind	Sie können jetzt Aufgaben im Wartungsfenster erstellen, ohne in der Aufgabe ein Ziel angeben zu müssen, falls eines nicht erforderlich ist (AWS CLI und nur über die API). Dies gilt für Automatisierung AWS Lambda und AWS Step Functions Aufgabentypen. Wenn Sie beispielsweise eine Automatisierungsaufgabe erstellen und die zu aktualisierenden Ressourcen in den Runbook-Parametern für Automatisierung angegeben sind, müssen Sie kein Ziel mehr in der Aufgabe selbst angeben. Weitere Informationen finden Sie unter Registrieren von Wartungsfensteraufgaben ohne Ziele und Planen von Automatisierungen mit Wartungsfenstern.	23. Dezember 2020

Neue Automation-Features	Eine neue freigegebene Eigenschaft wurde zu Systems Manager Automation Runbooks hinzugefügt. Mit der `onCancel`-Eigenschaft können Sie angeben, zu welchem Schritt die Automatisierung gehen soll, falls ein Benutzer die Automatisierung abbricht. Weitere Informationen finden Sie unter Eigenschaften, die von allen Aktionen gemeinsam genutzt werden.	21. Dezember 2020
Neues Thema: Arbeiten mit Zuordnungen mithilfe von IAM	Es wurde ein neues Thema zum Systems Manager State Manager-Kapitel hinzugefügt, das die bewährten Methoden zum Erstellen von Zuordnungen mit IAM beschreibt. Weitere Informationen finden Sie unter Arbeiten mit Mappings mithilfe von IAM.	18. Dezember 2020
State Manager unterstützt jetzt Multi-Regionen und Multi-Konten	Verknüpfungen können jetzt mit mehreren Regionen oder Konten erstellt oder aktualisiert werden. Weitere Informationen finden Sie unter Erstellen von Zuordnungen.	15. Dezember 2020

Neues Tool: Fleet Manager	Fleet Manager, ein Tool in AWS Systems Manager, ist eine einheitliche Benutzeroberfläche (UI), mit der Sie Ihre Serverflotte, die vor Ort oder vor Ort läuft AWS, remote verwalten können. Mit Fleet Manager können Sie sich den Zustand und den Leistungsstatus Ihrer gesamten Serverflotte von einer Konsole aus ansehen. Sie können auch Daten aus einzelnen Instances sammeln, um allgemeine Problembehandlungs- und Verwaltungsaufgaben über die Konsole auszuführen. Weitere Informationen finden Sie unter AWS Systems Manager Fleet Manager.	15. Dezember 2020

Neues Tool: Change Manager	Amazon Web Services hat Change Manager, ein unternehmensweites Change-Management-Framework zum Anfordern, Genehmigen, Implementieren und Melden von Betriebsänderungen an Ihrer Anwendungskonfiguration und Infrastruktur, veröffentlicht. Wenn Sie ein einziges delegiertes Administratorkonto verwenden AWS Organizations, können Sie Änderungen an mehreren oder mehreren AWS-Konten verwalten. AWS-Regionen Alternativ können Sie mit einem lokalen Konto Änderungen für einen einzigen AWS-Konto verwalten. Wird Change Manager für die Verwaltung von Änderungen sowohl an AWS Ressourcen als auch an lokalen Ressourcen verwendet. Weitere Informationen finden Sie unter AWS Systems Manager Change Manager.	15. Dezember 2020

Neues Tool: Application Manager	Application Managerhilft Ihnen dabei, Probleme mit Ihren AWS Ressourcen im Kontext Ihrer Anwendungen zu untersuchen und zu beheben. Application Managerfasst Betriebsinformationen aus mehreren Tools AWS-Servi ces und Systems Manager Manager-Tools in einem einzigen AWS-Manag ementkonsole zusammen. Weitere Informationen finden Sie unter AWS Systems Manager Application Manager.	15. Dezember 2020

AWS Systems Manager unterstützt EC2 Amazon-Instances für macOS	Zeitgleich mit der Veröffentlichung der Amazon Elastic Compute Cloud (Amazon EC2) -Unterstützung für macOS Instances unterstützt Systems Manager jetzt viele Operationen auf EC2 Instances fürmacOS. Zu den unterstützten Versionen gehören macOS 10.14.x (Mojave) und 10.15.x (Catalina). Weitere Informationen finden Sie unter den folgenden Themen. • Informationen zur Installation SSM Agent auf EC2 Instances für macOS finden Sie unter Installation und Konfiguration SSM Agent auf EC2 Instances für macOS. . • Informationen zum Patchen von EC2 Instanzen für macOS finden Sie unter So werden Patches installiert und Eine benutzerdefinierte Patch-Baseline erstellen (macOS). • Allgemeine Informationen zur Unterstützung von EC2 Instances für macOS finden Sie unter Amazon EC2 Mac-Instances im EC2 Amazon-Benutzerhandbuch.	30. November 2020

Pseudo-Parameter des Wartungsfensters: Neuer Ressourcentyp wird für `{{TARGET_ID}}` und `{{RESOURCE_ID}}` unterstützt	Ein zusätzlicher Ressourcentyp steht nun für die Verwendung mit den Pseudo-Parametern `{{TARGET_ID}}` und `{{RESOURCE_ID}}` zur Verfügung. Sie können jetzt den Ressourcentyp `AWS::RDS::DBCluster` mit diesen beiden Pseudo-Parametern verwenden. Weitere Informationen zu Pseudoparametern des Wartungsfensters finden Sie unter Verwenden von Pseudoparametern beim Registrieren von Wartungsfensteraufgaben.	27. November 2020
Session ManagerPlugin für die AWS CLI Version 1.2.30.0	Eine neue Version des Session Manager Plugins für AWS CLI wurde veröffentlicht. Weitere Informationen finden Sie unter Aktuelle Version und Versionsverlauf des Session Manager-Plugins.	24. November 2020
Neues Thema: SSM-Dokumentversionen vergleichen	Sie können nun die Unterschiede im Inhalt zwischen Versionen von SSM-Dokumenten in der Systems Manager-Dokumentenkonsole vergleichen. Weitere Informationen finden Sie unter Vergleichen von SSM-Dokumentversionen.	24. November 2020

Systems Manager unterstützt jetzt VPC-Endpunktrichtlinien	Sie können jetzt Richtlinien für VPC-Schnittstellenendpunkte für Systems Manager erstellen . Weitere Informationen finden Sie unter Erstellen einer VPC-Endpunktrichtlinie.	18. November 2020
Neues Thema: Angeben eines Zeitüberschreitungswerts für Leerlaufsitzungen	Sie können nun festlegen, wie lange ein Benutzer inaktiv sein soll, bevor eine Sitzung mit Session Manager endet. Weitere Informationen finden Sie unter Angeben eines Zeitüberschreitungswerts für Leerlaufsitzungen.	18. November 2020
Neues Session Manager-Protokollierungsfeature	Sie können jetzt einen kontinuierlichen Stream von Sitzungsdatenprotokollen im JSON-Format an Amazon Logs senden. CloudWatch Weitere Informationen finden Sie unter Streaming-Sitzungsdaten mithilfe von Amazon CloudWatch Logs.	18. November 2020
Neues Thema: Überprüfen der Signatur des SSM Agent	Sie können nun die kryptografische Signatur des Installationspakets für den SSM Agent auf Linux-Instances verifizieren. Weitere Informationen finden Sie unter SSM-Dokumentschemata und -Features.	17. November 2020

Neues Thema: Grundlegendes zu Automatisierungsstatus	Dem Kapitel „Systems Manager Automation“ wurde ein neues Thema hinzugefügt, in dem die Status für Aktionen und Automatisierungen beschrieben werden. Weitere Informationen finden Sie unter Grundlegendes zu Automation-Status.	17. November 2020
Neue Quelltypen für den `aws:downloadContent` -Plugin	Git und HTTP werden jetzt als Quelltypen für den `aws:downloadContent` -Plugin unterstützt. Weitere Informationen finden Sie unter `aws:downloadContent` .	17. November 2020
Neues Schemafeature für Systems Manager-Dokument (SSM-Dokument)	In SSM-Dokumenten mit Schema-Version 2.2 oder neuer unterstützt der `precondition` -Parameter jetzt die Referenzierung der Eingabeparameter Ihres Dokuments. Weitere Informationen finden Sie unter SSM-Dokumentschemata und -funktionen.	17. November 2020

Neue Datenquelle inExplorer: AWS Config	Explorerzeigt jetzt Informationen zur AWS Config Konformität an, einschließlich einer allgemeinen Zusammenfassung der konformen und nicht konformen AWS Config Regeln, der Anzahl der konformen und nicht konformen Ressourcen sowie spezifische Details zu den einzelnen Regeln (wenn Sie eine nicht konforme Regel oder Ressource genauer untersuchen). Weitere Informationen finden Sie unter Bearbeiten von Systems Manager-Datenquellen.	11. November 2020
Neues Thema: Ausführen von Auto-Scaling-Gruppen mit Zuordnungen	Es wurde ein neuer Abschnitt zu State Manager hinzugefügt, der die bewährten Methoden zum Erstellen von Zuordnungen zum Ausführen von Auto-Scaling-Gruppen beschreibt. Weitere Informationen finden Sie unter Ausführen von Auto-Scaling-Gruppen mit Zuordnungen.	10. November 2020
Quick Setup unterstützt jetzt das Targeting einer Ressourcengruppe	Quick Setup unterstützt nun die Auswahl einer Ressourcengruppe als Ziel für den lokalen Einrichtungstyp. Weitere Informationen finden Sie unter Auswählen von Zielen für Quick Setup.	5. November 2020

Patch Manager bietet Unterstützung für Debian Server 10 LTS, Oracle Linux 7.9 LTS und Ubuntu Server 20.10 STR	Sie können jetzt Patch Manager verwenden, um Debian Server 10 LTS-, Oracle Linux 7.9 LTS- und Ubuntu Server 20.10 STR-Instances zu patchen. Weitere Informationen finden Sie unter den folgenden Themen: • Patch Manager-Voraussetzungen • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf Debian Server • Funktionsweise von Patch-Baseline-Regeln auf Oracle Linux • Funktionsweise von Patch-Baseline-Regeln auf Ubuntu Server	4. November 2020

Neue Unterstützung für EventBridge AWS Systems ManagerChange Calendar	Amazon bietet EventBridge jetzt Unterstützung für Change Calendar Ereignisse und Ereignisse in den Veranstal tungsregeln. Wenn sich der Status eines Kalenders ändert, EventBridge können Sie die Zielaktion einleiten, die Sie als EventBridge Regel definiert haben. Informationen zum Arbeiten mit EventBridge und Systems Manager Manager-Ereignissen finden Sie in den folgenden Themen. • Konfiguration EventBrid ge für Systems Manager Manager-Ereignisse • Referenz: EventBridge Amazon-Ereignistypen und -muster für Systems Manager	4. November 2020

Für CloudWatch die Erstellung OpsItems aus Alarmen konfigurieren	Sie können Amazon so konfigurieren CloudWatch , dass automatisch ein OpsItem Systems Manager erstellt wirdOpsCenter, wenn ein Alarm in den `ALARM` Status wechselt. Auf diese Weise können Sie Probleme mit AWS Ressourcen von einer einzigen Konsole aus schnell diagnostizieren und beheben. Weitere Informationen finden Sie unter Konfiguration für CloudWatch die Erstellung OpsItems aus Alarmen.	4. November 2020
Unterstützung für Ubuntu Server 20.10	AWS Systems Manager unterstützt jetzt Ubuntu Server 20.10 Short-Term Release (STR). Weitere Informationen finden Sie unter den folgenden Themen: • Unterstützte Betriebssysteme • Installieren von SSM Agent für eine Hybrid-Umgebung (Linux) • Manuelle Installation von SSM Agent auf Ubuntu Server-Instances • Prüfen des SSM Agent-Status und Starten des Agenten	22. Oktober 2020

Neues Thema: Konfigurierbare Shell-Profile zulassen	Sie können jetzt konfiguri erbare Shell-Profile mit Session Manager zulassen. Indem Sie konfigurierbare Shell-Profile zulassen, können Sie Voreinstellungen innerhalb von Sitzungen wie Shell-Einstellungen, Umgebungs variablen, Arbeitsverzeichnis sen und Ausführen mehrerer Befehle beim Starten einer Sitzung anpassen. Weitere Informationen finden Sie unter Konfigurierbare Shell-Profile zulassen.	21. Oktober 2020
In den Ergebnissen zur Patch-Konformität wird jetzt angegeben, welche Patches für die Problembehebung verantwortlich CVEs sind	Wenn Sie bei den meisten unterstützten Linux-Systemen Patch-Compliance-Ergebnisse für Ihre verwalteten Instances anzeigen, geben die Details, die Sie jetzt sehen können, an, welche Bulletin-Probleme mit Common Vulnerabilities and Exposure (CVE) durch welche verfügbaren Patches behoben werden. Mithilfe dieser Informationen können Sie feststellen, wie dringend Sie einen fehlenden oder fehlgeschlagenen Patch installieren müssen. Weitere Informationen finden Sie unter Anzeigen von Patch-Com pliance-Ergebnissen.	20. Oktober 2020

Erweiterte Unterstützung für Linux-Patch-Metadaten	Sie können jetzt viele Details zu verfügbaren Linux-Patches in Patch Manager aufrufen. Sie können Patch-Daten wie Architektur, Epoche, Version, CVE-ID, Advisory ID, Bugzilla ID, Repository und mehr anzeigen. Darüber hinaus wurde die API-Operation DescribeAvailablePatches aktualisiert, um Linux-Bet riebssysteme und Filterung gemäß diesen neu verfügbar en Patch-Metadatentypen zu unterstützen. Weitere Informati onen finden Sie unter den folgenden Themen: • Anzeigen verfügbarer Patches • DescribeAvailablePatches und Patch in der AWS Systems Manager -API-Refe renz • describe-available-patchesim AWS Systems Manager Abschnitt der AWS CLI Befehlsreferenz	16. Oktober 2020
Session ManagerPlugin für die AWS CLI Version 1.2.7.0	Eine neue Version des Session Manager Plugins für AWS CLI wurde veröffentlicht. Weitere Informationen finden Sie unter Aktuelle Version und Versionsverlauf des Session Manager-Plugins.	15. Oktober 2020

Neues Thema: Schema des Sitzungsdokuments	Das neue Thema Schema des Sitzungsdokuments beschreibt die Schemaelemente für ein Sitzungsdokument. Mithilfe dieser Informationen können Sie benutzerdefinierte Sitzungsdokumente erstellen, in denen Sie Einstellungen für die Sitzungstypen festlegen, die Sie mit Session Manager verwenden.	15. Oktober 2020
Neues Thema: Freitextsuche für SSM-Dokumente	Das Suchfeld auf der Seite Systems Manager-Dokumente unterstützt jetzt die freie Textsuche. Die Freitextsuche vergleicht den bzw. die eingegebenen Suchbegriffe mit dem Dokumentnamen in jedem SSM-Dokument. Weitere Informationen finden Sie unter Verwenden Der Freitextsuche.	15. Oktober 2020
Neues Thema: Fehlerbehebung bei der Verfügbarkeit von Amazon EC2 Managed Instances	Das neue Thema Fehlerbehebung bei der Verfügbarkeit von Amazon EC2 Managed Instances hilft Ihnen zu untersuchen, warum eine EC2 Amazon-Instance, von der Sie bestätigt haben, dass sie läuft, nicht in den Listen verfügbarer verwalteter Instances in Systems Manager verfügbar ist.	6. Oktober 2020

Neuorganisation des Kapitels Parameter Store	Um Ihnen dabei zu helfen, die benötigten Informationen effizienter zu finden, haben wir die Inhalte im Kapitel Parameter Store des AWS Systems Manager -Benutzer handbuch neuorganisiert. Die meisten Inhalte sind jetzt in den Abschnitten Einrichte n vonParameter Store und Arbeiten mit Parameter Store organisiert. Darüber hinaus wurde das Thema AWS Systems ManagerParameter Store um folgende Unterabsc hnitte erweitert: • Welche Vorteile bietet Parameter Store meiner Organisation? • An wen richtet sich Parameter Store? • Über welche Features verfügt Parameter Store? • Was ist ein Parameter?	1. Oktober 2020

Neue Themen im Zusammenhang mit Patch-Compliance	Die folgenden Themen wurden hinzugefügt, um Ihnen zu helfen, verwaltete Instances zu identifizieren, die keine Patch-Compliance haben, die verschiedenen Arten von Patch-Compliance-Scans zu verstehen und die entsprechenden Schritte zu ergreifen, damit Ihre Instances die Compliance erfüllen. • Identifizierung nicht konformer Instances • Patchen nicht konformer Instances • Anzeigen der Patch-Compliance-Ergebnisse	24. September 2020
SSM Agent Version 3.0	Systems Manager hat eine neue Version von SSM Agent veröffentlicht.	21. September 2020

Neue und aktualisierte Themen: Amazon EventBridge ersetzt CloudWatch Events für Eventmanagement	CloudWatch Events und EventBridge sind derselbe zugrunde liegende Service und dieselbe API, EventBridge bieten aber mehr Funktionen und sind jetzt die bevorzugte Methode zur Verwaltung Ihrer Veranstaltungen in AWS. (Änderungen, die Sie in einer der beiden CloudWatch oder in jeder Konsole vornehmen, EventBridge spiegeln sich in jeder Konsole wider.) Die Verweise auf CloudWatch Ereignisse und bestehende Verfahren im gesamten AWS Systems Manager Benutzerhandbuch wurden aktualisiert, um der EventBridge Unterstützung Rechnung zu tragen. Darüber hinaus wurden die folgenden neuen Themen hinzugefügt. • Überwachen von Systems Manager-Ereignissen • Konfiguration EventBridge für Systems Manager Manager-Ereignisse • Beispiele für Zieltypen von Systems Manager • Referenz: EventBridge Amazon-Ereignistypen und -muster für Systems Manager	18. September 2020

Integrieren AWS Security Hub CSPM und Patch Manager	Sie können jetzt integrieren Patch Manager mit AWS Security Hub CSPM. Security Hub CSPM bietet einen umfassenden Überblick über Ihren Sicherheitsstatus AWS und hilft Ihnen dabei, Ihre Umgebung anhand von Industriestandards und Best Practices zu überprüfen. Wenn Security Hub CSPM integriert istPatch Manager, überwacht es den Patching-Status Ihrer Flotten unter Sicherheitsgesichtspunkten. Weitere Informationen finden Sie unter Integration mit. Patch ManagerAWS Security Hub CSPM	17. September 2020

Pseudo-Parameter des Wartungsfensters: Neue Ressourcentypen werden für `{{TARGET_ID}}` und `{{RESOURCE_ID}}` unterstützt	Beim Registrieren einer Wartungsfenster-Aufgabe geben Sie mithilfe der Option `--task-invocation-parameters` die Parameter an, die für jede der vier Arten von Aufgaben eindeutig sind. Sie können auch mithilfe der Pseudoparameter-Syntax wie `{{TARGET_ID}}` und `{{RESOURCE_ID}}` auf bestimmte Werte verweisen. Während der Ausführung übergibt die Wartungsfenster-Aufgabe anstelle der Pseudoparameter-Platzhalter richtige Werte. Zwei zusätzliche Ressourcentypen stehen nun für die Verwendung mit den Pseudo-Parametern `{{TARGET_ID}}` und `{{RESOURCE_ID}}` zur Verfügung. Sie können jetzt die Ressourcentypen `AWS::RDS::DBInstance` und `AWS::SSM::ManagedInstance` mit diesen beiden Pseudo-Parametern verwenden. Weitere Informationen zu Pseudoparametern des Wartungsfensters finden Sie unter Verwenden von Pseudoparametern beim Registrieren von Wartungsfensteraufgaben.	14. September 2020

Patchen von Instances auf Abruf mit der neuen Option „Patch now“ (Jetzt patchen)	Sie können jetzt die Systems Manager-Konsole verwenden, um Instances zu patchen oder nach fehlenden Patches zu suchen. Sie können dies tun, ohne einen Zeitplan erstellen oder ändern zu müssen, oder vollständige Patch-Konfigurationsoptionen angeben, um einem sofortigen Patch-Bedarf gerecht zu werden. Sie müssen nur angeben, ob Patches gescannt oder installiert werden sollen, und müssen die Ziel-Instances angeben. Patch Manager wendet automatisch die aktuelle Standard-Patch-Baseline für Ihre Instance-Typen an und wendet bewährte Methoden dazu, wie viele Instances gleichzeitig gepatcht werden und wie viele Fehler zulässig sind, bevor der Vorgang fehlschlägt. Weitere Informationen finden Sie unter Instances auf Abruf patchen .	9. September 2020

Neues Thema: Prüfen des SSM Agent-Status und Starten des Agenten	Das neue Thema Prüfen des SSM Agent-Status und Starten des Agentenbietet Befehle, um zu überprüfen, ob SSM Agent auf jedem unterstützenden Betriebssystem ausgeführt wird. Es enthält auch die Befehle, mit denen der Agent gestartet wird, wenn er nicht ausgeführt wird.	7. September 2020
Patch Manager unterstützt jetzt Ubuntu Server 20.04 LTS	Sie können jetzt Patch Manager verwenden, um Ubuntu Server 20.04 LTS-Instances zu patchen. Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf Ubuntu Server	31. August 2020
Neues Thema für häufige Anwendungsfälle und bewährte Methoden	Wir haben ein neues Thema hinzugefügt, um den Benutzern schnell die Unterschiede zwischen Maintenance Windows und State Manager zu vermitteln. Weitere Informationen finden Sie unter Wählen zwischen State Manager und Maintenance Windows.	28. August 2020

Neue OpsCenter-Features	OpsCenter enthalten neue Features, mit denen Sie Automatisierungs-Runbooks schnell finden und ausführen können, um Probleme zu beheben. Weitere Informationen finden Sie unter Automatisierungs-Runbook-Funktionen in OpsCenter.	19. August 2020
Neue Datenquelle inExplorer: AWS -Support Fällen	Explorerzeigt jetzt Informationen über Support Fälle an. Sie müssen entweder ein Unternehmens- oder ein Geschäftskonto mit eingerichtet haben Support. Weitere Informationen finden Sie unter Bearbeiten von Systems Manager-Datenquellen.	13. August 2020
Distributor bietet jetzt ein Drittanbieter-Paket von Trend Micro.	Distributor enthält jetzt ein Drittanbieter-Paket von Trend Micro. Sie können Distributor verwenden, um den Trend Micro Cloud One Agent auf Ihren verwalteten Instances zu installieren. Trend Micro Cloud One hilft Ihnen, Ihre Workloads in der Cloud zu sichern. Weitere Informationen finden Sie unter AWSDistributor.	12. August 2020

Der `aws:configurePackage` -Dokument-Plugin enthält jetzt den Parameter AdditionalArguments.	Das Systems Manager Command-Dokument-Plugin `aws:configurePackage` unterstützt jetzt die Bereitstellung zusätzlicher Parameter für Ihre Skripte (Installation, Deinstallation und Update) mit dem neuen `additionalArguments` -Parameter. Weitere Informationen finden Sie im Thema `aws:configurePackage` .	11. August 2020
AppConfigDer -Inhalt wurde in ein separates Benutzerhandbuch verschoben	Informationen zu AWS AppConfig wurden in ein separates Benutzerhandbuch verschoben. Weitere Informationen finden Sie unter Was ist AWSAppConfig? AppConfig hat auch eine separate Landingpage zur Dokumentation mit Links zum Benutzerhandbuch, zur AppConfig API-Referenz und zu einem neuen AppConfig Workshop.	3. August 2020

Quick Setupunterstützt jetzt AWS Organizations	Quick Setupunterstützt jetzt AWS Organizations die schnelle Konfigura tion der erforderlichen Sicherheitsrollen und häufig verwendeten Systems Manager Manager-Tools für mehrere Konten und Regionen. Weitere Informati onen finden Sie unter AWS Systems ManagerQuick Setup.	23. Juli 2020
Neue Datenquelle inExplorer: Zuordnungs-Compliance	Explorer zeigt jetzt Zuordnung s-Compliance-Daten von State Manager an. Weitere Informationen finden Sie unter Bearbeiten von Systems Manager-Datenquellen.	23. Juli 2020

Neues Systems Manager-Befehlsdokument zum Ein- und Ausschalten von Kernel Live Patching	Das Dokument `AWS-ConfigureKernelLivePatching` ist nun für die Verwendung mit Run Command verfügbar, wenn Sie Kernel Live Patching auf Amazon Linux 2-Instances ein- oder ausschalten möchten. Dieses Dokument ersetzt die Notwendigkeit, eigene benutzerdefinierte Befehlsdokumente für diese Aufgaben zu erstellen. Weitere Informationen finden Sie unter Verwenden von Kernel-Live-Patching auf Amazon Linux 2-Instances	22. Juli 2020
Aktualisierte Automation-Kontingente	Service quotas für Automation wurden aktualisiert, einschließlich einer separaten Warteschlange für die Automatisierung der Ratenregelung. Weitere Informationen finden Sie unter AWS Systems Manager Automation.	20. Juli 2020

Angeben der Anzahl der Zeitplanversatztage für ein Wartungsfenster mithilfe der Konsole	Über die Systems Manager-Konsole können Sie nun eine Anzahl von Tagen angeben, die nach dem in einem CRON-Ausdruck angegebenen Datum und der angegebenen Uhrzeit gewartet werden soll, bevor ein Wartungsfenster ausgeführt wird. (Bisher war diese Option nur verfügbar , wenn Sie ein AWS SDK oder ein Befehlszeilentool verwendeten.) Wenn Ihr CRON-Ausdruck beispiels weise die Ausführung eines Wartungsfensters am dritten Dienstag jedes Monats um 23:30 Uhr plant – `cron(0 30 23 ? * TUE#3 *)` – und Sie einen Zeitplanversatz von 2 angeben, wird das Fenster erst zwei Tage später um 23:30 Uhr ausgeführt. Weitere Informationen finden Sie unter Cron- und Rate-Ausd rücke für Systems Manager und Angeben der Anzahl der Zeitplanversatztage für ein Wartungsfenster.	17. Juli 2020

Aktualisierung PowerShell mit Run Command	Um Ihnen bei der Aktualisierung PowerShell auf Version 5.1 auf Ihren Windows Server 2012- und 2012 R2-Instances zu helfen, haben wir dem AWS Systems Manager Benutzerhandbuch eine exemplarische Vorgehensweise hinzugefügt. Weitere Informationen finden Sie unter Update PowerShell mit Run Command.	30. Juni 2020
Patch Manager unterstützt jetzt CentOS 8.0 und 8.1	Sie können jetzt Patch Manager verwenden, um CentOS 8.0- und 8.1-Instances zu patchen. Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • So installieren Sie den SSM Agent in Hybrid-Linux-Knoten	27. Juni 2020

AppConfigintegriert mit AWS CodePipeline	AppConfigist eine integrierte Bereitstellungsaktion für AWS CodePipeline (CodePipeline). CodePipeline ist ein vollständig verwalteter Continuous Delivery Service, der Sie bei der Automatisierung Ihrer Release-Pipelines für schnelle und zuverlässige Anwendungs- und Infrastrukturupdates unterstützt. CodePipeline automatisiert die Erstellungs-, Test- und Bereitstellungsphasen Ihres Release-Prozesses bei jeder Codeänderung auf der Grundlage des von Ihnen definierten Release-Modells. Die Integration von AppConfig mit CodePipeline bietet die folgenden Vorteile. Weitere Informationen finden Sie unter AppConfigIntegration mit CodePipeline. • Kunden, die früher CodePipeline die Orchestrierung verwalteten, verfügen jetzt über eine einfache Möglichkeit, Konfigurationsänderungen an ihren Anwendungen vorzunehmen, ohne ihre gesamte Codebasis bereitstellen zu müssen. • Kunden, die das System AppConfig zur Verwaltun	25. Juni 2020

	g von Konfigurationsbere itstellungen verwenden möchten, aber nur eingeschr änkt zur Verfügung stehen, weil ihr aktueller Code oder Konfigurationsspeicher AppConfig nicht unterstützt wird, stehen jetzt zusätzlic he Optionen zur Verfügung . CodePipeline unterstützt AWS CodeCommitGitHub, und BitBucket (um nur einige zu nennen).	
Neues Kapitel: Produkt- und Service-Integrationen	Damit Sie besser verstehen , wie Systems Manager mit AWS-Services anderen Produkten und Services integriert werden kann, wurde dem AWS Systems Manager Benutzerhandbuch ein neues Kapitel hinzugefügt. Weitere Informationen finden Sie unter Produkt- und Service-Integrationen mit Systems Manager.	23. Juni 2020

Umstrukturierung des Automation-Kapitels	Damit Sie leichter finden, was Sie suchen, haben wir Themen im Automation-Kapitel des AWS Systems Manager - Benutzerhandbuch umstrukturiert. Beispielsweise sind die Automation-Aktionen und -Runbooks jetzt Abschnitte der obersten Ebene des Kapitels. Weitere Informationen finden Sie unter AWS Systems Manager Automation.	23. Juni 2020

Angeben der Anzahl der Zeitplanversatztage für ein Wartungsfenster	Mit einem Befehlszeilentool oder AWS SDK können Sie jetzt angeben, wie viele Tage nach dem in einem CRON-Ausdruck angegebenen Datum und Uhrzeit gewartet werden sollen, bevor ein Wartungsfenster ausgeführt wird. Wenn Ihr CRON-Ausdruck beispielsweise die Ausführung eines Wartungsfensters am dritten Dienstag jedes Monats um 23:30 Uhr plant – `cron(0 30 23 ? * TUE#3 *)` – und Sie einen Zeitplanversatz von 2 angeben, wird das Fenster erst zwei Tage später um 23:30 Uhr ausgeführt. Weitere Informationen finden Sie unter Cron- und Rate-Ausdrücke für Systems Manager und Angeben der Anzahl der Zeitplanversatztage für ein Wartungsfenster.	19. Juni 2020

Patch Manager-Unterstützung für Kernel-Live-Patching auf Amazon Linux 2-Instances	Kernel-Live-Patching für Amazon Linux 2 ermöglicht es Ihnen, Patches für Schwachstellen und kritische Fehler auf einen laufenden Linux-Kernel anzuwenden, ohne Neustarts oder Unterbrechungen der laufenden Anwendungen. Sie können jetzt das Feature aktivieren und Kernel-Live-Patches mithilfe von Patch Manager anwenden. Informationen finden Sie unter Verwenden von Kernel-Live-Patching auf Amazon Linux 2-Instances.	16. Juni 2020
Patch Manager erhöht die Oracle Linux-Versionsunterstützung	Bisher hat Patch Manager nur Version 7.6 von Oracle Linux unterstützt. Wie in den Patch Manager-Voraussetzungen aufgeführt, deckt der Support jetzt die Versionen 7.5–7.8 ab.	16. Juni 2020

Beispielszenario für die Verwendung des `InstallOverrideList` -Parameters in Patch-Operationen	Das neue Thema Beispielszenario für die Verwendung des Parameters `InstallOverrideList` beschreibt eine Strategie für die Verwendung des Parameters `InstallOverrideList` im Dokument `AWS-RunPatchBaseline` , um verschiedene Patch-Typen auf eine Zielgruppe in verschiedenen Wartungsfenstern bei Verwendung einer einzelnen Patch-Baseline anzuwenden.	11. Juni 2020
Vordefinierte Bereitstellungsstrategien für AppConfig	AppConfig bietet jetzt vordefinierte Bereitstellungsstrategien. Weitere Informationen finden Sie unter Erstellen einer Bereitstellungsstrategie.	10. Juni 2020

Patch Manager unterstützt jetzt Red Hat Enterprise Linux (RHEL) 7.8-8.2	Sie können Patch Manager jetzt zum Patchen von RHEL 7.8–8.2-Instances verwenden . Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf RHEL • Manuelle Installation von SSM Agent auf Red Hat Enterprise Linux-Instances • So installieren Sie den SSM Agent in Hybrid-Linux-Knoten	9. Juni 2020

Explorer unterstützt das Delegieren der Administration	Wenn Sie Explorer Daten aus mehreren zusammenf ühren AWS-Regionen und AWS-Konten die Ressource ndatensynchronisierung mit verwenden AWS Organizat ions, empfehlen wir Ihnen, einen delegierten Administr ator für zu konfigurieren. Explorer Ein delegierter Administrator verbessert die Explorer-Sicherheit, indem die Anzahl der Explorer-Administr atoren, die die Synchroni sierung von Ressource ndaten für mehrere Konten und Regionen erstellen oder löschen können, auf eine einzelne Person eingeschr änkt wird. Sie müssen auch nicht mehr am AWS Organizat ions -Hauptkonto angemeldet sein, um die Synchronisierung von Ressourcendaten in Explorer zu verwalten. Weitere Informationen finden Sie unter Konfigurieren eines delegiert en Administrators.	3. Juni 2020

State Manager-Zuordnung erst für das nächste angegebene Cron-Intervall übernehmen	Wenn Sie nicht möchten, dass eine State Manager-Zuordnung unmittelbar nach der Erstellung ausgeführt wird, wählen Sie die Option Zuordnung erst beim nächsten angegebenen Cron-Intervall anwenden in der Systems Manager-Konsole aus. Weitere Informationen finden Sie unter Erstellen von Zuordnungen.	3. Juni 2020
Neue Datenquelle inExplorer: AWS Compute Optimizer	Explorerzeigt jetzt Daten von an AWS Compute Optimizer . Dazu gehören die Anzahl der zu wenig bereitgestellten und zu viel bereitgestellten EC2 Instances, Optimierungsergebnisse, On-Demand-Preisdetails und Empfehlungen zu Instance-Typ und Preis. Weitere Informationen finden Sie AWS Compute Optimizer in den Einzelheiten zur Einrichtung unter Einrichtung verwandter Dienste.	26. Mai 2020

Installieren von Windows Service Packs und Linux-Nebenversionsupgrades mit Patch Manager	Das neue Thema Tutorial: Erstellen einer Patch-Baseline für die Installation von Windows Service Packs (Konsole) zeigt, wie Sie eine Patch-Baseline erstellen, die ausschließlich der Installation von Windows Service Packs gewidmet ist. Das Thema Erstellen einer benutzerdefinierten Patch-Baseline (Linux) wurde mit Informationen zum Einschluss von Nebenversionsupgrades für Linux-Betriebssysteme in Patch-Baselines aktualisiert.	21. Mai 2020
Neuorganisation des Kapitels Parameter Store	Alle Themen, die sich mit Konfigurations- oder Einrichtungsoptionen für Parameter Store-Operationen befassen, wurden im Abschnitt Einrichtung von Parameter Store zusammengefasst. Dies umfasst auch die Themen Verwalten von Parameterstufen und Erhöhen des Parameter Store-Durchsatzes, die aus anderen Teilen des Kapitels hierher verschoben wurden.	18. Mai 2020

Neues Thema zum Erstellen von Datums- und Uhrzeitzeichenfolgen für die Interaktion mit Systems Manager-API-Operationen.	Im neuen Thema Erstellen formatierter Datums- und Uhrzeitzeichenfolgen für Systems Manager wird die Erstellung formatierter Datums- und Uhrzeitzeichenfolgen für die Interaktion mit Systems Manager-API-Operationen beschrieben.	13. Mai 2020
Informationen zu Berechtigungen zum Verschlüsseln von Parametern SecureString	Im neuen Thema Beschränken des Zugriffs auf Systems Manager Manager-Parameter mithilfe von IAM-Richtlinien wird der Unterschied zwischen der Verschlüsselung Ihrer `SecureString` Parameter mit AWS KMS key und der Von AWS verwalteter Schlüssel Verwendung von bereitgestellt von erklärt. AWS	13. Mai 2020

Patch Manager unterstützt jetzt Debian Server und Oracle Linux 7.6 Betriebssysteme	Sie können jetzt Patch Manager verwenden, um Debian Server- und Oracle Linux-Instances zu patchen. Patch Managerunterstützt Patchen der Versionen Debian Server 8.x und 9.x und Oracle Linux 7.6. Weitere Informationen finden Sie unter den folgenden Themen: • Wie Sicherheitspatches ausgewählt werden • Wie Patches installiert werden • Funktionsweise von Patch-Baseline-Regeln auf Debian Server • Funktionsweise von Patch-Baseline-Regeln auf Oracle Linux	7. Mai 2020
Erstellen Sie Zuordnungen, State Manager die darauf abzielen AWS -Ressourcengruppen	Zusätzlich zur Ausrichtung auf Tags, einzelne Instances und alle Instances in Ihrem AWS-Konto können Sie jetzt State Manager-Zuordnungen erstellen, die Instances in AWS -Ressourcengruppen zum Ziel haben. Weitere Informationen finden Sie unter Informationen zu Zielen und Ratensteuerungen in State Manager-Zuordnungen.	7. Mai 2020

Geben Sie zur Überprüfung einen neuen `aws:ec2:image` Datentyp ein Parameter StoreAMI IDs	Wenn Sie einen `String`-Parameter erstellen, können Sie jetzt einen Datentyp als `aws:ec2:image` angeben, um sicherzustellen, dass der eingegebene Parameterwert ein gültiges Amazon Machine Image (AMI)-ID-Format aufweist. Durch die Unterstützung von AMI-ID-Formaten können Sie vermeiden, dass alle Skripts und Vorlagen jedes Mal mit einer neuen ID aktualisiert werden, wenn sich das AMI ändert, das Sie in Ihren Prozessen verwenden möchten. Sie können einen Parameter mit dem Datentyp `aws:ec2:image` erstellen und für seinen Wert die ID eines AMI eingeben. Dies ist das AMI, von dem Sie neue Instances erstellen möchten. Sie verweisen dann in Ihren Vorlagen und Befehlen auf diesen Parameter. Wenn Sie ein anderes AMI verwenden möchten, aktualisieren Sie den Parameterwert. Parameter Store validiert die neue AMI-ID und Sie müssen Ihre Skripts und Vorlagen nicht aktualisieren. Weitere Informationen finden Sie unter Unterstützung systemeigener Parameter für Amazon Machine Image IDs.	5. Mai 2020

Verwalten von Beendigungscodes in Run Command-Befehlen	In Run Command können Sie definieren, wie Beendigungscodes in Ihren Skripts gehandhabt werden. Standardmäßig wird der Beendigungscode des letzten in einem Skript ausgeführten Befehls als Beendigungscode für das gesamte Skript gemeldet. Sie können jedoch mit der folgenden Methode eine bedingte Shell-Anweisung einschließen, damit das Skript beendet wird, wenn ein Befehl vor dem letzten Befehl fehlschlägt. Beispiele finden Sie im neuen Thema Verwalten von Beendigungscodes in Run Command-Befehlen.	5. Mai 2020

Neue öffentliche Parameter für Availability Zones und lokale Zonen freigegeben	Öffentliche Parameter wurden veröffentlicht, um Informationen zu AWS Availability Zones und lokalen Zonen programmgesteuert verfügbar zu machen. Diese Parameter ergänzen die vorhandenen öffentlichen Parameter für die globale Infrastruktur für AWS-Services und AWS-Regionen. Weitere Informationen finden Sie unter Öffentliche Parameter aufrufen für AWS-Services, Regionen, Endpunkte, Availability Zones, Local Zones und Wavelength Zones.	4. Mai 2020
Neue Datenquelle inExplorer: AWS Trusted Advisor	Explorerzeigt jetzt Daten von an AWS Trusted Advisor. Dies umfasst den Status von Überprüfungen zu bewährten Methoden und Empfehlungen in den folgenden Bereichen: Kostenoptimierung, Sicherheit, Fehlertoleranz, Leistung und Service Quotas. Weitere Informationen finden Sie Trusted Advisor in den Details zur Einrichtung unter Verwandte Dienste einrichten.	4. Mai 2020

Erstellen von State Manager-Zuordnungen, die Chef-Rezepte ausführen	Sie können State Manager-Zuordnungen erstellen, die Chef-Kochbücher und -Rezepte ausführen, indem Sie das `AWS-ApplyChefRecipes` -Dokument verwenden. Dieses Dokument bietet die folgenden Vorteile für die Ausführung von Chef-Rezepten: • Unterstützt mehrere Versionen von Chef (Chef 11 bis Chef 14). • Installiert die Chef-Clientsoftware automatisch auf Ziel-Instances. • Führt optional Systems Manager-Compliance-Prüfungen für Ziel-Instances aus und speichert die Ergebnisse der Compliance-Prüfungen in einem S3-Bucket. • Führt mehrere Cookbooks und Rezepte in einem einzigen Durchlauf des Dokuments aus. • Führt optional Rezepte im `why-run`-Modus aus, um anzuzeigen, welche Rezepte sich auf Ziel-Instances ändern, ohne Änderungen vorzunehmen.	19. März 2020

	• Wendet optional benutzerdefinierte JSON-Attribute auf `chef-client` -Durchläufe an. Weitere Informationen finden Sie unter Erstellen von Zuordnungen, die Chef-Rezepte ausführen	
Synchronisieren Sie Inventardaten von mehreren AWS-Konten zu einem zentralen Amazon S3 S3-Bucket	Sie können Systems Manager Manager-Inventardaten von mehreren AWS-Konten zu einem zentralen S3-Bucket synchronisieren. Die Konten müssen in definiert sein AWS Organizations. Weitere Informationen finden Sie unter Erstellen einer Inventory Resource Data Sync für mehrere Konten, die in AWS Organizations definiert sind.	16. März 2020
Speichern von AppConfig-Konfigurationen in Amazon S3	Zuvor unterstützte AppConfig nur Anwendungskonfigurationen, die in Systems Manager-(SSM-)Dokumenten oder Parameter Store-Parametern gespeichert wurden. Zusätzlich zu diesen Optionen unterstützt AppConfig jetzt das Speichern von Konfigurationen in Amazon S3. Weitere Informationen finden Sie unter Informationen über Konfigurationen in Amazon S3.	13. März 2020

SSM Agent ist standardmäßig auf Amazon ECS-optimierten installiertAMIs	SSM Agent ist jetzt standardmäßig auf Amazon ECS-optimierten AMIs installiert. Weitere Informationen finden Sie unter Arbeiten mit SSM Agent.	25. Februar 2020
Erstellen von AppConfig-Konfigurationen in der Konsole	Mit AppConfig können Sie jetzt bei Erstellung eines Konfigurationsprofils eine Anwendungskonfiguration in der Konsole erstellen. Weitere Informationen finden Sie unter Erstellen einer Konfiguration und eines Konfigurationsprofils.	13. Februar 2020

Nur Patches automatisch genehmigen, die bis zu einem bestimmten Datum veröffentlicht wurden	Neben der Option, Patches eine bestimmte Anzahl von Tagen nach der Veröffentlichung automatisch für die Installation zu genehmigen, unterstützt Patch Manager nun die Möglichkeit, nur Patches automatisch zu genehmigen, die an oder vor einem von Ihnen angegebenen Datum veröffentlicht wurden. Wenn Sie beispielsweise den 7. Juli 2020 als Stichtag in Ihrer Patch-Baseline angeben, werden keine Patches automatisch installiert, die an oder nach dem 8. Juli 2020 veröffentlicht wurden. Weitere Informationen finden Sie unter Informationen zu benutzerdefinierten Baselines und Arbeiten mit benutzerdefinierten Patch-Baselines (Konsole).	12. Februar 2020

Verwenden Sie den Pseudoparameter {{RESOURCE_ID}} in Wartungsfensteraufgaben	Geben Sie beim Registrieren einer Aufgabe im Wartungsfenster die Parameter an, die für den Aufgabentyp eindeutig sind. Sie können mithilfe der Pseudoparameter-Syntax wie `{{TARGET_ID}}` , `{{TARGET_TYPE}}` und `{{WINDOW_TARGET_ID}}` auf bestimmte Werte verweisen. Während der Ausführung übergibt die Wartungsfenster-Aufgabe anstelle der Pseudoparameter-Platzhalter richtige Werte. Zur Unterstützung von Ressourcen, die als Ziel Teil einer Ressourcengruppe sind, können Sie den `{{RESOURCE_ID}}` -Pseudoparameter verwenden, um Werte für Ressourcen wie DynamoDB-Tabellen, S3-Buckets und andere unterstützte Typen zu übergeben. Weitere Informationen finden Sie in den folgenden Themen in der Anleitung: Erstellen und Konfigurieren eines Wartungsfensters (AWS CLI): • Verwenden von Pseudo-Parametern bei der Registrierung von Aufgaben im Wartungsfenster	6. Februar 2020

	• Beispiele: Registrieren von Aufgaben für ein Wartungsfenster	
Befehle schnell erneut ausführen	Systems Manager enthält zwei Optionen, mit denen Sie einen Befehl von der Run CommandSeite in der AWS Systems Manager Konsole erneut ausführen können. Erneut ausführen : Über diese Schaltfläche können Sie denselben Befehl ausführen, ohne Änderungen daran vorzunehmen. In neu kopieren: Über diese Schaltfläche kopieren Sie die Einstellungen eines Befehls in einen neuen Befehl und erhalten die Möglichkeit, diese Einstellungen zu bearbeiten, bevor Sie den Befehl ausführen. Weitere Informationen finden Sie unter Befehle erneut ausführen.	5. Februar 2020

Wechsel vom Kontingent für erweiterte Instances zurück zum Kontingent für Standard-Instances	Wenn Sie zuvor alle On-Premises-Instances, die in Ihrer Hybrid-Umgebung ausgeführt werden, für die Verwendung des Kontingents für erweiterte Instances konfiguriert haben, können Sie diese Instances nun schnell so konfigurieren, dass sie das Kontingent für Standard-Instances verwenden. Die Rückkehr zur Stufe „Standard instanzen“ gilt für alle Hybrid-Instances in einer und einer AWS-Konto einzigen. AWS-Region Das Zurücksetzen auf das Standard-Instances-Kontingent wirkt sich auf die Verfügbarkeit einiger Systems-Manager-Tools aus. Weitere Informationen finden Sie unter Zurücksetzen des Kontingents für erweiterte Instances auf das Kontingent für Standard-Instances.	16. Januar 2020

Neue Option zum Überspringen von Instance-Neustarts nach der Patch-Installation	Zuvor wurden verwaltete Instances immer neu gestartet, nachdem der Patch Manager Patches auf ihnen installiert hatte. Mit einem neuen `RebootOption` -Parameter im SSM-Dokument `AWS-RunPatchBaseline` können Sie angeben, ob die Instances nach der Installation neuer Patches automatisch neu gestartet werden sollen. Weitere Informationen finden Sie unter Parametername: RebootOption im Thema Über das SSM-Dokument. `AWS-RunPatchBaseline`	15. Januar 2020
Neues Thema: „ PowerShell Skripts auf Linux-Instanzen ausführen“	Ein neues Thema, das beschreibt, wie Sie PowerShell Skripts auf Run Command Linux-Instances ausführen können. Weitere Informationen finden Sie unter PowerShell Skripts auf Linux-Instances ausführen.	10. Januar 2020

Aktualisierungen zum ‚Konfigurieren von SSM Agent zur Nutzung eines Proxys‘	Die Werte, die bei der Konfiguration von SSM Agent zur Nutzung eines Proxys anzugeben sind, wurden aktualisiert, um die Optionen sowohl für HTTP-Proxyserver als auch für HTTPS-Pro xyserver wiederzugeben. Weitere Informationen finden Sie unter Konfigurieren von SSM Agent zur Nutzung eines Proxys.	9. Januar 2020

Das neue Kapitel „Sicherheit“ beschreibt Verfahren zur Sicherung von Systems Manager-Ressourcen	Im neuen Kapitel Sicherheit im Benutzerhandbuch von AWS Systems Manager wird erläutert, wie Sie das Modell der geteilten Verantwortung beim Einsatz von Systems Manager anwenden können. Die Themen in diesem Kapitel zeigen Ihnen, wie Sie Systems Manager konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie lernen auch, wie Sie andere verwenden können AWS-Services , die Ihnen helfen, Ihre Systems Manager Manager-Ressourcen zu überwachen und zu sichern. **Note** Als Teil dieses Updates wurde das Kapitel „Authentifizierung und Zugriffskontrolle“ im Benutzerhandbuch durch einen neuen, einfacheren Abschnitt Identity and Access Management für AWS Systems Manager ersetzt.	24. Dezember 2019

Neue eigene Automation-Beispielrunbooks	Eine Reihe von eigenen Automation-Beispielrunbooks wurde dem Benutzerhandbuch hinzugefügt. Diese Beispiele zeigen, wie verschiedene Automation-Aktionen verwendet werden können, um die Bereitstellung, Fehlerbehebung und Wartungsaufgaben zu vereinfachen, und sollen Ihnen helfen, Ihre eigenen benutzerdefinierten Automation-Runbooks zu schreiben. Weitere Informationen finden Sie unter Beispiele für benutzerdefinierte Automation-Runbooks. Sie können den Inhalt des von Amazon verwalteten Automation-Runbooks auch in der Systems Manager-Konsole anzeigen. Weitere Informationen finden Sie unter Referenz zum Systems Manager Automation-Runbook.	23. Dezember 2019

Support für Oracle Linux	Systems Manager unterstützt jetzt Oracle Linux 7.5 und 7.7. Informationen zur manuellen Installation SSM Agent auf EC2 Instances für Oracle Linux Instances finden Sie unter Oracle Linux. Informationen zur Installation von SSM Agent auf Oracle Linux-Servern in einer Hybridumgebung finden Sie unter So installieren Sie den SSM Agent auf Hybrid-Linux-Knoten.	19. Dezember 2019

Starten Sie Session Manager Sitzungen von der EC2 Amazon-Konsole aus	Sie können jetzt Session Manager Sitzungen von der Amazon Elastic Compute Cloud (Amazon EC2) -Konsole aus starten. Für die Arbeit mit sitzungsbezogenen Aufgaben von der EC2 Amazon-Konsole aus sind unterschiedliche IAM-Berechtigungen für Benutzer und Administratoren erforderlich. Sie können Berechtigungen für die Verwendung der Session Manager Konsole und AWS CLI nur für die Verwendung der EC2 Amazon-Konsole oder für die Verwendung aller drei Tools erteilen. Weitere Informationen finden Sie unter den folgenden Themen. • Schnellstart: Standard IAM-Richtlinien für Session Manager • Eine Sitzung starten (EC2 Amazon-Konsole)	18. Dezember 2019

CloudWatch Unterstützung für Run Command Metriken und Alarme	AWS Systems Manager veröffentlicht jetzt Metriken über den Status von Run Command Befehlen an CloudWatch, sodass Sie Alarme auf der Grundlage dieser Metriken einrichten können. Zu den Terminals tatuswerten für Befehle, für die Sie Metriken verfolgen können, gehören `Success`, `Failed` und `Delivery Timed Out`. Weitere Informati onen finden Sie unter Run CommandMetriken mit Amazon überwachen CloudWatch.	17. Dezember 2019
Neues Systems Manager-T ool: Change Calendar	Verwenden Sie Systems Manager Change Calendar, um Zeiträume (Ereignisse) anzugeben, in denen Sie Codeänderungen (z. B. von Systems Manager Automation-Runbooks oder AWS Lambda -Funktionen) an Ressourcen einschränken oder verhinder n möchten. Ein Änderungs kalender ist ein neuer Systems Manager-Dokumenttyp, der iCalendar 2.0-Daten im Klartextformat speichert. Weitere Informationen finden Sie unter AWS Systems Manager Change Calendar.	11. Dezember 2019

Neues Systems Manager Manager-Tool: AWSAppConfig	Mit AppConfig können Sie Anwendungskonfigurationen schnell erstellen, verwalten und bereitstellen. AppConfig unterstützt kontrollierte Bereitstellungen für Anwendungen jeder Größe. Sie können es AppConfig mit Anwendungen verwenden, die auf EC2 Instances AWS Lambda, Containern, mobilen Anwendungen oder IoT-Geräten gehostet werden. Um Fehler bei der Bereitstellung von Anwendungskonfigurationen zu vermeiden, enthält AppConfig Validierungen. Eine Validierung stellt durch eine syntaktische oder semantische Prüfung sicher, dass die Konfiguration, die Sie bereitstellen möchten, wie beabsichtigt funktioniert. Während einer Konfigurationsbereitstellung überwacht AppConfig die Anwendung, um sicherzustellen, dass die Bereitstellung erfolgreich ist. Falls im System ein Fehler auftritt oder die Bereitstellung einen Alarm startet, setzt AppConfig die Änderung zurück, um die Auswirkungen auf die Benutzer Ihrer Anwendung zu minimieren.	25. November 2019

Weitere Informationen finden Sie unter AWSAppConfig.

Neues Systems Manager-Tool: Systems Manager Explorer	AWS Systems Manager Explorer ist ein anpassbares Operations-Dashboard, das Informationen über Ihre AWS Ressourcen meldet. Explorerzeigt eine aggregierte Ansicht der Betriebsdaten (OpsData) für Sie AWS-Konten und Across AWS-Regionen an. OpsData Enthält Metadaten zu Ihren EC2 Instanzen, Details zur Patch-Konformität und betriebliche Arbeitselemente (OpsItems). Explorer Explorerbietet Informationen darüber, wie sie auf Ihre Geschäftsbereiche oder Anwendungen verteilt OpsItems sind, wie sie sich im Laufe der Zeit entwickeln und wie sie sich je nach Kategorie unterscheiden. Sie können Informationen in Explorer gruppieren und filtern, um sich auf die Elemente zu konzentrieren, die für Sie relevant sind und eine Aktion erfordern. Wenn Sie Probleme mit hoher Priorität erkennen, können Sie mit Systems Manager OpsCenter Automation-Runbooks ausführen und die Probleme schnell beheben. Weitere Informationen finden Sie unter AWS Systems Manager Explorer.	18. November 2019

> **Note**
>
> Die Einrichtung für Systems Manager OpsCenter ist in die Einrichtung für Explorer integriert. Wenn Sie OpsCenter bereits eingerichtet haben, müssen Sie das integrierte Setup trotzdem ausführen, um die Einstellungen und Optionen zu überprüfen. Wenn Sie OpsCenter nicht eingerichtet haben, können Sie mit dem integrierten Setup beide Tools einrichten. Weitere Informationen finden Sie unter Erste Schritte mit Explorer und OpsCenter.

Verbesserte Parameter suchfunktionen	Mit den Werkzeugen für die Parametersuche können Sie Parameter jetzt leichter finden, wenn Sie viele Parameter in Ihrem Konto haben oder sich nicht an den genauen Namen eines Parameters erinnern. Mit dem Suchwerkzeug können Sie nach `contains` filtern. Zuvor unterstützten die Suchwerkzeuge die Suche nach Parameternamen nur mit `equals` und `begins-with` . Weitere Informationen finden Sie unter Suche nach öffentlichen Systems Manager-Parametern.	15. November 2019

Neuer konsolenbasierter Document Builder für Automation | Unterstützung für die Ausführung von Skripts in Automation-Schritten	Sie können Systems Manager Automation jetzt verwenden, um standardisierte Betriebsp laybooks zu erstellen und gemeinsam zu nutzen, um die Konsistenz zwischen Benutzern AWS-Konten, und AWS-Regionen sicherzus tellen. Mit der Möglichkeit, Skripts auszuführen und Ihren Automation-Runbooks mit Markdown eingebundene Dokumentation hinzuzufügen, können Sie Fehler reduziere n und manuelle Schritte wie das Navigieren in schriftlicher Prozeduren in Wikis und das Ausführen von Terminalb efehlen eliminieren. Weitere Informationen finden Sie unter den folgenden Themen. • Walkthrough: Verwenden von Document Builder zum Erstellen eines benutzerd efinierten Automatisierungs-Runbooks • `aws:executeScript` (Referenz zu Automation-Aktionen) • Erstellen von Automation-Runbooks mit Document Builder	14. November 2019

	• Neue Automation-Funktionen in Systems Manager im AWS News-Blog	
Ausführen einer direkten Paketaktualisierung mit Distributor	Wenn Sie früher ein Update für ein Paket mit Distributor installieren wollten, bestand Ihre einzige Möglichkeit darin, das gesamte Paket zu deinstallieren und die neue Version neu zu installieren. Jetzt können Sie stattdessen eine direkte Aktualisierung durchführen. Während einer direkten Aktualisierung installiert Distributor entsprechend dem Aktualisierungsskript, das Sie Ihrem Paket hinzufügen, nur Dateien, die neu sind oder seit der letzten Installation geändert wurden. Bei Verwendung dieser Option muss Ihre Paketanwendung während der Aktualisierung nicht offline geschaltet werden, sodass sie weiterhin verfügbar ist. Weitere Informationen finden Sie unter den folgenden Themen. • Erstellen eines Pakets • Installieren oder Aktualisieren von Paketen	11. November 2019

Neues Feature für automatische SSM Agent-Aktualisierungen	Mit einem Klick können Sie alle Instanzen in Ihrem so konfigurieren, AWS-Konto dass automatisch nach neuen Versionen von SSM Agent gesucht und diese heruntergeladen werden. Wählen Sie dazu auf der Seite Verwaltete Instanzen in der AWS Systems Manager Konsole die Option auto Agentenaktualisierung aus. Weitere Informationen finden Sie unter Automatische Aktualisierungen von SSM Agent.	5. November 2019

Beschränken Sie Session Manager den Zugriff mithilfe der von AWS Ihnen bereitgestellten Tags	Eine zweite Methode zur Steuerung des Benutzerzugriffs auf Sitzungsaktionen ist jetzt verfügbar. Mit dieser neuen Methode können Sie IAM-Zugriffsrichtlinien mithilfe von AWS-bereitgestellten Sitzungs-Tags erstellen, anstatt die Variable `{aws:username}` zu verwenden. Die Verwendung dieser von AWS-mitgelieferten Sitzungs-Tags ermöglicht es Organisationen, den Benutzerzugriff auf Sitzungen mithilfe von Federated IDs zu kontrollieren. Weitere Informationen finden Sie unter Benutzer können nur von ihnen gestartete Sitzungen beenden.	2. Oktober 2019

Neues SSM-Command-Dokument wendet Ansible Playbooks an	Sie können State Manager-Zuordnungen erstellen, die Ansible Playbooks ausführen , indem Sie das `AWS-ApplyAnsiblePlaybooks` -Dokument verwenden. Dieses Dokument bietet die folgenden Vorteile für die Ausführung von Playbooks: • Unterstützung für die Ausführung komplexer Playbooks • Support für das Herunterladen von Playbooks von GitHub und Amazon Simple Storage Service (Amazon S3) • Unterstützung für komprimierte Playbook-Struktur • Erweiterte Protokollierung • Möglichkeit, anzugeben , welches Playbook ausgeführt werden soll, wenn Playbooks gebündelt werden Weitere Informationen finden Sie unter Erstellen von Zuordnungen, die Ansible Playbooks ausführen	24. September 2019

Unterstützung der Port-Weiterleitung für Session Manager	Session Manager unterstützt jetzt Port-Weiterleitungssitzungen. Die Port-Weiterleitung ermöglicht es Ihnen, Tunnel zwischen Ihren in privaten Subnetzen bereitgestellten Instances sicher zu erstellen, ohne den SSH-Service auf dem Server zu starten, den SSH-Port in der Sicherheitsgruppe zu öffnen oder einen Bastion-Host zu verwenden. Ähnlich wie bei SSH-Tunneln ermöglicht Ihnen die Port-Weiterleitung, Datenverkehr auf Ihrem Laptop weiterzuleiten, um Ports auf Ihrer Instance zu öffnen. Sobald die Port-Weiterleitung konfiguriert ist, können Sie eine Verbindung mit dem lokalen Port herstellen und auf die Serveranwendung zugreifen, die in der Instance ausgeführt wird. Weitere Informationen finden Sie unter den folgenden Themen: • Port-Weiterleitung mit AWS Systems ManagerSession Manager im AWS News Blog • Starten einer Sitzung (Port-Weiterleitung)	29. August 2019

Festlegen einer Standardparameterebene oder Automatisieren der Stufefenauswahl	Sie können jetzt eine Standardparameterebene angeben, die für Anforderungen zum Erstellen oder Aktualisieren eines Parameters verwendet werden soll, die keine Stufe angeben. Sie können die Standardstufe auf Standardparameter, erweiterte Parameter oder eine neue Option, Intelligent-Tiering, festlegen. Intelligent-Tiering bewertet jede PutParameter Anfrage und erstellt nur bei Bedarf einen erweiterten Parameter. (Erweiterte Parameter sind erforderlich, wenn die Größe des Parameterwerts mehr als 4 KB beträgt, dem Parameter eine Parameterrichtlinie zugeordnet ist oder die maximal 10 000 für die Standardstufe unterstützten Parameter bereits erstellt wurden.) Weitere Informationen zum Angeben einer Standardstufe und zur Verwendung von Intelligent-Tiering finden Sie unter Angeben einer Standardparameterstufe.	27. August 2019

Der Abschnitt „Arbeiten mit Assoziationen“ wurde mit CLI und PowerShell Verfahren aktualisiert	Der Abschnitt „Mit Assoziationen arbeiten“ wurde aktualisiert und enthält nun eine Verfahrensdokumentation für die Verwaltung von Verknüpfungen mithilfe von AWS CLI oder AWS -Tools für PowerShell. Weitere Informationen finden Sie unter Arbeiten mit Zuordnungen in Systems Manager.	26. August 2019
Der Abschnitt „Arbeiten mit Automatisierungsausführungen“ wurde mit CLI und PowerShell Verfahren aktualisiert	Der Abschnitt „Mit Automatisierungsausführungen arbeiten“ wurde aktualisiert und enthält nun eine prozedurale Dokumentation für die Ausführung von Automatisierungsworkflows mit dem AWS CLI oder. AWS -Tools für PowerShell Weitere Informationen finden Sie unter Arbeiten mit Automation-Ausführungen.	20. August 2019

OpsCenter lässt sich in Application Insights integrieren	OpsCenterlässt sich in Amazon CloudWatch Application Insights für .NET und SQL Server integrieren. Das bedeutet, dass Sie automatisch OpsItems für Probleme erstellen können, die in Ihren Anwendungen erkannt wurden. Informationen zur Konfiguration von Application Insights zum Erstellen OpsItems finden Sie unter Einrichten, Konfigurieren und Verwalten Ihrer Anwendung für die Überwachung im CloudWatch Amazon-Benutzerhandbuch.	7. August 2019

Neue Konsolenfunktion: AWS Systems ManagerQuick Setup	Quick Setupist eine neue Funktion in der Systems Manager Manager-Konsole, mit der Sie schnell mehrere Systems Manager Manager-Komponenten auf Ihren EC2 Instanzen konfigurieren können. Mit Quick Setup können Sie insbesondere die folgenden Komponenten auf den Instances konfiguri eren, die Sie mithilfe von Tags auswählen oder als Ziel festlegen: • Eine AWS Identity and Access Management (IAM-) Instanzprofilrolle für Systems Manager. • Eine geplante, zweimonat liche Aktualisierung von SSM Agent. • Eine geplante Sammlung von Inventory-Metadaten alle 30 Minuten. • Ein täglicher Scan Ihrer Instances, um fehlende Patches zu identifizieren. • Eine einmalige Installat ion und Konfiguration des CloudWatch Amazon-Ag enten. • Ein geplantes, monatlich es Update des CloudWatch Agenten.	7. August 2019

Weitere Informationen finden Sie unter AWS Systems Manager Quick Setup.

Registrieren einer Ressourcengruppe als Ziel eines Wartungsfensters	Zusätzlich zur Registrierung verwalteter Instanzen als Ziel eines Wartungsfensters können Sie jetzt auch eine Ressourcengruppe als Ziel für ein Wartungsfenster registrieren. Maintenance Windowsunterstützt alle AWS Ressourcentypen, die unter AWS -Ressourcengruppen anderem`AWS::EC2::Instance` ,, `AWS::DynamoDB::Table` `AWS::OpsWorks::Instance` `AWS::Redshift::Cluster` , und mehr unterstützt werden. Mit dieser Version können Sie auch Befehle an eine Ressourcengruppe senden, z. B. mithilfe der Run Command Konsole oder des AWS CLI send-commandBefehls. Weitere Informationen finden Sie unter den folgenden Themen: • Ziele zu einem Wartungsfenster zuweisen (Konsole) • Beispiele: Ziele für ein Wartungsfenster registrieren • Verwenden von Zielen und Häufigkeitskontrollen zum Senden von Befehlen an eine Gruppe von Instances	23. Juli 2019

Vereinfachte Paketerstellung und -Versioning mit AWS Systems Manager Distributor	Distributor verfügt über eine neuen, vereinfachten Workflow zur Paketerstellung, der für ein Paket das Manifest, Skripts und Datei-Hashes generiere n kann. Sie können auch den vereinfachte Workflow nutzen, wenn Sie einem bereits vorhandenen Paket eine Version hinzufügen.	22. Juli 2019
Bereich „New Document categories (Neue Dokumentk ategorien)“ für Systems Manager Automation	Systems Manager umfasst einen Bereich „Neue Dokumentkategorien“, der angezeigt wird, wenn Sie in der Konsole eine Automatis ierung ausführen. Verwenden Sie diesen Bereich zum Filtern der Automatisierungs-R unbooks nach Zweck.	18. Juli 2019

Unterstützung für das Starten von Session Manager-Sitzungen mit Betriebssystem-Anmeldeinformationen	Standardmäßig werden Session Manager-Sitzungen unter Verwendung der Anmeldeinformationen eines vom System generierten `ssm-user`-Kontos gestartet werden, der auf einer verwalteten Instance erstellt wird. Auf Linux-Computern können Sie jetzt stattdessen Sitzungen unter Verwendung der Anmeldeinformationen für ein Betriebssystemkonto starten. Weitere Informationen finden Sie unter Aktivieren der Run As-Unterstützung für Linux-Instances.	9. Juli 2019
Unterstützung für das Starten von Session Manager-Sitzungen mit SSH	Sie können den jetzt verwenden AWS CLI , um eine SSH-Sitzung auf einer verwalteten Instanz zu starten, indem SieSession Manager. Weitere Informationen zur Aktivierung von SSH-Sitzungen mit Session Manager finden Sie unter (Optional) Aktivieren von SSH-Session Manager-Sitzungen. Weitere Informationen zum Starten einer SSH-Sitzung mit Session Manager finden Sie unter Starten einer Sitzung (SSH).	9. Juli 2019

Unterstützung für das Ändern von Passwörtern auf verwalteten Instances	Sie können jetzt Passwörter auf Computern zurücksetzen, die Sie mit Systems Manager verwalten (verwaltete Instances). Sie können das Passwort über die Systems Manager-Konsole oder die AWS CLI zurücksetzen. Weitere Informationen finden Sie unter Zurücksetzen von Passwörtern auf verwalteten Instances.	9. Juli 2019
Änderungen von „Was ist AWS Systems Manager?“	Die einführenden Inhalte in Was ist AWS Systems Manager? wurden erweitert, um eine breitere Einführung in den Service zu bieten und kürzlich veröffentlichten Systems-Manager-Tools Rechnung zu tragen. Darüber hinaus wurden andere Inhalte in einzelne Themen verschoben, sodass sie leichter auffindbar sind.	10. Juni 2019

Neues Systems Manager-Tool: OpsCenter	OpsCenterbietet einen zentralen Ort, an dem Betriebsingenieure und IT-Experten betriebliche Arbeitsaufgaben (OpsItems) im Zusammenhang mit AWS Ressourcen einsehen, untersuchen und lösen können. OpsCenterwurde entwickelt, um die durchschnittliche Zeit bis zur Lösung von Problemen zu reduzieren, die sich auf AWS Ressourcen auswirken. Dieses Systems Manager-Tool aggregiert und standardisiert OpsItems über Services hinweg und bietet gleichzeitig kontextbezogene Untersuchungsdaten über jedes OpsItem, verwandte OpsItems und verwandte Ressourcen. OpsCenter bietet außerdem Systems-Manager-Automatisierungsdokumente (Runbooks), die Sie verwenden können, um Probleme schnell zu lösen. Sie können durchsuchbare, benutzerdefinierte Daten für jedes OpsItem angeben. Sie können auch automatisch generierte Zusammenfassungsberichte über OpsItems nach Status und Quelle anzeigen. Weitere Informationen finden Sie unter	6. Juni 2019

	AWS Systems ManagerOpsCenter.	
Änderungen am linken Navigationsbereich von Systems Manager in der AWS-Managementkonsole	Der linke Navigationsbereich von Systems Manager AWS-Managementkonsole enthält neue Überschriften, darunter eine neue Überschrift für Ops Center, die eine logischere Gruppierung der Systems Manager Manager-Tools ermöglichen.	6. Juni 2019

Überarbeitete Anleitung zum Erstellen und Konfigurieren eines Wartungsfensters mithilfe der AWS CLI	Tutorial: Erstellen und Konfigurieren eines Wartungsfensters (AWS CLI) wurde überarbeitet, um den Pfad durch die Übungsschritte zu vereinfachen. Sie erstellen ein einziges Wartungsfenster, identifizieren ein einziges Ziel und richten eine einfache Aufgabe ein, die im Wartungsfenster ausgeführt werden soll. Dabei stellen wir Informationen und Beispiele zur Verfügung, mit denen Sie eigene Befehle zur Aufgabenregistrierung erstellen können. Dazu zählen auch Informationen zur Verwendung von Pseudoparametern wie `{{TARGET_ID}}` . Zusätzliche Informationen und Beispiele finden Sie in den folgenden Themen: • Beispiele: Ziele für ein Wartungsfenster registrieren • Beispiele: Registrieren von Aufgaben für ein Wartungsfenster • Informationen zu den Optionen von register-task-with-maintenance -windows • Verwenden von Pseudo-Parametern bei der Registrierung von Aufgaben im Wartungsfenster	31. Mai 2019

Benachrichtigungen über SSM Agent-Updates	Wenn Sie über SSM Agent-Aktualisierungen benachrichtigt werden möchten, abonnieren Sie die Seite SSM Agent-Versionshinweise auf GitHub.	24. Mai 2019
Erhalten von Benachrichtigungen oder Auslösen von Aktionen basierend auf Änderungen in Parameter Store	Das Thema Benachrichtigungen einrichten oder Aktionen auf der Grundlage von Parameter Store Ereignissen auslösen hilft Ihnen jetzt dabei, EventBridge Amazon-Regeln einzurichten, um auf Änderungen in zu reagierenParameter Store. Sie können Benachrichtigungen erhalten oder andere Aktionen auslösen, wenn eines der folgenden Ereignisse eintritt: • Ein Parameter wird erstellt, aktualisiert oder gelöscht. • Eine Parameterbezeichnungsversion wird erstellt, aktualisiert oder gelöscht. • Ein Parameter läuft ab, läuft in Kürze ab oder wurde in einem angegebenen Zeitraum nicht geändert.	22. Mai 2019

Hauptrevidierungen an Inhalten zu Einrichtung oder ersten Schritten	Wir haben die Inhalte zu Einrichtung und Erste Schritte im AWS Systems Manager - Benutzerhandbuch erweitert und neu organisiert. Die Inhalte zur Einrichtung wurden in zwei Abschnitte unterteilt. Ein Abschnitt konzentriert sich auf Aufgaben zur Einrichtung von Systems Manager zur Konfiguration und Verwaltung Ihrer EC2 Instanzen. Der andere konzentriert sich auf Aufgaben zur Einrichtung von Systems Manager zur Konfiguration und Verwaltung Ihrer lokalen Server und virtuellen Maschinen (VMs) in einer Hybridumgebung. Beide Abschnitte präsentieren jetzt alle Einrichtungsthemen als wesentliche, nummerierte Schritte, in der empfohlene Reihenfolge der Durchführung. Ein neues Kapitel Erste Schritte ist darauf ausgelegt, Endbenutzern bei den ersten Schritte mit Systems Manager zu helfen, nachdem Konto- und Service-Konfigurationsaufgaben abgeschlossen sind. • Einrichtung AWS Systems Manager	15. Mai 2019

- Einrichtung AWS Systems Manager für Hybridumgebungen
- Erste Schritte mit AWS Systems Manager

Patches für von Microsoft veröffentlichte Anwendungen jetzt in Patch-Baselines (Windows) enthalten	Patch Manager unterstützt jetzt Patch-Updates für von Microsoft veröffentlichte Anwendungen auf Windows Server-Instances. Bisher wurden nur Patches für das Windows Server-Betriebssystem unterstützt. Patch Manager bietet zwei vordefinierte Patch-Baselines für Windows Server-Instances. Die Patch-Baseline `AWS-WindowsPredefinedPatchBaseline-OS` gilt nur für Betriebssystem-Patches. `AWS-WindowsPredefinedPatchBaseline-OS-Applications` gilt sowohl für das Windows Server-Betriebssystem als auch für von Microsoft veröffentlichte Anwendungen unter Windows. Weitere Informationen zum Erstellen einer benutzerdefinierten Patch-Baseline, die Patches für von Microsoft veröffentlichte Anwendungen enthält, finden Sie im ersten Verfahren unter Erstellen einer benutzerdefinierten Patch-Baseline. Im Rahmen dieses Updates werden auch die Namen der AWS bereitgestellten vordefinierten Patch-Baselines geändert. Weitere Informati	7. Mai 2019

	onen finden Sie unter Vordefinierte Baselines.	
Beispiele für die Registrierung von Wartungsfensterzielen mit dem AWS CLI	Das neue Thema über Beispiele: Registrieren von Zielen für ein Wartungsfenster bietet drei Beispielbefehle, um verschiedene Möglichkeiten zu zeigen, wie Sie die Ziele für ein Wartungsfenster angeben können, wenn Sie die AWS CLI verwenden. Darüber hinaus wird in diesem Thema der beste Anwendungsfall für jeden der Beispielbefehle erläutert.	3. Mai 2019

Updates für Themen für Patch-Gruppen	Das Thema Patch-Gruppen wurde aktualisiert und enthält nun einen Abschnitt darüber, wie verwaltete Instances die entsprechende Patch-Baseline bestimmen, die während der Patch-Vorgänge zu verwenden ist. Darüber hinaus wurden Anweisungen hinzugefügt, wie Sie mithilfe der AWS CLI oder der Systems Manager Manager-Konsole Patchgrup pen oder PatchGroupTags zu Ihren verwalteten Instances hinzufügen können und wie Sie eine Patchgruppe oder PatchGroupeine Patch-Baseline hinzufügen. (Sie **`PatchGroup`** müssen ohne Leerzeichen angeben, ob Sie Tags in EC2 Instanz-M etadaten zugelassen haben.) Weitere Informationen finden Sie unter Erstellen einer Patch-Gruppe und Hinzufüge n einer Patch-Gruppe zu einer Patch-Baseline.	1. Mai 2019

Neue Parameter Store-Funktionen	Parameter Store bietet die folgenden neuen Features: • Erweiterte Parameter : Mit Parameter Store können Sie jetzt Parameter einzeln konfigurieren, um entweder ein Standardparameterkontingent (das Standardkontingent) oder ein Kontingent für erweiterte Parameter zu verwenden. Erweiterte Parameter bieten ein größeres Größenkontingent für den Parameterwert, ein höheres Kontingent für die Anzahl der Parameter, die Sie pro AWS-Konto und erstellen können AWS-Region, und die Möglichkeit, Parameterrichtlinien zu verwenden. Weitere Informationen über erweiterte Parameter finden Sie unter Über erweiterte Parameter von Systems Manager. • Parameterrichtlinien: Parameterrichtlinien unterstützen Sie bei der Verwaltung einer wachsenden Menge von Parametern, indem Sie einem Parameter bestimmte Kriterien zuweisen können, wie etwa Ablaufdatum oder	25. April 2019

Gültigkeitsdauer. Parameter richtlinien sind besonders hilfreich, da sie Sie zwingen, in Parameter Store gespeicherte Passwörte r und Konfigurationsdate n zu aktualisieren oder zu löschen. Parameter richtlinien sind nur verfügbar für Parameter, die das Kontingent für erweitert e Parameter verwenden . Weitere Informationen finden Sie im Artikel zum Arbeiten mit Parameter richtlinien.

- Höherer Durchsatz: Sie können jetzt das Durchsatz kontingent von Parameter Store auf ein Maximum von 1 000 Transaktionen pro Sekunde erhöhen. Weitere Informationen finden Sie im Artikel zum Erhöhen des Parameter Store-Dur chsatzes.

Aktualisierungen des Abschnitts über Automatisierungen	Der Abschnitt über Automatisierungen wurde aktualisiert und ist nun übersichtlicher. Darüber hinaus wurden dem Abschnitt drei neue Themen hinzugefügt: • Ausführen einer Automatisierung Schritt für Schritt • Eine Automatisierung ausführen, für die Genehmigungen erforderlich sind • Planen von Automatisierungen mit State Manager-Zuordnungen	17. April 2019

Verschlüsseln Sie Sitzungsdaten mit einem Schlüssel AWS KMS	Session ManagerVerwendet standardmäßig TLS 1.2, um Sitzungsdaten zu verschlüsseln, die zwischen den lokalen Computern der Benutzer in Ihrem Konto und Ihren EC2 Instanzen übertragen werden. Jetzt können Sie wählen, ob Sie diese Daten mit einem, das in erstellt wurde AWS KMS key , weiter verschlüsseln möchten. AWS Key Management Service Sie können einen KMS-Schlüssel verwenden, der in Ihrem AWS-Konto erstellt wurde, oder einen Schlüssel , der von einem anderen Konto für Sie freigegeben wurde. Informationen zum Angeben eines KMS-Schlüssels zum Verschlüsseln von Sitzungsdaten finden Sie unter Aktivieren der AWS KMS Schlüsselverschlüsselung von Sitzungsdaten (Konsole) , Session ManagerEinstellungen erstellen (AWS CLI) oder Session ManagerEinstellungen aktualisieren (AWS CLI).	4. April 2019

Konfiguration von Amazon SNS SNS-Benachrichtigungen für AWS Systems Manager	Es wurden Anweisungen zur Verwendung der AWS CLI oder Systems Manager Manager-Konsole zur Konfiguration von Amazon SNS SNS-Benachrichtigungen Run Command und Run Command Aufgaben hinzugefügt, die für ein Wartungsfenster registriert sind. Weitere Informationen finden Sie unter Konfigurieren von Amazon SNS-Benachrichtigungen für AWS Systems Manager.	6. März 2019

Erweiterte Instances für Server und VMs in Hybridumgebungen	AWS Systems Manager bietet eine Stufe „Standard-Instances“ und eine Stufe „Advanced-Instances“ für Server und VMs in Ihrer Hybridumgebung. Mit der Stufe „Standard-Instances“ können Sie maximal 1.000 Server oder pro pro Server registrieren. VMs AWS-Konto AWS-Region Wenn Sie mehr als 1.000 Server oder VMs nur ein Konto und eine Region registrieren müssen, verwenden Sie die Stufe „Advanced-Instances“. In der Advanced-Instance-Stufe können Sie beliebig viele Instanzen erstellen, aber alle für Systems Manager konfigurierten Instanzen sind auf einer pay-per-use Basis verfügbar. Mit erweiterten Instanzen können Sie auch eine Verbindung zu Ihren Hybridmaschinen herstellen, indem Sie. AWS Systems Manager Session Manager Session Managerbietet interaktiven Shell-Zugriff auf Ihre Instanzen. Weitere Informationen zum Aktivieren von erweiterten Instances finden Sie im Artikel zum Verwenden des Kontingents für erweiterte Instances.	4. März 2019

Erstellen von State Manager-Zuordnungen, die freigegebene SSM-Dokumente verwenden	Sie können State Manager Verknüpfungen erstellen, die SSM-Command and Automation-Runbooks verwenden, die von anderen gemeinsam genutzt werden. AWS-Konten Das Erstellen von Verknüpfungen mithilfe gemeinsam genutzter SSM-Dokumente trägt dazu bei, dass Ihre Amazon EC2 - und Hybrid-Infrastruktur in einem konsistenten Zustand bleibt, auch wenn sich Instances nicht im selben Konto befinden. Weitere Informationen zur Freigabe von SSM-Dokumenten finden Sie unter AWS Systems Manager -Dokumente. Weitere Informationen zum Erstellen einer State Manager-Zuordnung finden Sie unter Erstellen einer Zuordnung.	28. Februar 2019
Listen der Systems Manager Manager-Ereignisse anzeigen, die für EventBridge Amazon-Regeln unterstützt werden	Das neue Thema Überwachung von Systems Manager-Ereignissen mit Amazon EventBridge bietet eine Zusammenfassung der verschiedenen von Systems Manager ausgegebenen Ereignisse, für die Sie Regeln zur Ereignisüberwachung einrichten können EventBridge.	25. Februar 2019

Hinzufügen von Tags beim Erstellen von Systems Manager-Ressourcen	Systems Manager unterstützt jetzt die Möglichkeit, bestimmten Ressourcentypen Tags hinzuzufügen, wenn Sie sie erstellen. Zu den Ressourcen, die Sie taggen können, wenn Sie sie mit dem AWS CLI oder einem SDK erstellen, gehören Wartungsfenster, Patch-Baselines, Parameter Store Parameter und SSM-Dokumente. Sie können auch einer verwalteten Instance Tags zuweisen, wenn Sie eine Aktivierung dafür erstellen. Wenn Sie die Systems Manager-Konsole verwenden, können Sie Wartungsfenstern, Patch-Baselines und Parametern Tags hinzufügen.	24. Februar 2019

Automatische IAM-Rollenerstellung für Systems Manager-Inventory	Bisher mussten Sie eine AWS Identity and Access Management (IAM-) Rolle erstellen und dieser Rolle separate Richtlinien zuordnen, um Inventardaten auf der Seite „Inventardetailansicht“ in der Konsole anzuzeigen. Es ist nicht mehr erforderlich, diese Rolle zu erstellen oder ihr Richtlinien zuzuweisen. Wenn Sie auf der Seite „Inventardetailansicht“ eine Remote-Datensynchronisierung wählen, erstellt Systems Manager automatisch die `Amazon-GlueServicePolicyForSSM` Rolle und weist ihr die Amazon- GlueServicePolicyFor SSM- {S3 bucket name} -Richtlinie und die AWSGlueServiceRoleRichtlinie zu. Weitere Informationen finden Sie unter Abfragen von Bestandsdaten aus mehreren Regionen und Konten.	14. Februar 2019

Maintenance Windows-Anleitungen zum Aktualisieren von SSM Agent	Der Maintenance Windows-Dokumentation wurden zwei neue Anleitungen hinzugefügt. In den exemplarischen Vorgehensweisen wird detailliert beschrieben, wie Sie die Systems Manager Manager-Konsole verwenden oder ein Wartungsfenster erstellen, das automatisch beibehalten wird SSM Agent up-to-date. AWS CLI Weitere Informationen finden Sie im Artikel über Maintenance Windows-Anleitungen.	11. Februar 2019
Verwenden von öffentlichen Parameter Store-Parametern	Ein kurzer Abschnitt mit einer Beschreibung der öffentlichen Parameter Store-Parameter wurde hinzugefügt. Weitere Informationen finden Sie unter Verwenden von öffentlichen Systems Manager-Parametern .	31. Januar 2019

Verwenden Sie die, AWS CLI um Einstellungen zu erstellen Session Manager	Es wurden Anweisungen zur Verwendung von hinzugefügt, AWS CLI um Session Manager Einstellungen wie CloudWatch Protokolle, S3-Bucket-Protokollierungsoptionen und Einstellungen für die Sitzungsverschlüsselung zu erstellen. Weitere Informationen finden Sie unter Verwenden von, AWS CLI um Session Manager Einstellungen zu erstellen.	22. Januar 2019
Ausführen von Systems Manager-Automatisierungsworkflows mithilfe von State Manager	AWS Systems Manager State Managerunterstützt jetzt das Erstellen von Zuordnungen, die SSM Automation-Runbooks verwenden. State Managerbisher wurden nur `command` `policy` Dokumente unterstützt, was bedeutete, dass Sie nur Verknüpfungen erstellen konnten, die auf verwaltete Instanzen abzielten. Mit Unterstützung für SSM Automation-Runbooks können Sie jetzt Zuordnungen erstellen, die unterschiedliche Arten von AWS -Ressourcen zum Ziel haben. Weitere Informationen finden Sie im Artikel über das Ausführen von Systems Manager Automation-Workflows mithilfe von State Manager.	22. Januar 2019

Referenz-Updates für Cron- und Rate-Ausdrücke und Planungsoptionen für Wartungsfenster	Das Referenzthema Cron- und Rate-Ausdrücke für Systems Manager wurde überarbeitet. Die neue Version stellt weitere Beispiele und verbesserte Erläuterungen zur Verwendung von Cron- und Rate-Ausdrücken zum Planen Ihrer Wartungsfenster und State Manager-Zuordnungen bereit. Darüber hinaus wird im neuen Thema Maintenance Windows-Optionen für Planung und aktive Zeiträume erläutert, wie die verschiedenen Zeitplanoptionen für Wartungsfenster (Startdatum, Enddatum, Zeitzone, Zeitplanhäufigkeit) miteinander in Verbindung stehen.	6. Dezember 2018
Aktivieren von SSM Agent-Debugging-Protokollierung	Sie können die SSM Agent-Debug-Protokollierung aktivieren, indem Sie die Datei seelog.xml.template auf der verwalteten Instance bearbeiten. Weitere Informationen finden Sie unter Aktivieren der SSM Agent-Debug-Protokollierung.	30. November 2018

Support für ARM64 Prozessorarchitekturen	AWS Systems Manager unterstützt jetzt ARM64 Versionen der Betriebssysteme Amazon Linux 2, Red Hat Enterprise Linux 7.6 und Ubuntu Server (18.04 LTS und 16.04 LTS). Weitere Informationen finden Sie in den Anweisungen zum Installieren von Amazon Linux 2, RHEL und Ubuntu Server 18.04 und 16.04 LTS mit Snap-Paketen. Weitere Informationen zum Instance-Typ A1 finden Sie unter General Purpose Instances im EC2 Amazon-Benutzerhandbuch.	26. November 2018

Erstellen und Bereitstellen mit AWS Systems Manager Distributor	Mithilfe von AWS Systems Manager Distributor Paketen Sie Ihre eigene Software — oder suchen nach AWS bereitgestellten Agenten-Softwarepaketen, z. B. `AmazonCloudWatchAgent` —, um sie auf verwalteten Instances zu installieren. AWS Systems Manager Distributorveröffentlicht Ressourcen, z. B. Softwarepakete, auf verwalteten Instanzen. AWS Systems Manager Beim Veröffentlichen eines Pakets werden bestimmte Versionen des Paketdokuments — eines Systems Manager Manager-Dokuments, das Sie erstellen, wenn Sie das Paket hinzufügen Distributor — für verwaltete Instanzen angekündigt, die Sie anhand einer verwalteten Instanz IDs AWS-Konto IDs, Tags oder einer identifizieren. AWS-Region Weitere Informationen finden Sie unter AWS Systems ManagerDistributor.	20. November 2018

Führen Sie AWS Systems Manager Automatisierungs-Workflows gleichzeitig für mehrere Benutzer AWS-Regionen und AWS-Konten von einem zentralen Konto aus aus	Sie können AWS Systems Manager Automatisierungsworkflows gleichzeitig über mehrere AWS-Regionen AWS-Konten und/oder AWS Organisationseinheiten (OUs) von einem Automatisierungsverwaltungskonto aus ausführen. Die gleichzeitige Ausführung von Automatisierungen in mehreren Regionen und Konten oder OUs reduziert den Zeitaufwand für die Verwaltung Ihrer AWS Ressourcen und erhöht gleichzeitig die Sicherheit Ihrer Computerumgebung. Weitere Informationen finden Sie unter Ausführen von Automatisierungsworkflows in mehreren und. AWS-Regionen AWS-Konten	19. November 2018

Inventardaten von mehreren abfragen AWS-Regionen und AWS-Konten	Systems Manager Inventory ist in Amazon Athena integriert, sodass Sie Inventardaten von mehreren AWS-Regionen und AWS-Konten abfragen können. Die Athena-Integration verwendet die Ressourcendatensynchronisierung, sodass Sie Inventardaten von all Ihren verwalteten Instanzen auf der Seite „Inventardetailansicht“ in der AWS Systems Manager Konsole anzeigen können. Weitere Informationen finden Sie unter Abfragen von Bestandsdaten aus mehreren Regionen und Konten.	15. November 2018

Erstellen von State Manager-Zuordnungen, die MOF-Dateien ausführen	Sie können Managed Object Format (MOF)-Dateien ausführen, um einen gewünschten Status auf von Windows Server verwalteten Instances mit State Manager mithilfe des SSM-Dokuments `AWS-ApplyDSCMofs` zu erzwingen. Das `AWS-ApplyDSCMofs` -Dokument weist zwei Ausführungsmodi auf. Mit dem ersten Modus können Sie die Zuordnung so konfigurieren, dass sie scannt und meldet, wenn sich die verwalteten Instances derzeit in dem Zielstatus befinden, der in den angegebenen MOF-Dateien definiert ist. Im zweiten Modus können Sie die MOF-Dateien ausführen und die Konfiguration Ihrer Instances basierend auf den Ressourcen und ihren in den MOF-Dateien definierten Werten ändern. Mit dem `AWS-ApplyDSCMofs` -Dokument können Sie MOF-Konfigurationsdateien von Amazon Simple Storage Service (Amazon S3), einem lokal freigegebenen Verzeichnis, oder einer sicheren Website mit einer HTTPS-Domain herunterladen und ausführen. Weitere Informationen	15. November 2018

	finden Sie unter Erstellen von Zuordnungen, die MOF-Dateien ausführen.	
Einschränken des administrativen Zugriff in Session Manager-Sitzungen	Session Manager-Sitzungen werden unter Verwendung der Anmeldeinformationen für ein Benutzerkonto namens `ssm-user` gestartet, das mit dem Root- oder Administrator-Berechtigungen erstellt wurde. Informationen zum Einschränken der administrativen Kontrolle für dieses Konto sind jetzt im Thema Deaktivieren oder Aktivieren von administrativen Berechtigungen für das Konto ssm-user verfügbar.	13. November 2018
YAML-Beispiele in Automatisierungsaktionsreferenz	Die Automatisierungsaktionsreferenz enthält jetzt ein YAML-Beispiel für die einzelnen Aktionen, die bereits ein JSON-Beispiel enthalten.	31. Oktober 2018

Zuweisen von Compliance-Schweregraden zu Assoziationen	Sie können State Manager-Zuordnungen jetzt Compliance-Schweregrade zuweisen. Diese Schweregrade werden im Compliance-Dashboard gemeldet und können auch zum Filtern Ihrer Compliance-Berichte verwendet werden. Die Schweregrade, die Sie zuweisen können, sind Kritisch, Hoch, Mittel, Niedrig und Nicht angegeben. Weitere Informationen finden Sie unter Erstellen einer Zuordnung (Konsole).	26. Oktober 2018
Verwenden von Ziel- und Ratensteuerungen mit Automatisierung und State Manager	Steuern Sie die Ausführung von Automatisierungen und State Manager-Zuordnungen innerhalb Ihrer Flotte von Ressourcen mithilfe von Zielen, Gleichzeitigkeit und Fehlerschwellenwerten. Weitere Informationen finden Sie unter Verwenden von Ziel- und Ratensteuerungen für die Ausführung von Automatisierungsworkflows zu einer Flotte und Verwenden von Zielen und Ratensteuerungen bei State Manager-Zuordnungen.	23. Oktober 2018

Festlegen von aktiven Zeitbereichen und internationalen Zeitzonen für Wartungsfenster	Sie können auch Daten festlegen, vor oder nach denen Wartungsfenster nicht ausgeführt werden sollte (Start- und Enddatum), und Sie können die internationale Zeitzone als Grundlage für den Wartungsfenster-Zeitplan festlegen. Weitere Informationen finden Sie unter Erstellen eines Wartungsfensters (Konsole) und Aktualisieren eines Wartungsfensters (AWS CLI).	9. Oktober 2018
Führen einer benutzerdefinierten Liste mit Patches für Ihre Patch-Baseline in einem S3-Bucket	Mit dem neuen Parameter 'InstallOverrideList' im SSM-Befehlsdokument `AWS-RunPatchBaseline` können Sie eine HTTPS-URL oder eine URL im Pfadstil von Amazon Simple Storage Service (Amazon S3) für eine Liste von zu installierenden Patches angeben. Diese in einem S3-Bucket im YAML-Format geführte Patch-Installationsliste überschreibt die von der Standard-Patch-Baseline angegebenen Patches. Weitere Informationen finden Sie unter Parametername:. InstallOverrideList	5. Oktober 2018

Erweiterte Kontrolle über die Installation von Patch-Abhängigkeiten	Wenn ein Patch zuvor in Ihrer Liste für abgelehnte Patches als Abhängigkeit eines anderen Patches identifiziert wurde, wäre es trotzdem installiert worden. Jetzt können Sie wählen, ob Sie diese Abhängigkeiten installieren möchten oder nicht. Weitere Informationen finden Sie unter Erstellen einer Patch-Baseline.	5. Oktober 2018
Erstellen dynamischer Automatisierungsworkflows mit bedingten Verzweigungen	Die `aws:branch` -Automatisierungsaktion ermöglicht das Erstellen eines dynamischen Automatisierungsworkflows, der verschiedene Auswahlmöglichkeiten in einem einzigen Schritt evaluiert und dann auf der Grundlage dieser Evaluierung zu einem anderen Schritt in dem Automatisierungs-Runbook springt. Weitere Informationen finden Sie unter Verwendung bedingter Anweisungen in Runbooks.	26. September 2018

Verwenden Sie die AWS CLI , um die Session Manager Einstellungen zu aktualisieren	Anweisungen zur Verwendung der CLI zur Aktualisierung von Session Manager Einstellungen wie CloudWatch Logs und S3-Bucket-Logging-Optionen wurden dem AWS Systems Manager Benutzerhandbuch hinzugefügt. Weitere Informationen finden Sie unter Verwenden von, AWS CLI um Session Manager Einstellungen zu aktualisieren.	25. September 2018
Aktualisierte SSM Agent-Voraussetzung für Session Manager	Session Manager erfordert jetzt SSM Agent Version 2.3.68.0 oder höher. Weitere Informationen zu Session Manager-Voraussetzungen finden Sie unter Abschließen der Session Manager-Voraussetzungen.	17. September 2018

Verwalten von Instances, ohne eingehende Ports öffnen oder Bastion-Hosts mithilfe von Session Manager pflegen zu müssen	Mithilfe Session Manager eines vollständig verwaltet en Tools in AWS Systems Manager können Sie Ihre EC2 Instanzen über eine interaktive browserbasierte Shell mit einem Klick oder über das verwalten. AWS CLISession Managerbietet eine sichere und überprüfbare Instanzverwaltung, ohne dass eingehende Ports geöffnet, Bastion-Hosts verwaltet oder SSH-Schlüssel verwaltet werden müssen. Session Managerermöglicht es Ihnen außerdem, Unternehm ensrichtlinien einzuhalten, die einen kontrollierten Zugriff auf Instances, strenge Sicherhei tspraktiken und vollständ ig überprüfbare Protokoll e mit Instanzzugriffsdetails vorschreiben, und gleichzeitig Endbenutzern einen einfachen plattformübergreifenden Zugriff mit nur einem Klick auf Ihre Instances gewähren. EC2 Weitere Informationen finden Sie unter Weitere Informati onen zu Session Manager.	11. September 2018

Andere AWS-Services aus einem Systems Manager Automation-Workflow aufrufen	Sie können andere AWS-Services und andere Systems Manager Manager-Tools in Ihrem Automatisierungs-Workflow aufrufen, indem Sie drei neue Automatisierungsaktionen (oder Plug-ins) in Ihren Automations-Runbooks verwenden. Weitere Informationen finden Sie unter Verwenden von Aktionsausgaben als Eingaben.	28. August 2018
Verwenden von Systems Manager-spezifischen Bedingungsschlüsseln in IAM-Richtlinien	Das Thema Angeben von Bedingungen in einer Richtlinie wurde um eine Liste der IAM-Bedingungsschlüssel für Systems Manager ergänzt, die Sie in Richtlinien integrieren können. Verwenden Sie diese Schlüssel zum Angeben der Bedingungen, unter denen eine Richtlinie wirksam werden soll. Das Thema enthält auch Links zu Beispielrichtlinien und anderen verwandten Themen.	18. August 2018

Aggregieren von Bestandsdaten mit Gruppen, um zu sehen, welche Instances zur Erfassung eines Bestandstyps konfiguriert sind und welche nicht	Gruppen ermöglichen es Ihnen, schnell eine Anzahl der verwalteten Instances zu sehen, die für das Erfassen eines oder mehrerer Bestandstypen konfiguriert sind bzw. nicht konfiguriert sind. Mit Gruppen geben Sie einen oder mehrere Inventory-Typen sowie einen Filter an, der den `exists`-Operator verwendet. Weitere Informationen finden Sie unter Aggregieren von Bestandsdaten.	16. August 2018
Anzeigen von Verlauf Änderungsnachverfolgung für Inventory und Configuration Compliance	Sie können den Verlauf und die Änderungsnachverfolgung für von Ihren verwalteten Instances erfasstes Inventory anzeigen. Sie können auch den Verlauf und die Änderungsnachverfolgung für Patch Manager-Patch-Einspielungen und State Manager-Zuordnungen, die von Configuration Compliance gemeldet werden, ansehen. Weitere Informationen finden Sie unter Anzeigen von Inventory-Verlauf und Änderungsnachverfolgung.	9. August 2018

Parameter Store integriert sich mit Secrets Manager	Parameter Storeist jetzt integriert, AWS Secrets Manager sodass Sie Secrets Manager abrufen können, wenn Sie andere verwenden AWS-Services , die bereits Verweise auf Parameter Store Parameter unterstützen. Zu diesen Services gehören Amazon EC2, Amazon Elastic Container Service, AWS Lambda, CloudForm ation, AWS CodeBuild AWS CodeDeploy, und andere Systems Manager Manager-Tools. Indem Sie Parameter Store zum Verweisen auf Secrets Manager-Geheimniss e verwenden, erstellen Sie einen konsistenten und sicheren Prozess zum Aufrufen und Verwenden von Geheimnissen und zum Referenzieren von Daten in Ihrem Code und den Konfigurationsskripts. Weitere Informationen finden Sie unter Verweisen auf AWS Secrets Manager Geheimnisse anhand von Parameter Store Parametern.	26. Juli 2018

Anfügen von Bezeichnungen zu Parameter Store-Parametern	Eine Parameter-Bezeichnung ist ein benutzerdefinierter Alias, mit dem Sie verschiedene Versionen eines Parameters verwalten können. Wenn Sie einen Parameter ändern, speichert Systems Manager automatisch eine neue Version und erhöht die Versionsnummer um 1. Dank einer Bezeichnung können Sie sich den Zweck einer Parameterversion merken, wenn mehrere Versionen vorhanden sind. Weitere Informationen finden Sie unter Bezeichnen von Parametern.	26. Juli 2018

Erstellen dynamischer Automatisierungsworkflows	Standardmäßig werden die Schritte (oder Aktionen), die Sie im Abschnitt „mainSteps“ eines Automatisierungs-Runbooks definieren, nacheinander ausgeführt. Wenn eine Aktion abgeschlossen ist, beginnt die nächste, im Abschnitt „mainSteps“ angegebene Aktion. Mit dieser Version können Sie jetzt Automatisierungsworkflows erstellen, die eine bedingte Verzweigung durchführen. Dies bedeutet, dass Sie Automatisierungsworkflows erstellen können, die dynamisch auf geänderte Bedingungen reagieren und zu einem angegebenen Schritt springen. Informationen finden Sie unter Verwendung bedingter Anweisungen in Runbooks.	18. Juli 2018

SSM Agent ist jetzt auf Ubuntu Server 16.04 AMIs unter Verwendung von Snap vorinstalliert	Beginnend mit Instances, die anhand von Ubuntu Server 16.04-AMIs (identifiziert durch `20180627`) erstellt wurden, ist nun der SSM Agent unter Verwendung von Snap-Pakete vorinstalliert. Auf Instances, die aus früheren AMIs erstellt wurden, sollten Sie weiterhin deb-Installationsprogrammpakete verwenden. Weitere Informationen finden Sie unter Über SSM Agent-Installationen auf 64-Bit-Ubuntu Server 16.04-Instances.	7. Juli 2018
Überprüfen der für den SSM Agent erforderlichen minimalen S3-Bucket-Berechtigungen	Das neue Thema Minimale S3-Bucket-Berechtigungen für SSM Agent enthält Informationen zu den Amazon Simple Storage Service (Amazon S3)-Buckets, auf die Ressourcen möglicher weise zugreifen müssen, um Systems Manager-Operationen ausführen zu können. Sie können diese Buckets in einer benutzerdefinierten Richtlinie angeben, wenn Sie den S3-Bucket-Zugriff für ein Instance-Profil oder einen VPC-Endpunkt auf das für die Verwendung von Systems Manager erforderliche Minimum beschränken möchten.	5. Juli 2018

Anzeigen des vollständigen Ausführungsverlaufs für eine bestimmte State Manager-Zuordnungs-ID	Das neue Thema Anzeigen von Zuordnungsverläufen beschreibt, wie alle Ausführungen für eine bestimmte Zuordnungs-ID und anschließend Ausführungsdetails für eine oder mehrere Ressourcen angezeigt werden.	2. Juli 2018
Patch Manager bietet Support für Amazon Linux 2	Sie können nun mit Patch Manager Patches auf Amazon Linux 2-Instances anwenden. Allgemeine Informationen zur Betriebssystemunterstützung von Patch Manager finden Sie unter Voraussetzungen für Patch Manager. Informationen zu den unterstützten Schlüssel-Wert-Paaren für Amazon Linux 2 bei der Definition eines Patch-Filters finden Sie PatchFilterin der AWS Systems Manager API-Referenz.	26. Juni 2018
Befehlsausgabe an Amazon CloudWatch Logs senden	Das neue Thema Konfiguration von Amazon CloudWatch Logs für Run Command beschreibt, wie die Run Command Ausgabe an CloudWatch Logs gesendet wird.	18. Juni 2018

Schnelles Erstellen oder Löschen von Resource Data Sync für Inventory unter Verwendung von CloudFormation	Sie können CloudFormation es verwenden, um eine Ressourcendatensynchronisierung für Systems Manager Inventory zu erstellen oder zu löschen. Um sie zu verwenden CloudFormation, fügen Sie die AWS::SSM::ResourceDataSync-Ressource zu Ihrer CloudFormation Vorlage hinzu. Weitere Informationen finden Sie unter Arbeiten mit AWS CloudFormation - Vorlagen im AWS CloudFormation -Benutzerhandbuch. Sie können auch manuell eine Ressource Data Sync für Inventory erstellen. Einzelheiten dazu finden Sie unter Erstellen von Resource Data Sync für Inventory.	11. Juni 2018

AWS Systems Manager Aktualisierungsbenachrichtigungen für das Benutzerhandbuch sind jetzt über RSS verfügbar	Die HTML-Version des Systems Manager-Benutzerhandbuchs unterstützt jetzt einen RSS-Feed für Aktualisierungen, die auf der Seite Aktualisierungsverlauf der Systems Manager-Dokumentation dokumentiert sind. Der RSS-Feed umfasst Aktualisierungen ab Juni 2018 und später. Zuvor angekündigte Aktualisierungen stehen nach wie vor auf der Seite Aktualisierungsverlauf der Systems Manager-Dokumentation zur Verfügung. Verwenden Sie die RSS-Schaltfläche in der oberen Menüanzeige, um den Feed zu abonnieren.	6. Juni 2018
Angabe eines Beendigungscodes in Skripts, um verwaltete Instances neu zu starten	Das neue Thema Neustarten von verwalteten Instances von Skripts beschreibt, wie Sie Systems Manager anweisen, verwaltete Instances neu zu starten, indem Sie einen Beendigungscode in Skripts angeben, die Sie mit Run Command ausführen.	3. Juni 2018

Erstellen Sie ein Ereignis in Amazon, EventBridge wenn benutzerdefiniertes Inventar gelöscht wird	Das neue Thema Aktionen zum Löschen von Lagerbest and anzeigen in EventBridge beschreibt, wie Sie Amazon so konfigurieren, EventBrid ge dass jedes Mal, wenn ein Benutzer benutzerdefinierte s Inventar löscht, ein Ereignis erstellt wird.	1. Juni 2018

Updates vor Juni 2018

In der folgenden Tabelle sind wichtige Änderungen in jeder Version des AWS Systems Manager - Benutzerhandbuchs vor Juni 2018 beschrieben.

Änderungen	Beschreibung	Datum der Veröffent lichung
Inventarisieren Sie alle verwalteten Instanzen in Ihrem AWS-Konto	Sie können alle verwalteten Instanzen in Ihrem inventari sieren, AWS-Konto indem Sie eine globale Inventarz uordnung erstellen. Weitere Informationen finden Sie unter Inventarisieren Sie alle verwalteten Knoten in Ihrem AWS-Konto. **Note** Globale Bestandszuordnungen sind in SSM Agent-Version 2.0.790.0 oder höher verfügbar. Weitere Informationen zur Aktualisierung von SSM Agent auf Ihren Instances finden Sie unter Aktualisierung von SSM Agent mithilfe von Run Command.	3. Mai 2018
SSM Agent ist standardmäßig auf	SSM Agent ist standardmäßig auf Ubuntu Server 18.04 LTS 64-Bit- und 32-Bit-AMIs installiert.	2. Mai 2018

Änderungen	Beschreibung	Datum der Veröffentlichung
Ubuntu Server 18 installiert.		
Neues Thema	In dem neuen Thema Ausführen von Befehlen mit einer bestimmten Dokumentversion wird beschrieben, wie mit dem document-version-Parameter angegeben wird, welche Version eines SSM-Dokuments bei der Befehlsausführung verwendet werden soll.	1. Mai 2018
Neues Thema	Im neuen Thema Löschen eines benutzerdefinierten Bestands wird beschrieben, wie Sie benutzerdefinierte Bestandsdaten mithilfe der AWS CLI aus Amazon S3 löschen. Außerdem wird beschrieben, wie Sie mit `SchemaDeleteOption` benutzerdefinierten Bestand verwalten, indem Sie einen benutzerdefinierten Bestandstyp deaktivieren oder löschen. Diese neue Funktion verwendet den DeleteInventoryAPI-Vorgang.	19. April 2018
Amazon-SNS-Benachrichtigungen für SSM Agent	Abonnieren Sie ein Amazon SNS-Thema, um Benachrichtigungen zu erhalten, wenn eine neue Version von SSM Agent verfügbar ist. Weitere Informationen finden Sie unter Abonnieren von SSM Agent-Benachrichtigungen.	9. April 2018
CentOS Patch-Support	Systems Manager unterstützt jetzt das Patchen von CentOS-Instances. Weitere Informationen zu unterstützten CentOS-Versionen finden Sie unter Patch Manager-Voraussetzungen. Weitere Information zur Funktionsweise von Patch-Vorgängen finden Sie unter So funktionieren Patch Manager-Operationen.	29. März 2018

Änderungen	Beschreibung	Datum der Veröffentlichung
Neuer -Abschnitt	Um eine einzige Quelle für Referenzinformationen im AWS Systems Manager -Benutzerhandbuch bereitzustellen, wurde der neue Abschnitt AWS Systems Manager Referenz eingeführt. Zusätzliche Inhalte werden diesem Abschnitt hinzugefügt, sobald diese verfügbar werden.	15. März 2018
Neues Thema	Das neue Thema Paketnamen-Formate für genehmigte und abgelehnte Patch-Listen erläutert die Paketnameformate, die Sie in die Listen der genehmigten und abgelehnten Patches für eine benutzerdefinierte Patch-Baseline eingeben können. Beispiel-Formate werden für jeden Betriebssystem-Typ bereitgestellt, der Patch Manager unterstützt.	9. März 2018
Neues Thema	Systems Manager ist jetzt in Chef integriertChef InSpec. InSpec ist ein Open-Source-Runtime-Framework, mit dem Sie menschenlesbare Profile auf GitHub Amazon S3 erstellen können. Anschließend können Sie Systems Manager verwenden, um Compliance-Scans auszuführen und konforme und nicht konforme Instances anzuzeigen. Weitere Informationen finden Sie unter Verwenden von Chef InSpec-Profilen mit Systems Manager Compliance.	7. März 2018
Neues Thema	In dem neuen Thema Verwenden von serviceverknüpften Rollen für Systems Manager wird beschrieben, wie Sie eine AWS Identity and Access Management (IAM) -Servicevverknüpfte Rolle mit Systems Manager verwenden. Derzeit sind serviceverknüpfte Rollen nur erforderlich, wenn Sie Systems Manager Inventory zum Sammeln von Metadaten über Tags und Ressourcengruppen verwenden.	27. Februar 2018

Änderungen	Beschreibung	Datum der Veröffentlichung
Neue und aktualisierte Themen	Sie können nun Patch Manager verwenden, um Patches zu installieren, die sich in einem anderen Quell-Repository befinden als die Standardversion, die in der Instance konfiguriert ist. Dies ist nützlich für das Einspielen von Patches in Instances mit Updates, die sich nicht auf Sicherheit beziehen, mit dem Content von Personal Package Archives (PPA) für Ubuntu Server, mit Updates für interne Unternehmensanwendungen und so weiter. Sie geben beim Erstellen einer benutzerdefinierten Patch-Baseline alternative Patch-Quell-Repositorys an. Weitere Informationen finden Sie unter den folgenden Themen: • So geben Sie ein alternatives Patch-Quell-Repository an (Linux) • Arbeiten mit benutzerdefinierten Patch-Baselines • Erstellen einer Patch-Baseline mit benutzerdefinierten Repositorys für verschiedene Betriebssystemversionen	6. Februar 2018
Neues Thema	In dem neuen Thema SSM-Befehlsdokumente zum Patchen verwalteter Knoten werden die sieben verfügbaren SSM-Dokumente, die Ihnen dabei helfen, Ihre verwalteten Instances mit den neuesten sicherheitsrelevanten Updates zu patchen, beschrieben.	10. Januar 2018
Wichtige Updates zur Linux-Unterstützung	Verschiedene Themen wurden mit den folgenden Informationen aktualisiert: • SSM Agent ist standardmäßig in AMIs von September 2017 und später, die auf Amazon Linux 1 basieren, installiert. • Auf anderen Versionen von Linux müssen Sie den SSM Agent manuell installieren, auch auf Nicht-Basis-Images wie Amazon ECS-optimierten AMIs.	9. Januar 2018

Änderungen	Beschreibung	Datum der Veröffentlichung
Neues Thema	Das neue Thema SSM-Befehlsdokument zum Patchen: `AWS-RunPatchBaseline` enthält Einzelheiten dazu, wie dieses SSM-Dokument auf Windows- und Linux-Systemen funktioniert. Außerdem erhalten Sie Informationen zu den zwei verfügbaren Parametern im Dokument `AWS-RunPatchBaseline` , `Operation` und `Snapshot ID`.	5. Januar 2018
Neue Themen	Der neue Abschnitt So funktionieren Patch Manager-Operationen enthält technische Details, die erklären, wie Patch Manager bestimmt, welche Sicherheitspatches installiert werden und wie er sie auf dem jeweiligen unterstützten Betriebssystem installiert. Er bietet außerdem Informationen darüber, wie Patch-Baseline-Regeln auf verschiedenen Verteilungen des Linux-Betriebssystems funktionieren.	2. Januar 2018
Systems Manager-Automatisierungsaktionsreferenz umbenannt und verschoben	Auf Grundlage des Kundenfeedbacks wird die Automatisierungsaktionsreferenz jetzt Systems Manager Automation–Runbook-Referenz genannt. Außerdem haben wir die Referenz in den Knoten „Freigegebene Ressourcen > Dokumente“ verschoben, sodass sie sich näher an Referenz für Befehlsdokument-Plugins befinden. Weitere Informationen finden Sie unter Systems Manager Automation Aktionen-Referenz.	20. Dezember 2017

Änderungen	Beschreibung	Datum der Veröffentlichung
Neue Kapitel und Inhalte zur Überwachung	Ein neues Kapitel,Einloggen und Überwachen AWS Systems Manager, enthält Anweisungen zum Senden von Metriken und Protokolldaten an Amazon CloudWatch Logs. Ein neues ThemaSenden von Knotenprotokollen an Unified CloudWatch Logs (CloudWatch Agent), enthält Anweisungen für die Migration von Aufgaben zur Instanzüberwachung (nur für Windows Server 64-Bit-Instances) vom Agenten SSM Agent zum CloudWatch Agenten.	14. Dezember 2017
Neues Kapitel	Ein neues Kapitel enthält umfassende Informationen zur Verwendung von AWS Identity and Access Management (IAM) und AWS Systems Manager zur Sicherung des Zugriffs auf Ihre Ressourcen mithilfe von Anmeldeinformationen. Identity and Access Management für AWS Systems Manager Diese Anmeldeinformationen bieten die erforderlichen Berechtigungen für den Zugriff auf AWS Ressourcen, z. B. für den Zugriff auf Daten, die in S3-Buckets gespeichert sind, und das Senden von Befehlen an und das Lesen der Tags auf EC2 Instances.	11. Dezember 2017
Änderungen an der linken Navigationsleiste	Wir haben die Überschriften im linken Navigationsbereich dieses Benutzerhandbuchs geändert und an die Überschriften in der neuen AWS Systems Manager -Konsole angepasst.	8. Dezember 2017

Änderungen	Beschreibung	Datum der Veröffentlichung
Mehrere Änderungen für re:Invent 2017	• Offizieller Start von AWS Systems Manager: AWS Systems Manager (ehemals Amazon EC2 Systems Manager) ist eine einheitliche Oberfläche, mit der Sie Betriebsdaten zentralisieren und Aufgaben AWS ressourcenübergreifend automatisieren können. Sie können hier auf die neue AWS Systems Manager Konsole zugreifen. Weitere Informationen finden Sie unter Was ist AWS Systems Manager?. • YAML-Support: Sie können SSM-Dokumente im YAML-Format erstellen. Weitere Informationen finden Sie unter AWS Systems Manager-Documents.	29. November 2017
Verwendung von Run Command zur Aufnahme VSS-fähiger Snapshots von EBS-Volumes	Mithilfe Run Command können Sie anwendungskonsistente Snapshots aller Amazon Elastic Block Store (Amazon EBS) -Volumes erstellen, die an Ihre Amazon Windows-Instances angehängt sind. EC2 Der Snapshot-Vorgang erstellt mit dem Windows Volume Shadow Copy Service (VSS) Backups VSS-fähiger Anwendungen auf Image-Ebene. Dazu gehören auch Daten von schwebenden Transaktionen zwischen diesen Anwendungen und dem Datenträger. Des Weiteren müssen Sie Ihre Instances herunterfahren oder trennen, wenn Sie eine Sicherung aller angefügten Volumes durchführen möchten. Weitere Informationen finden Sie unter Verwenden von Microsoft VSS-fähigen Snapshots AWS Systems Manager im EC2 Amazon-Benutzerhandbuch.	20. November 2017

Änderungen	Beschreibung	Datum der Veröffentlichung
Verbesserte Systems Manager-Sicherheit durch Verwendung von VPC-Endpunkten	Sie können die Sicherheitslage Ihrer verwalteten Instances (einschließlich verwalteter Instances in Ihrer Hybrid-Umgebung) verbessern, indem Sie Systems Manager so konfigurieren, dass ein Schnittstellen-VPC-Endpunkt verwendet wird. Schnittstellenendpunkte werden von einer Technologie unterstützt PrivateLink, mit der Sie über private IP-Adressen privat auf Amazon EC2 und Systems Manager APIs zugreifen können. PrivateLink schränkt den gesamten Netzwerkverkehr zwischen Ihren verwalteten Instances, Systems Manager und EC2 dem Amazon-Netzwerk ein (verwaltete Instances haben keinen Zugriff auf das Internet). Zudem benötigen Sie kein Internet-Gateway, kein NAT-Gerät und kein Virtual Private Gateway. Weitere Informationen finden Sie unter Verbessern der Sicherheit von EC2 Instances mithilfe von VPC-Endpunkten für Systems Manager.	7. November 2017

Änderungen	Beschreibung	Datum der Veröffentlichung
Inventory-Support für Dateien, Services, Windows-Rollen und die Windows-Registry	SSM Inventory unterstützt jetzt das Sammeln folgender Informationen von Ihren verwalteten Instances. • Files (Dateien): Name, Größe, Version, Installationsdatum, Änderung und Zeitpunkt der letzten Zugriffe usw. • Services: Name, Anzeigename, Status, abhängige Services, Servicetyp, Starttyp usw. • Windows-Registry: Registry-Schlüsselpfad, Wertname, Werttyp und Wert. • Windows-Rollen: Name, Anzeigename, Pfad, Featuretyp, Installationsstatus usw. Bevor Sie versuchen, Informationen für diese Bestandstypen zu sammeln, aktualisieren Sie SSM Agent auf den Instances, die Sie für den Bestand wünschen. Durch Ausführen der neuesten Version von SSM Agent stellen Sie sicher, dass Sie Metadaten für alle unterstützten Bestandstypen sammeln können. Informationen zur Aktualisierung von SSM Agent mithilfe von State Manager finden Sie unter Exemplarische Vorgehensweise: Automatisches Update SSM Agent mit AWS CLI. Weitere Informationen zu Inventory finden Sie unter Weitere Informationen über Systems Manager Inventory.	6. November 2017
Aktualisierungen der Automation-Dokumentation	Mehrere Probleme mit der Information über die Einrichtung und Konfiguration des Zugriffs für Systems Manager Automation behoben. Weitere Informationen finden Sie unter Einrichten der Automatisierung.	31. Oktober 2017

Änderungen	Beschreibung	Datum der Veröffentlichung
Integration von GitHub und Amazon S3	Remote-Skripts ausführen: Systems Manager unterstützt jetzt das Herunterladen und Ausführen von Skripts von einem privaten oder öffentlichen GitHub-Repository und von Amazon S3. Mit dem `AWS-RunRemoteScript` vordefinierten SSM-Dokument oder dem `aws:downloadContent` Plugin in einem benutzerdefinierten SSM-Dokument können Sie Ansible Playbooks und Skripte in Python, Ruby oder, um nur einige zu nennen PowerShell, ausführen. Diese Änderungen verbessern die Infrastruktur als Code weiter, wenn Sie Systems Manager verwenden, um die Konfiguration und Bereitstellung von EC2 Instanzen und lokal verwalteten Instanzen in Ihrer Hybridumgebung zu automatisieren. Weitere Informationen erhalten Sie unter Ausführen von Skripts von GitHub und Ausführen von Skripts von Amazon S3. Zusammengesetzte SSM-Dokumente erstellen: Systems Manager unterstützt jetzt die Ausführung von einem oder mehreren sekundären SSM-Dokumenten über ein primäres SSM-Dokument. Diese primären Dokumente, die andere Dokumente ausführen, werden als zusammengesetzte Dokumente bezeichnet. Mit Composite Documents können Sie einen Standardsatz sekundärer SSM-Dokumente AWS-Konten für allgemeine Aufgaben wie das Booten von Antivirensoftware oder den Beitritt zu Domänen erstellen und gemeinsam nutzen. Sie können zusammengesetzte und sekundäre Dokumente ausführen, die in Systems Manager, GitHub oder Amazon S3 gespeichert sind. Nach dem Erstellen eines zusammengesetzten Dokuments können Sie es mithilfe des vordefinierten `AWS-RunDocument` -SSM-Dokuments ausführen. Weitere Informationen erhalten Sie unter Erstellen von zusammeng	26. Oktober 2017

Änderungen	Beschreibung	Datum der Veröffentlichung
	esetzten Dokumenten und Ausführen von -Dokumenten von Remote-Standorten. SSM-Dokumenten-Plugin-Referenz: Zum einfacheren Zugriff haben wir die SSM-Plugin-Referenz für SSM-Dokumente aus der Systems Manager-API-Referenz in das Benutzerhandbuch verschoben. Weitere Informationen finden Sie unter Referenz für Befehlsdokument-Plugins.	
Support für Parameterversionen in Parameter Store	Wenn Sie einen Parameter bearbeiten, erhält Parameter Store jetzt automatisch die Versionsnummer 1. Sie können einen Parameternamen und eine bestimmte Versionsnummer in API-Aufrufen und SSM-Dokumenten angeben. Wenn Sie keine Versionsnummer angeben, verwendet das System automatisch die neueste Version. Parameterversionen bieten eine Schutzebene für den Fall, dass ein Parameter versehentlich geändert wird. Sie können die Werte aller Versionen anzeigen und bei Bedarf auf ältere Versionen verweisen. Sie können auch Parameterversionen verwenden, um zu sehen, wie oft ein Parameter im Lauf eines bestimmten Zeitraums geändert wurde. Weitere Informationen finden Sie unter Arbeiten mit Parameterversionen in Parameter Store.	24. Oktober 2017
Unterstützung für Markierungen von Systems Manager-Dokumenten	Sie können jetzt die AddTagsToResourceAPI, die oder die verwenden, AWS -Tools für PowerShell um Systems Manager Manager-Dokumente mit Schlüssel-Wert-Paaren zu kennzeichnen. AWS CLI Das Markieren hilft bei der schnellen Identifizierung bestimmter Ressourcen basierend auf den ihnen zugewiesenen Tags. Dies ist ein Zusatz zur vorhandenen Markierungsunterstützung für verwaltete Instances, Wartungsfenster, Parameter Store-Parameter und Patch-Baselines.	3. Oktober 2017

Änderungen	Beschreibung	Datum der Veröffentlichung
Verschiedene Dokumentations-Aktualisierungen zur Korrektur von Fehlern oder Aktualisierung von Inhalt basierend auf Feedback	• Verwalten von Knoten in Hybrid- und Multi-Cloud-Umgebungen mit Systems Manager wurde mit Informationen zu Raspbian Linux aktualisiert. • Verwalten von EC2-Instances mit Systems ManagerMit neuen Anforderungen für Windows Server Instanzen aktualisiert. SSM Agenterfordert Windows PowerShell 3.0 oder höher, um bestimmte SSM-Dokumente auf Windows Server Instanzen auszuführen (z. B. das ältere `AWS-ApplyPatchBaseline` SSM-Dokument). Vergewissern Sie sich, dass Ihre Windows Server-Instances auf Windows Management Framework 3.0 oder höher ausgeführt werden. Das Framework umfasst PowerShell. Weitere Informationen finden Sie unter Windows Management Framework 3.0	2. Oktober 2017
Beheben Sie Probleme mit nicht erreichbaren Windows-Instanzen mithilfe des EC2 Rescue-Automation-Workflows	EC2Rescue kann Ihnen helfen, Probleme auf EC2 Windows Server Amazon-Instances zu diagnostizieren und zu beheben. Sie können das Tool mithilfe des AWSSupport-ExecuteEC2Rescue-Dokuments als Systems Manager Automation-Workflow ausführen. Das AWSSupport-ExecuteEC2Rescue-Dokument wurde entwickelt, um eine Kombination aus Aktionen, CloudFormation Aktionen und Lambda-Funktionen von Systems Manager auszuführen, die die Schritte automatisieren, die normalerweise für die Verwendung EC2 von Rescue erforderlich sind. Weitere Informationen finden Sie unter Führen Sie das EC2 Rescue-Tool auf nicht erreichbaren Instanzen aus.	29. September 2017

Änderungen	Beschreibung	Datum der Veröffentlichung
SSM Agent standardmäßig auf Amazon Linux installiert	SSM Agent ist standardmäßig auf Amazon Linux AMIs von September 2017 und später, die auf Amazon Linux basieren, installiert. Installieren Sie SSM Agent auf anderen Linux-Versionen manuell. Dies wird in Arbeiten mit SSM Agent auf EC2-Instances für Linux beschrieben.	27. September 2017
Run Command-Verbesserungen	Run Command umfasst die folgenden Erweiterungen. • Sie können die Befehlsausführung auf bestimmte Instances beschränken, indem Sie eine IAM-Richtlinie erstellen und zuweisen, die eine Bedingung beinhaltet, dass der Benutzer nur Befehle auf Instances ausführen kann, die mit bestimmten EC2 Amazon-Tags gekennzeichnet sind. Weitere Informationen finden Sie unter Den Zugriff von Run Command anhand von Tags beschränken. • Sie haben mehr Optionen für das Targeting von Instances mithilfe von EC2 Amazon-Tags. Sie können jetzt beim Senden von Befehlen mehrere Tag-Schlüssel und mehrere Tag-Werte angeben. Weitere Informationen finden Sie unter Ausführen von Befehlen in großem Maßstab.	12. September 2017
Auf Raspbian unterstützter Systems Manager	Systems Manager kann jetzt auf Raspbian Jessie- und Raspbian Stretch-Geräten, einschließlich Raspberry Pi (32-Bit), ausgeführt werden.	7. September 2017
Automatisches Senden von SSM Agent Protokollen an Amazon CloudWatch Logs	Sie können jetzt auf Ihren Instances eine einfache Konfigurationsänderung vornehmen, damit SSM Agent Protokolldateien an CloudWatch sendet. Weitere Informationen finden Sie unter SSM AgentProtokolle an CloudWatch Logs senden.	7. September 2017

Änderungen	Beschreibung	Datum der Veröffentlichung
Resource Data Sync verschlüsseln	Mit Systems Manager Resource Data Sync können Sie die auf Hunderten von verwalteten Instances erfassten Bestandsdaten in einem zentralen S3-Bucket zusammenfassen. Sie können Resource Data Sync jetzt mit einem AWS Key Management Service -Schlüssel verschlüsseln. Weitere Informationen finden Sie unter Walkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten.	1. September 2017
Neue Anleitungen zu State Manager	Der State Manager-Dokumentation wurden zwei neue Anleitungen hinzugefügt: Exemplarische Vorgehensweise: Automatisches Update SSM Agent mit AWS CLI Exemplarische Vorgehensweise: Automatisches Aktualisieren von PV-Treibern auf EC2 Instanzen für Windows Server	31. August 2017
Configuration Compliance für Systems Manager	Mit Configuration Compliance können Sie Ihre Flotte verwalteter Instances auf Patch-Compliance und Konfigurationsinkonsistenzen prüfen. Sie können Daten aus mehreren Bereichen sammeln AWS-Konten und AWS-Regionen aggregieren und dann nach bestimmten Ressourcen suchen, die nicht den Vorschriften entsprechen. Standardmäßig werden von Configuration Compliance die Compliance-Daten zu Patch Manager-Patches und State Manager-Zuordnungen angezeigt. Sie können auch den Service anpassen und Ihre eigenen Compliance-Typen auf Grundlage Ihrer IT- oder Business-Anforderungen erstellen. Weitere Informationen finden Sie unter AWS Systems Manager-Compliance.	28. August 2017

Änderungen	Beschreibung	Datum der Veröffentlichung
Neue Automatisierungsaktion: `aws:executeAutomation`	Führt einen sekundären Automatisierungsworkflow durch Aufrufen eines sekundären Automatisierungs-Runbooks aus. Mit dieser Aktion können Sie die Automatisierungs-Runbooks für die gängigsten Workflows erstellen und während der Ausführung der Automatisierung auf diese Dokumente verweisen. Mit dieser Aktion können Sie Ihre Automatisierungs-Runbooks vereinfachen, indem Sie die Notwendigkeit für wiederholte Schritte bei ähnlichen Runboks entfernen. Weitere Informationen finden Sie unter `aws:executeAutomation` - Führen Sie eine weitere Automatisierung durch.	22. August 2017
Automatisierung als Ziel eines CloudWatch Ereignisses	Sie können einen Automatisierungs-Workflow starten, indem Sie ein Automation-Runbook als Ziel eines CloudWatch Amazon-Ereignisses angeben. Sie können Workflows nach einem Zeitplan starten oder wenn ein bestimmtes AWS Systemereignis eintritt. Weitere Informationen finden Sie unter Führen Sie Automatisierungen auf EventBridge der Grundlage von Ereignissen aus.	21. August 2017
State Manager-Zuordnungs-Versionierung und allgemeine Aktualisierungen	Sie können jetzt verschiedene State Manager-Zuordnungsversionen erstellen. Es gilt ein Kontingent von 1 000 Versionen pro Zuordnung. Sie können auch Namen für Ihre Zuordnungen angeben. Die State Manager-Dokumentation wurde aktualisiert, um veraltete Informationen und Inkonsistenzen zu behandeln. Weitere Informationen finden Sie unter AWS Systems Manager State Manager.	21. August 2017

Änderungen	Beschreibung	Datum der Veröffentlichung
Änderungen an Maintenance Windows	Maintenance Windows umfasst die folgenden Änderungen oder Verbesserungen: • Vorher konnte Maintenance Windows Aufgaben nur mithilfe von Run Command durchführen. Sie können jetzt Aufgaben mithilfe von Systems Manager Automation AWS Lambda, und ausführen AWS Step Functions. • Sie können die Ziele eines Wartungsfensters bearbeiten sowie einen Zielnamen, eine Beschreibung und einen Eigentümer angeben. • Sie können Aufgaben in einem Wartungsfenster bearbeiten. Dies schließt die Angabe eines neuen SSM-Dokuments für Run Command- und Automation-Aufgaben ein. • Alle Run Command Parameter werden jetzt unterstützt DocumentHash, einschließlich DocumentHashType, TimeoutSeconds, Kommentar und NotificationConfig. • Sie können jetzt eine `safe`-Kennzeichnung verwenden, wenn Sie versuchen, ein Ziel abzumelden. Wenn diese Option aktiviert ist, wird vom System ein Fehler zurückgegeben, falls eine beliebige Aufgabe auf das Ziel verweist. Weitere Informationen finden Sie unter AWS Systems Manager Maintenance Windows.	16. August 2017

Änderungen	Beschreibung	Datum der Veröffentlichung
Neue Automatisierungsaktion: `aws:approve`	Diese neue Aktion für Automation-Runbooks hält eine Automation-Ausführung zeitweise an, bis die Aktion von designierten Prinzipalen genehmigt oder abgelehnt wird. Nach Erreichen der erforderlichen Anzahl an Genehmigungen wird die Ausführung der Automatisierung fortgesetzt. Weitere Informationen finden Sie unter Systems Manager Automation Aktionen-Referenz.	10. August 2017
Assume-Rolle bei Automation nicht mehr erforderlich	Bisher mussten Sie für die Automatisierung eine Servicerolle (oder assume-Rolle) festlegen, damit der Service in Ihrem Auftrag Aktionen ausführen konnte. Diese Rolle ist für die Automatisierung nicht mehr erforderlich, da der Service jetzt den Kontext des Benutzers verwendet, der die Ausführung aufgerufen hat. In den folgenden Situationen müssen Sie jedoch nach wie vor eine Servicerolle zur Automatisierung angeben: • Wenn Sie die Zugriffsberechtigungen eines Benutzers für eine Ressource einschränken, aber dem Benutzer die Ausführung eines Automation-Workflows gestatten möchten, der höhere Berechtigungen erfordert. In diesem Szenario können Sie eine Servicerolle mit höheren Berechtigungen erstellen und dem Benutzer das Ausführen des Workflows gestatten. • Vorgänge, deren Ausführung erwartungsgemäß länger als 12 Stunden dauern, erfordern eine Servicerolle. Weitere Informationen finden Sie unter Einrichten der Automatisierung.	3. August 2017

Änderungen	Beschreibung	Datum der Veröffentlichung
Konfigurations-Compliance	Verwenden Sie Amazon EC2 Systems Manager Configuration Compliance, um Ihre Flotte verwalteter Instances auf Patch-Konformität und Konfigurationsinkonsistenzen zu überprüfen. Sie können Daten aus mehreren Bereichen sammeln und aggregieren AWS-Konten und dann nach bestimmten Ressourcen suchen AWS-Regionen, die nicht den Vorschriften entsprechen. Weitere Informationen finden Sie unter AWS Systems Manager-Compliance.	8. August 2017
SSM-Dokumentverbesserungen	SSM-Befehls- und Richtliniendokumente bieten jetzt plattformübergreifende Unterstützung. Das bedeutet, dass ein einzelnes SSM-Dokument Plugins für die Betriebssysteme Windows und Linux verarbeiten kann. Mit plattformübergreifender Unterstützung können Sie die Anzahl der verwalteten Dokumente konsolidieren. Plattformübergreifende Unterstützung wird in SSM-Dokumenten mit Schema-Version 2.2 oder höher angeboten. SSM-Command-Dokumente mit Schema-Version 2.0 oder höher können jetzt mehrere Plugins desselben Typs enthalten. Beispiel: Sie können ein Command-Dokument erstellen, mit dem das Plugin `aws:runRunShellScript` mehrmals aufgerufen wird. Weitere Informationen zu den Änderungen bei Schema-Version 2.2 finden Sie unter AWS Systems Manager - Dokumente. Weitere Informationen zu SSM-Plugins finden Sie in der Referenz zu Befehlsdokumenten-Plugins.	12. Juli 2017

Änderungen	Beschreibung	Datum der Veröffentlichung
Linux-Patching	Patch Manager kann jetzt die folgenden Linux-Distributionen patchen: 64-Bit- und 32-Bit-Systeme • Amazon Linux 2014.03, 2014.09 oder höher • Ubuntu Server 16.04 LTS, 14.04 LTS oder 12.04 LTS • Red Hat Enterprise Linux (RHEL) 6.5 oder höher Nur 64-Bit-Systeme • Amazon Linux 2015.03, 2015.09 oder höher • Red Hat Enterprise Linux(RHEL) 7.x oder höher Weitere Informationen finden Sie unter AWS Systems Manager Patch Manager. Note • Für das Einspielen von Patches auf Linux-Instances muss der SSM Agent mit Version 2.0.834.0 oder höher auf Ihren Instances ausgeführt werden. Informationen zur Aktualisierung des Agenten finden Sie im Abschnitt Beispiel: Aktualisierung von SSM Agent unter Ausführen von Befehlen über die Konsole. • Das `AWS-ApplyPatchBaseline` SSM-Dokument wird durch das `AWS-RunPatchBaseline` -Dokument ersetzt.	6. Juli 2017

Änderungen	Beschreibung	Datum der Veröffentlichung
Ressourcen-Datensynchronisierung	Sie können mit Systems Manager Resource Data Sync Bestandsdaten aus allen Ihren verwalteten Instances in einen einzelnen Amazon S3-Bucket senden. Resource Data Sync aktualisiert die Daten dann automatisch, wenn neue Bestandsdaten erfasst werden. Wenn alle Inventardaten in einem Ziel-S3-Bucket gespeichert sind, können Sie Dienste wie Amazon Athena und Amazon Quick Suite verwenden, um die aggregierten Daten abzufragen und zu analysieren.Weitere Informationen finden Sie unter. Erstellen einer Resource Data Sync für Inventory Ein Beispiel für die Arbeit mit Resource Data Sync finden Sie unter Walkthrough: Verwenden von Resource Data Sync zum Aggregieren von Bestandsdaten.	29. Juni 2017
Systems Manager-Parameterhierarchien	Das Verwalten Dutzender oder Hunderter von Systems Manager-Parametern als unsortierte Liste ist zeitaufwendig und fehleranfällig. Mit Parameterhierarchien können Sie Systems Manager-Parameter leichter organisieren und verwalten. Bei einer Hierarchie handelt es sich um einen Parameternamen mit einem Pfad, den Sie mit Schrägstrichen definieren. Hier finden Sie ein Beispiel mit drei Hierarchieebenen im Namen. Damit wird Folgendes identifiziert: /Environment/Type of computer/Application/Data `/Dev/DBServer/MySQL/db-string13` Weitere Informationen finden Sie unter Arbeiten mit Parameterhierarchien in Parameter Store.	22. Juni 2017

Änderungen	Beschreibung	Datum der Veröffentlichung
SSM Agent-Unterstützung für SUSE Linux Enterprise Server	Sie können SSM Agent auf 64-Bit SUSE Linux Enterprise Server (SLES) installieren. Weitere Informationen finden Sie unter Arbeiten mit SSM Agent auf EC2-Instances für Linux.	14. Juni 2017

Dokumentkonventionen

Nachfolgend finden Sie allgemeine typografischen Konventionen für das AWS Systems Manager - Benutzerhandbuch.

Differenzierte Beispiele für lokale Betriebssysteme oder Befehlszeilensprachen

Wir verwenden Registerkarten zur Darstellung verschiedener Befehlsbeispiele, die auf dem lokalen Betriebssystemtyp eines Benutzers basieren. In den Beispielen für Linux und macOS verwenden wir den umgekehrten Schrägstrich (\), um lange Befehle in mehrere Zeilen aufzuteilen. In den Windows Server-Beispielen verwenden wir das Caret-Zeichen (^), um Befehle in mehrere Zeilen aufzuteilen.

Beispiel:

Linux & macOS

```
aws ssm update-service-setting \
    --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
instance/activation-tier \
    --setting-value advanced
```

Windows

```
aws ssm update-service-setting ^
    --setting-id arn:aws:ssm:region:aws-account-id:servicesetting/ssm/managed-
instance/activation-tier ^
    --setting-value advanced
```

Elemente auf der Benutzeroberfläche

Formatierung: fett ausgezeichneter Text

Beispiel: Wählen Sie File, Properties.

Benutzereingabe (Text, den ein Benutzer eingibt)

Format: Text in einer nicht proportionalen Schriftart

Beispiel: Geben Sie als Namen **`my-new-resource`** ein.

Platzhaltertext für einen erforderlichen Wert

Formatierung: Text in *italics*

Beispiel:

```
aws ec2 register-image --image-location amzn-s3-demo-bucket/image.manifest.xml
```