



Guida per l'utente

# AWS AppConfig



# AWS AppConfig: Guida per l'utente

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discreditì Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

# Table of Contents

Che cos'è AWS AppConfig? .....	1
Nozioni di base su AWS AppConfig .....	1
AWS AppConfig casi d'uso .....	1
Panoramica dei vantaggi .....	2
Come funziona AWS AppConfig .....	3
Prezzi per AWS AppConfig .....	5
AWS AppConfig quote .....	5
Risorse aggiuntive .....	5
Blog .....	6
SDKs .....	6
Configurazione AWS AppConfig .....	7
Iscriviti per un Account AWS .....	7
Crea un utente con accesso amministrativo .....	7
Concessione dell'accesso programmatico .....	9
Comprendere del supporto IPv6 .....	10
Configura le autorizzazioni per il rollback automatico .....	11
Passaggio 1: creare la politica di autorizzazione per il rollback in base agli allarmi CloudWatch .....	12
Passaggio 2: crea il ruolo IAM per il rollback in base CloudWatch agli allarmi .....	13
Fase 3: aggiunta di una relazione di trust .....	14
Creazione in corso .....	15
Comprendere del ruolo IAM del profilo di configurazione .....	16
Creazione di un namespace .....	19
Creazione di un' AWS AppConfig applicazione (console) .....	19
Creazione di un' AWS AppConfig applicazione (riga di comando) .....	20
Creazione di ambienti .....	22
Creazione di un AWS AppConfig ambiente (console) .....	22
Creazione di un AWS AppConfig ambiente (riga di comando) .....	23
Creazione di un profilo di configurazione in AWS AppConfig .....	25
Creazione di un profilo di configurazione del feature flag .....	29
Creazione di un profilo di configurazione in formato libero .....	61
Creazione di un profilo di configurazione per fonti di dati non native .....	77
Implementazione .....	80
Utilizzo delle strategie di distribuzione .....	81

Utilizzo di strategie di distribuzione .....	84
Creazione di una strategia di distribuzione .....	86
Distribuzione di una configurazione .....	90
Implementa una configurazione (console) .....	91
Implementa una configurazione (riga di comando) .....	92
Distribuzione con CodePipeline .....	96
Come funziona l'integrazione .....	97
Ripristino di una configurazione .....	97
Recupero .....	100
Che cos'è Agent AWS AppConfig ? .....	101
Come utilizzare AWS AppConfig Agent per recuperare i dati di configurazione .....	103
Utilizzo di Agent con AWS AppConfigAWS Lambda .....	104
Utilizzo di AWS AppConfig Agent con Amazon EC2 e macchine locali .....	205
Utilizzo di AWS AppConfig Agent con Amazon ECS e Amazon EKS .....	224
Recupero dei flag di funzionalità .....	243
Utilizzo di un manifesto per abilitare funzionalità di recupero aggiuntive .....	247
Generazione di un client utilizzando la specifica OpenAPI .....	258
Utilizzo della modalità di sviluppo locale di Agent AWS AppConfig .....	260
considerazioni sull'uso del browser e dei dispositivi mobili .....	265
Dati di configurazione e recupero delle bandiere .....	266
Autenticazione e Amazon Cognito .....	266
Caching .....	267
Segmentazione .....	267
Larghezza di banda (casi d'uso mobili) .....	268
Casi d'uso aggiuntivi delle bandiere .....	268
Recupero dei dati di configurazione senza Agent AWS AppConfig .....	268
(Esempio) Recupero di una configurazione chiamando AWS AppConfig APIs .....	271
Estensione dei flussi di lavoro AWS AppConfig .....	273
Comprendere AWS AppConfig le estensioni .....	273
Passaggio 1: Stabilisci cosa vuoi fare con le estensioni .....	274
Passo 2: Determina quando vuoi che l'estensione venga eseguita .....	275
Passaggio 3: Creare un'associazione di estensioni .....	276
Passaggio 4: Implementa una configurazione e verifica che le azioni di estensione vengano eseguite .....	277
Lavorare con le AWS estensioni create .....	277
Utilizzo dell'estensione Amazon CloudWatch Evidently .....	278

Utilizzo degli eventi AWS AppConfig di distribuzione nell' EventBridge estensione Amazon	279
Utilizzo degli eventi AWS AppConfig di distribuzione nell'estensione Amazon SNS	281
Utilizzo degli eventi AWS AppConfig di distribuzione nell'estensione Amazon SQS	284
Usare l'estensione Jira	286
Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig	292
Fase 1: Creare una funzione Lambda per un'estensione personalizzata AWS AppConfig	293
Passaggio 2: configura le autorizzazioni per un'estensione personalizzata AWS AppConfig	300
Fase 3: Creare un'estensione personalizzata AWS AppConfig	301
Fase 4: Creare un'associazione di estensioni per un'estensione personalizzata AWS AppConfig	306
Esempi di codice	308
Creazione o aggiornamento di una configurazione in formato libero archiviata nell'archivio di configurazione ospitato	308
Creazione di un profilo di configurazione per un segreto archiviato in Secrets Manager	311
Implementazione di un profilo di configurazione	312
Utilizzo di AWS AppConfig Agent per leggere un profilo di configurazione in formato libero	317
Utilizzo di AWS AppConfig Agent per leggere un indicatore di funzionalità specifico	319
Utilizzo di AWS AppConfig Agent per recuperare un flag di funzionalità con varianti	320
Utilizzo dell'azione GetLatestConfiguration API per leggere un profilo di configurazione in formato libero	322
Pulizia dell'ambiente	333
Deletion protection (Protezione da eliminazione)	339
Ignorare o forzare un controllo di protezione da eliminazione	340
Sicurezza	343
Implementazione dell'accesso con privilegi minimi	343
Crittografia dei dati a riposo per AWS AppConfig	344
AWS PrivateLink	349
Considerazioni	349
Creazione di un endpoint di interfaccia	350
Creazione di una policy dell'endpoint	350
Rotazione dei tasti di Secrets Manager	351
Impostazione della rotazione automatica dei segreti di Secrets Manager distribuita da AWS AppConfig	351
Monitoraggio	354
CloudTrail registri	355

---

AWS AppConfig eventi relativi ai dati in CloudTrail .....	356
AWS AppConfig eventi di gestione in CloudTrail .....	358
AWS AppConfig esempi di eventi .....	358
Registrazione delle metriche per le chiamate sul piano AWS AppConfig dati .....	360
Creazione di un allarme per una metrica CloudWatch .....	362
Monitoraggio delle implementazioni per il rollback automatico .....	363
Metriche consigliate da monitorare per il rollback automatico .....	364
Cronologia dei documenti .....	371
	cd

# Che cos'è AWS AppConfig?

AWS AppConfig i flag di funzionalità e le configurazioni dinamiche aiutano i produttori di software a regolare in modo rapido e sicuro il comportamento delle applicazioni negli ambienti di produzione senza implementare codice completo. AWS AppConfig accelera la frequenza di rilascio del software, migliora la resilienza delle applicazioni e aiuta a risolvere più rapidamente i problemi emergenti.

Grazie ai flag di funzionalità, è possibile rilasciare gradualmente nuove funzionalità per gli utenti e misurare l'impatto di tali modifiche prima di implementare completamente le nuove funzionalità per tutti gli utenti. Grazie ai flag operativi e alle configurazioni dinamiche, è possibile aggiornare gli elenchi di blocchi, gli elenchi di autorizzazioni, i vincoli di limitazione (della larghezza di banda della rete), la verbosità di registrazione ed eseguire altre ottimizzazioni operative per reagire rapidamente ai problemi negli ambienti di produzione.

## Nozioni di base su AWS AppConfig

Il video seguente può aiutarti a comprendere le funzionalità di AWS AppConfig

Visualizza altri AWS video sul [YouTube canale Amazon Web Services](#).

## AWS AppConfig casi d'uso

AWS AppConfig supporta un ampio spettro di casi d'uso:

- Funzionalità di contrassegni e interruttori: rilascia nuove funzionalità in modo sicuro ai tuoi clienti in un ambiente controllato. Ripristina istantaneamente le modifiche in caso di problemi.
- Ottimizzazione delle applicazioni: introduci con attenzione le modifiche alle applicazioni testando al contempo l'impatto di tali modifiche sugli utenti negli ambienti di produzione.
- Elenco consentito o elenco bloccato: controlla l'accesso alle funzionalità premium o blocca istantaneamente utenti specifici senza distribuire nuovo codice.
- Storage di configurazione centralizzato: mantieni i dati di configurazione organizzati e coerenti su tutti i carichi di lavoro. È possibile utilizzare AWS AppConfig per distribuire i dati di configurazione archiviati nell'archivio di configurazione AWS AppConfig ospitato Gestione dei segreti AWS, Systems Manager Parameter Store o Amazon S3.

# Panoramica dei vantaggi

La seguente breve panoramica delinea i vantaggi dell'utilizzo AWS AppConfig.

Migliora l'efficienza e rilascia le modifiche più velocemente

L'utilizzo di feature flag con nuove funzionalità accelera il processo di rilascio delle modifiche agli ambienti di produzione. Invece di affidarsi a rami di sviluppo di lunga durata che richiedono fusioni complicate prima di un rilascio, i feature flag consentono di scrivere software utilizzando lo sviluppo basato su trunk. I flag di funzionalità consentono di implementare in modo sicuro il codice di pre-release in una pipeline nascosta agli utenti. CI/CD Quando sei pronto per rilasciare le modifiche, puoi aggiornare il flag della funzionalità senza distribuire nuovo codice. Una volta completato il lancio, il flag può ancora funzionare come interruttore a blocchi per disabilitare una nuova funzionalità o funzionalità senza la necessità di ripristinare la distribuzione del codice.

Evita modifiche o guasti non intenzionali con le funzionalità di sicurezza integrate

AWS AppConfig offre le seguenti funzionalità di sicurezza per evitare l'attivazione dei flag di funzionalità o l'aggiornamento dei dati di configurazione che potrebbero causare errori delle applicazioni.

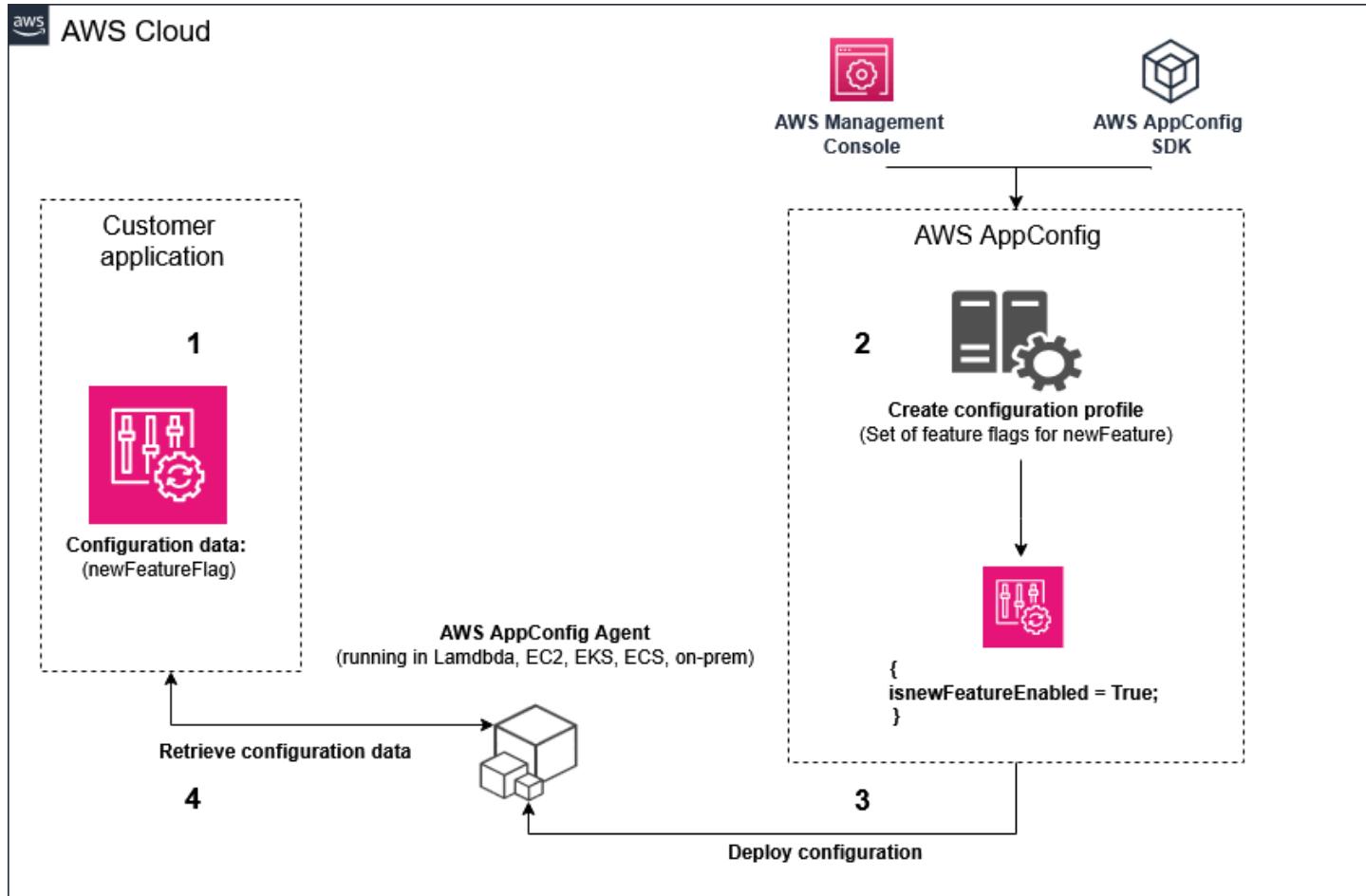
- Validatori: un validatore garantisce che i dati di configurazione siano corretti sintatticamente e semanticamente prima di implementare le modifiche agli ambienti di produzione.
- Strategie di implementazione: una strategia di implementazione consente di rilasciare lentamente le modifiche agli ambienti di produzione nell'arco di minuti o ore.
- Monitoraggio e rollback automatico: AWS AppConfig si integra con Amazon CloudWatch per monitorare le modifiche alle applicazioni. Se l'applicazione non funziona correttamente a causa di una modifica errata della configurazione e tale modifica fa scattare un allarme CloudWatch, ripristina AWS AppConfig automaticamente la modifica per ridurre al minimo l'impatto sugli utenti dell'applicazione.

Implementazioni di feature flag sicure e scalabili

AWS AppConfig si integra con AWS Identity and Access Management (IAM) per fornire un accesso preciso e basato sui ruoli al servizio. AWS AppConfig si integra anche con AWS Key Management Service (AWS KMS) per la crittografia e il controllo. AWS CloudTrail Prima di essere rilasciati ai clienti esterni, tutti i controlli di AWS AppConfig sicurezza sono stati inizialmente sviluppati e convalidati da clienti interni che utilizzano il servizio su larga scala.

# Come funziona AWS AppConfig

Questa sezione fornisce una descrizione dettagliata di come AWS AppConfig funziona.



## 1. Identifica i valori di configurazione nel codice in cui desideri gestire AWS AppConfig

Prima di creare un profilo di configurazione in AWS AppConfig, ti consigliamo di identificare nel codice i dati di configurazione che desideri gestire dinamicamente utilizzando AWS AppConfig. Tra gli esempi più validi vi sono i contrassegni o gli interruttori delle funzionalità, gli elenchi di opzioni consentite e bloccate, la complessità dei log, i limiti del servizio e le regole di limitazione, solo per citarne alcuni. Questi tipi di configurazione cambiano frequentemente e possono causare problemi se non sono corretti.

Se i dati di configurazione esistono già nel cloud, ad esempio in Parameter Store o Amazon S3, puoi sfruttare le funzionalità di AWS AppConfig convalida, distribuzione ed estensione per semplificare ulteriormente la gestione dei dati di configurazione.

## 2. Crea un profilo di configurazione in AWS AppConfig

Un profilo di configurazione include, tra le altre cose, un URI che consente di AWS AppConfig localizzare i dati di configurazione nella posizione archiviata e un tipo di profilo. AWS AppConfig supporta due tipi di profili di configurazione: flag di funzionalità e configurazioni a forma libera. Entrambi i tipi possono ridurre il rischio e la complessità dello sviluppo e della distribuzione del software separando le versioni di funzionalità dalle distribuzioni di codice. Consentono inoltre l'erogazione continua e la mitigazione del rischio attraverso implementazioni graduali. Inoltre, i flag di funzionalità consentono di eseguire test in produzione con utenti reali, mentre le configurazioni in formato libero consentono di recuperare i dati di configurazione da altri servizi. AWS Entrambi i tipi di profilo consentono iterazioni, sperimentazioni, personalizzazioni più rapide e una gestione efficiente del ciclo di vita del software. Per ulteriori informazioni sulla creazione di un profilo di configurazione, vedere. [Creazione di un profilo di configurazione in AWS AppConfig](#)

Un profilo di configurazione può anche includere validatori opzionali per garantire che i dati di configurazione siano corretti dal punto di vista sintattico e semantico. AWS AppConfig esegue un controllo utilizzando i validatori quando si avvia una distribuzione. Se vengono rilevati errori, la distribuzione torna ai dati di configurazione precedenti.

Quando si crea un profilo di configurazione, si crea anche un'applicazione in AWS AppConfig. Un'applicazione è semplicemente uno spazio dei nomi o un costrutto organizzativo come una cartella.

## 3. Distribuisci i dati di configurazione

Quando avvii una distribuzione, AWS AppConfig esegue le seguenti attività:

1. Recupera i dati di configurazione dal data store sottostante utilizzando il nome del percorso di posizione nel profilo di configurazione.
2. Verifica che i dati di configurazione siano corretti dal punto di vista sintattico e semantico utilizzando i validatori specificati al momento della creazione del profilo di configurazione.
3. Invia una copia dei dati ad AWS AppConfig Agent per essere letti dall'applicazione. Questa copia è denominata dati distribuiti.

Per ulteriori informazioni sulla distribuzione di una configurazione, vedere. [Distribuzione di flag di funzionalità e dati di configurazione in AWS AppConfig](#)

## 4. Recuperare la configurazione

Per recuperare i dati, l'applicazione effettua una chiamata HTTP al server localhost in cui AWS AppConfig Agent ha memorizzato nella cache una copia locale dei dati di configurazione distribuiti. Il recupero dei dati è un evento misurato. AWS AppConfig L'agente supporta diversi casi d'uso, come descritto in. [Come utilizzare AWS AppConfig Agent per recuperare i dati di configurazione](#)

Se AWS AppConfig Agent non è supportato per il tuo caso d'uso, puoi configurare l'applicazione AWS AppConfig per verificare la presenza di aggiornamenti di configurazione richiamando direttamente le azioni [StartConfigurationSession](#) l'[GetLatestConfiguration](#)API.

Per ulteriori informazioni sul recupero di una configurazione, consulta. [Recupero dei flag delle funzionalità e dei dati di configurazione in AWS AppConfig](#)

## Prezzi per AWS AppConfig

Il prezzo AWS AppConfig si pay-as-you-go basa sui dati di configurazione e sul recupero dei flag di funzionalità. Si consiglia di utilizzare l' AWS AppConfig agente per ottimizzare i costi. Per ulteriori informazioni, consultare [AWS Systems Manager Prezzi](#).

## AWS AppConfig quote

È possibile visualizzare informazioni sugli AWS AppConfig endpoint e sulle quote di servizio in. [Riferimenti generali di Amazon Web Services](#)

 Note

AWS AppConfig è una capacità di. AWS Systems Manager

Per informazioni sulle quote per i servizi che memorizzano AWS AppConfig le configurazioni, vedere. [Comprensione delle quote e delle limitazioni dell'archivio di configurazione](#)

## Risorse aggiuntive

Le seguenti risorse possono aiutarti a saperne di più su. AWS AppConfig

## Blog

I seguenti blog possono aiutarti a saperne di più sulle AWS AppConfig sue funzionalità:

- [Perché dovresti usare AWS AppConfig](#)
- [Scatena la potenza delle bandiere di funzionalità con AWS AppConfig](#)
- [Utilizzo dei flag di funzionalità AWS AppConfig](#)
- [Le migliori pratiche per la convalida dei flag di AWS AppConfig funzionalità e dei dati di configurazione](#)

## SDKs

Per informazioni sulle AWS AppConfig specifiche lingue, consulta le seguenti risorse SDKs:

- [AWS Command Line Interface](#)
- [AWS SDK per .NET](#)
- [AWS SDK per C++](#)
- [AWS SDK for Go](#)
- [AWS SDK per Java V2](#)
- [AWS SDK per JavaScript](#)
- [AWS SDK per PHP V3](#)
- [AWS SDK per Python](#)
- [AWS SDK per Ruby V3](#)

# Configurazione AWS AppConfig

Se non l'hai già fatto, registrati Account AWS e crea un utente amministrativo.

## Iscriviti per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la <https://portal.aws.amazon.com/billing/registrazione>.
2. Segui le istruzioni online.

Nel corso della procedura di registrazione riceverai una telefonata o un messaggio di testo e ti verrà chiesto di inserire un codice di verifica attraverso la tastiera del telefono.

Quando ti iscrivi a un Account AWS, Utente root dell'account AWS viene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire le [attività che richiedono l'accesso di un utente root](#).

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a <https://aws.amazon.com/> e scegliendo Il mio account.

## Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi [Console di gestione AWS](#) come proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

Per informazioni sull'accesso utilizzando un utente root, consulta la pagina [Signing in as the root user](#) della Guida per l'utente di Accedi ad AWS .

## 2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

Per istruzioni, consulta [Abilitare un dispositivo MFA virtuale per l'utente Account AWS root \(console\)](#) nella Guida per l'utente IAM.

### Crea un utente con accesso amministrativo

#### 1. Abilita il Centro identità IAM.

Per istruzioni, consulta [Abilitazione del AWS IAM Identity Center](#) nella Guida per l'utente di AWS IAM Identity Center .

#### 2. Nel Centro identità IAM, assegna l'accesso amministrativo a un utente.

Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta [Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory](#) nella Guida per l'AWS IAM Identity Center utente.

### Accesso come utente amministratore

- Per accedere come utente del Centro identità IAM, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente del Centro identità IAM.

Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta [AWS Accedere al portale di accesso](#) nella Guida per l'Accedi ad AWS utente.

### Assegnazione dell'accesso ad altri utenti

#### 1. Nel Centro identità IAM, crea un set di autorizzazioni conforme alla best practice per l'applicazione di autorizzazioni con il privilegio minimo.

Segui le istruzioni riportate nella pagina [Creazione di un set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center .

#### 2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

Per istruzioni, consulta [Aggiungere gruppi](#) nella Guida per l'utente di AWS IAM Identity Center .

## Concessione dell'accesso programmatico

Gli utenti hanno bisogno di un accesso programmatico se vogliono interagire con l'AWS Console di gestione AWS esterno di. Il modo per concedere l'accesso programmatico dipende dal tipo di utente che accede a AWS.

Per fornire agli utenti l'accesso programmatico, scegli una delle seguenti opzioni.

Quale utente necessita dell'accesso programmatico?	Per	Come
IAM	(Consigliato) Utilizza le credenziali della console come credenziali temporanee per firmare le richieste programmatiche a,, o. AWS CLI AWS SDKs AWS APIs	<p>Segui le istruzioni per l'interfaccia che desideri utilizzare.</p> <ul style="list-style-type: none"> <li>• Per la AWS CLI, consulta <a href="#">Login for AWS local development</a> nella Guida per l'AWS Command Line Interface utente.</li> <li>• Per AWS SDKs, consulta <a href="#">Login for AWS local development</a> nella AWS SDKs and Tools Reference Guide.</li> </ul>
Identità della forza lavoro (Utenti gestiti nel centro identità IAM)	Utilizza credenziali temporanee per firmare le richieste programmatiche a AWS CLI, AWS SDKs, o. AWS APIs	<p>Segui le istruzioni per l'interfaccia che desideri utilizzare.</p> <ul style="list-style-type: none"> <li>• Per la AWS CLI, vedere <a href="#">Configurazione dell'uso AWS IAM Identity Center nella AWS CLI Guida per l'utente</a>. AWS Command Line Interface</li> <li>• Per AWS SDKs gli strumenti e AWS APIs, consulta <a href="#">l'autenticazione di IAM Identity Center</a> nella Guida</li> </ul>

Quale utente necessita dell'accesso programmatico?	Per	Come
		di riferimento AWS SDKs and Tools.
IAM	Utilizza credenziali temporane e per firmare le richieste programmatiche a AWS CLI, AWS SDKs, o. AWS APIs	Seguendo le istruzioni riportate in <a href="#">Utilizzo delle credenziali temporanee con le AWS risorse nella Guida per l'utente IAM</a> .
IAM	(Non consigliato) Utilizza credenziali a lungo termine per firmare richieste programmatiche a AWS CLI, AWS SDKs o. AWS APIs	<p>Segui le istruzioni per l'interfaccia che desideri utilizzare.</p> <ul style="list-style-type: none"> <li>• Per la AWS CLI, consulta <a href="#">Autenticazione tramite credenziali utente IAM nella Guida per l'utente AWS Command Line Interface</a></li> <li>• Per gli strumenti AWS SDKs e gli strumenti, consulta <a href="#">Autenticazione tramite credenziali a lungo termine</a> nella Guida di riferimento agli strumenti e agli AWS SDKs strumenti.</li> <li>• Per AWS APIs, consulta la sezione <a href="#">Gestione delle chiavi di accesso per gli utenti IAM</a> nella Guida per l'utente IAM.</li> </ul>

## Comprendere il supporto IPv6

Tutte le API di AppConfig supportano le chiamate IPv4 e IPv6 complete.

## Piano di controllo APIs

Utilizzate il seguente endpoint IPv4 e le chiamate IPv6 dual-stack al piano di controllo:

appconfig.*Region*.api.aws

Ad esempio: appconfig.us-east-1.api.aws

Solo per questo, usa il seguente URL: IPv4

appconfig.*Region*.amazonaws.com

## Piano dati APIs

Per le chiamate dual-stack al piano dati, utilizzate il seguente endpoint:

appconfigdata.*Region*.api.aws

Ad esempio: appconfig.us-east-1.api.aws

Solo per questo, usa il seguente URL: IPv4

appconfigdata.*Region*.amazonaws.com

### Note

Per ulteriori informazioni, consulta [Endpoint e quote AWS AppConfig](#) nella Riferimenti generali di AWS.

## Configura le autorizzazioni per il rollback automatico

Puoi configurare AWS AppConfig il ripristino a una versione precedente di una configurazione in risposta a uno o più CloudWatch allarmi Amazon. Quando configuri una distribuzione per rispondere agli CloudWatch allarmi, specifichi un ruolo AWS Identity and Access Management (IAM). AWS AppConfig richiede questo ruolo in modo da poter monitorare gli CloudWatch allarmi. Questa procedura è facoltativa, ma altamente consigliata.

### Note

Osservare le seguenti informazioni.

- Il ruolo IAM deve appartenere all'account corrente. Per impostazione predefinita, AWS AppConfig può monitorare solo gli allarmi di proprietà dell'account corrente.
- Per informazioni sulle metriche da monitorare e su come configurare AWS AppConfig il rollback automatico, consulta [Monitoraggio delle implementazioni per il rollback automatico](#)

Utilizza le seguenti procedure per creare un ruolo IAM che AWS AppConfig consenta il rollback in base agli allarmi. CloudWatch Questa sezione include le seguenti procedure.

1. [Passaggio 1: creare la politica di autorizzazione per il rollback in base agli allarmi CloudWatch](#)
2. [Passaggio 2: crea il ruolo IAM per il rollback in base CloudWatch agli allarmi](#)
3. [Fase 3: aggiunta di una relazione di trust](#)

## Passaggio 1: creare la politica di autorizzazione per il rollback in base agli allarmi CloudWatch

Utilizza la seguente procedura per creare una policy IAM che AWS AppConfig autorizzi a richiamare l'azione dell'`DescribeAlarms`API.

Per creare una politica di autorizzazione IAM per il rollback basata sugli allarmi CloudWatch

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione, scegli Policy e Crea policy.
3. Nella pagina Crea policy, scegli la scheda JSON.
4. Sostituisci il contenuto predefinito nella scheda JSON con la seguente politica di autorizzazione, quindi scegli Avanti: Tag.

### Note

Per restituire informazioni sugli allarmi CloudWatch compositi, all'operazione `DescribeAlarms`API devono essere assegnate \* le autorizzazioni, come mostrato

qui. Non è possibile restituire informazioni sugli allarmi compositi se l'ambito `DescribeAlarms` è più ristretto.

## JSON

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "cloudwatch:DescribeAlarms"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

5. Immettere i tag per questo ruolo, quindi scegliere Next: Review (Successivo: Revisione).
6. Nella pagina Revisione, inserisci il **SSMCLOUDWATCHALARMDISCOVERYPOLICY** campo Nome.
7. Scegli Crea policy. Il sistema visualizza di nuovo la pagina Policies (Policy).

## Passaggio 2: crea il ruolo IAM per il rollback in base CloudWatch agli allarmi

Utilizza la procedura seguente per creare un ruolo IAM e assegnargli la policy creata nella procedura precedente.

Per creare un ruolo IAM per il rollback basato sugli allarmi CloudWatch

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione, scegliere Roles (Ruoli) e quindi Create role (Crea ruolo).
3. In Select type of trusted entity (Seleziona tipo di entità attendibile), scegliere AWS service (Servizio).
4. Immediatamente in Scegli il servizio che utilizzerà questo ruolo, scegli EC2: Consente alle EC2 istanze di chiamare AWS i servizi per tuo conto, quindi scegli Avanti: Autorizzazioni.
5. Nella pagina Politica sulle autorizzazioni indicate, cerca. **SSMCLOUDWATCHALARMDISCOVERYPOLICY**

6. Scegliere questa policy, quindi selezionare Next: Tags (Successivo: tag).
7. Immettere i tag per questo ruolo, quindi scegliere Next: Review (Successivo: Revisione).
8. Nella pagina Crea ruolo, inserisci **SSMCLOUDWATCHALARMDiscoveryRole** il campo Nome ruolo, quindi scegli Crea ruolo.
9. Nella pagina Roles (Ruoli), scegliere il ruolo appena creato. Viene visualizzata la pagina Riepilogo.

## Fase 3: aggiunta di una relazione di trust

Utilizza la procedura seguente per configurare il ruolo appena creato per considerare attendibile AWS AppConfig.

Per aggiungere una relazione di fiducia per AWS AppConfig

1. Nella pagina Summary (Riepilogo) per il ruolo creato in precedenza, scegliere la scheda Trust Relationships (Relazioni di trust), quindi scegliere Edit Trust Relationship (Modifica relazione di trust).
2. Modificare il criterio per includere solo "appconfig.amazonaws.com", come mostrato nell'esempio seguente:

JSON

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

3. Scegli Update Trust Policy (Aggiorna policy di trust).

# Creazione di flag di funzionalità e dati di configurazione in formato libero in AWS AppConfig

Gli argomenti di questa sezione consentono di completare le seguenti attività in AWS AppConfig. Queste attività creano artefatti importanti per la distribuzione dei dati di configurazione.

## 1. [Crea uno spazio dei nomi dell'applicazione](#)

Per creare uno spazio dei nomi di un'applicazione, si crea un AWS AppConfig artefatto chiamato applicazione. Un'applicazione è semplicemente un costrutto organizzativo come una cartella.

## 2. [Crea ambienti](#)

Per ogni AWS AppConfig applicazione, si definiscono uno o più ambienti. Un ambiente è un gruppo di AWS AppConfig destinazione di distribuzione logica, ad esempio le applicazioni in un Production ambiente Beta OR. È inoltre possibile definire ambienti per i sottocomponenti delle applicazioni, come AWS Lambda functions, Containers, WebMobile, e Back-end.

Puoi configurare gli CloudWatch allarmi Amazon per ogni ambiente per ripristinare automaticamente le modifiche di configurazione problematiche. Il sistema monitora gli allarmi durante una distribuzione della configurazione. Se viene attivato un allarme, il sistema ripristina la configurazione.

## 3. [Crea un profilo di configurazione](#)

I dati di configurazione sono una raccolta di impostazioni che influenzano il comportamento dell'applicazione. Un profilo di configurazione include, tra le altre cose, un URI che consente di AWS AppConfig localizzare i dati di configurazione nella posizione archiviata e un tipo di configurazione. AWS AppConfig supporta i seguenti tipi di profili di configurazione:

- Contrassegni di funzionalità: è possibile utilizzare i flag di funzionalità per abilitare o disabilitare le funzionalità all'interno delle applicazioni o per configurare caratteristiche diverse delle funzionalità dell'applicazione utilizzando gli attributi dei flag. AWS AppConfig memorizza le configurazioni dei feature flag nell'archivio di configurazione AWS AppConfig ospitato in un formato di feature flag che contiene dati e metadati sui flag e sugli attributi dei flag. L'URI per le configurazioni dei feature flag è semplice. hosted
- Configurazioni in formato libero: una configurazione a forma libera può archiviare dati in uno dei seguenti strumenti e in uno qualsiasi degli strumenti di Systems Servizi AWS Manager seguenti:
  - AWS AppConfig archivio di configurazione ospitato

- Amazon Simple Storage Service
- AWS CodePipeline
- Gestione dei segreti AWS
- AWS Systems Manager (SSM) Parameter Store
- Archivio documenti SSM

 Note

Se possibile, consigliamo di ospitare i dati di configurazione nell'archivio di configurazione AWS AppConfig ospitato in quanto offre la maggior parte delle funzionalità e dei miglioramenti.

#### 4. (Facoltativo, ma consigliato) [Crea flag di funzionalità con più varianti](#)

AWS AppConfig offre flag di funzionalità di base, che (se abilitati) restituiscono un set specifico di dati di configurazione per richiesta. Per supportare meglio i casi d'uso di segmentazione degli utenti e suddivisione del traffico, offre AWS AppConfig anche flag di funzionalità multivarianti, che consentono di definire una serie di possibili valori di flag da restituire per una richiesta. È inoltre possibile configurare diversi stati (abilitati o disabilitati) per i flag multivarianti. Quando si richiede un flag configurato con varianti, l'applicazione fornisce un contesto che AWS AppConfig valuta in base a una serie di regole definite dall'utente. A seconda del contesto specificato nella richiesta e delle regole definite per la variante, AWS AppConfig restituisce valori di flag diversi all'applicazione.

#### Argomenti

- [Comprensione del ruolo IAM del profilo di configurazione](#)
- [Creazione di uno spazio dei nomi per l'applicazione in AWS AppConfig](#)
- [Creazione di ambienti per l'applicazione in AWS AppConfig](#)
- [Creazione di un profilo di configurazione in AWS AppConfig](#)

## Comprensione del ruolo IAM del profilo di configurazione

Puoi creare il ruolo IAM che fornisce l'accesso ai dati di configurazione utilizzando AWS AppConfig. Oppure puoi creare tu stesso il ruolo IAM. Se crei il ruolo utilizzando AWS AppConfig, il sistema

crea il ruolo e specifica una delle seguenti politiche di autorizzazione, a seconda del tipo di origine di configurazione scelta.

L'origine della configurazione è un segreto di Secrets Manager

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "secretsmanager:GetSecretValue"  
      ],  
      "Resource": [  
        "arn:aws:secretsmanager:us-  
east-1:111122223333:secret:secret_name-a1b2c3"  
      ]  
    }  
  ]  
}
```

L'origine della configurazione è un parametro Parameter Store

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "ssm:GetParameter"  
      ],  
      "Resource": [  
        "arn:aws:ssm:us-east-1:111122223333:parameter/parameter_name"  
      ]  
    }  
  ]  
}
```

}

L'origine di configurazione è un documento SSM

JSON

{

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "ssm:GetDocument"
        ],
        "Resource": [
            "arn:aws:ssm:us-east-1:111122223333:document/document_name"
        ]
    }
]
```

}

Se si crea il ruolo utilizzando AWS AppConfig, il sistema crea anche la seguente relazione di trust per il ruolo.

JSON

{

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

}

## Creazione di uno spazio dei nomi per l'applicazione in AWS AppConfig

Le procedure in questa sezione consentono di creare un AWS AppConfig artefatto chiamato applicazione. Un'applicazione è semplicemente un costrutto organizzativo come una cartella che identifica lo spazio dei nomi dell'applicazione. Questo costrutto organizzativo ha una relazione con alcune unità di codice eseguibile. Ad esempio, è possibile creare un'applicazione chiamata MyMobileApp per organizzare e gestire i dati di configurazione per un'applicazione mobile installata dagli utenti. È necessario creare questi artefatti prima di poterli utilizzare AWS AppConfig per distribuire e recuperare i flag di funzionalità o i dati di configurazione in formato libero.

La procedura seguente offre la possibilità di associare un'estensione a un profilo di configurazione dei feature flag. Un'estensione aumenta la capacità di inserire logica o comportamento in punti diversi durante il AWS AppConfig flusso di lavoro di creazione o distribuzione di una configurazione. Per ulteriori informazioni, consulta [Comprendere AWS AppConfig le estensioni](#).

### Note

È possibile utilizzarla AWS CloudFormation per creare AWS AppConfig artefatti, tra cui applicazioni, ambienti, profili di configurazione, distribuzioni, strategie di distribuzione e versioni di configurazione ospitate. Per ulteriori informazioni, consulta [Riferimento al tipo di risorsa AWS AppConfig](#) nella Guida dell'utente di AWS CloudFormation .

### Argomenti

- [Creazione di un' AWS AppConfig applicazione \(console\)](#)
- [Creazione di un' AWS AppConfig applicazione \(riga di comando\)](#)

## Creazione di un' AWS AppConfig applicazione (console)

Utilizzare la procedura seguente per creare un' AWS AppConfig applicazione utilizzando la AWS Systems Manager console.

## Come creare un'applicazione

1. Apri la AWS Systems Manager console all'indirizzo <https://console.aws.amazon.com/systems-manager/appconfig/>.
2. Nel riquadro di navigazione, scegli Applications (Applicazioni), quindi Create application (Crea applicazione).
3. In Name (Nome), immettere un nome per l'applicazione.
4. In Description (Descrizione), immettere le informazioni sull'applicazione.
5. (Facoltativo) Nella sezione Estensioni, scegli un'estensione dall'elenco. Per ulteriori informazioni, consulta [Comprendere AWS AppConfig le estensioni](#).
6. (Facoltativo) Nella sezione Tag, inserisci una chiave e un valore opzionale. È possibile specificare un massimo di 50 tag per una risorsa.
7. Scegli Crea applicazione.

AWS AppConfig crea l'applicazione e quindi visualizza la scheda Ambienti. Passa a [Creazione di ambienti per l'applicazione in AWS AppConfig](#).

## Creazione di un' AWS AppConfig applicazione (riga di comando)

La procedura seguente descrive come utilizzare AWS CLI (su Linux o Windows) o AWS Strumenti per PowerShell creare un' AWS AppConfig applicazione.

Per creare un'applicazione passo dopo passo

1. Apri il AWS CLI.
2. Esegui il comando seguente per creare un'applicazione.

Linux

```
aws appconfig create-application \
--name A_name_for_the_application \
--description A_description_of_the_application \
--tags User_defined_key_value_pair_metadata_for_the_application
```

Windows

```
aws appconfig create-application ^
--name A_name_for_the_application ^
```

```
--description A_description_of_the_application ^
--tags User_defined_key_value_pair_metadata_for_the_application
```

## PowerShell

```
New-APPCApplication ^
-Name Name_for_the_application ^
-Description Description_of_the_application ^
-Tag Hashtable_type_user_defined_key_value_pair_metadata_for_the_application
```

Il sistema restituisce informazioni simili alle seguenti.

## Linux

```
{
  "Id": "Application ID",
  "Name": "Application name",
  "Description": "Description of the application"
}
```

## Windows

```
{
  "Id": "Application ID",
  "Name": "Application name",
  "Description": "Description of the application"
}
```

## PowerShell

```
ContentSize      : Runtime of the command
Description      : Description of the application
HttpStatus      : HTTP Status of the runtime
Id              : Application ID
Name            : Application name
ResponseMetadata : Runtime Metadata
```

# Creazione di ambienti per l'applicazione in AWS AppConfig

Per ogni AWS AppConfig applicazione, definisci uno o più ambienti. Un ambiente è un gruppo di AppConfig obiettivi di distribuzione logico, ad esempio applicazioni in un Production ambiente Beta OR, AWS Lambda funzioni o contenitori. È inoltre possibile definire ambienti per i sottocomponenti delle applicazioni, ad esempio WebMobile, eBack-end. Puoi configurare gli CloudWatch allarmi Amazon per ogni ambiente. Il sistema monitora gli allarmi durante una distribuzione della configurazione. Se viene attivato un allarme, il sistema ripristina la configurazione.

## Prima di iniziare

Se desideri abilitare AWS AppConfig il ripristino di una configurazione in risposta a un CloudWatch allarme, devi configurare un ruolo AWS Identity and Access Management (IAM) con autorizzazioni per consentire AWS AppConfig la risposta agli allarmi. CloudWatch È possibile scegliere questo ruolo nella procedura seguente. Per ulteriori informazioni, consulta [Configura le autorizzazioni per il rollback automatico](#).

## Argomenti

- [Creazione di un AWS AppConfig ambiente \(console\)](#)
- [Creazione di un AWS AppConfig ambiente \(riga di comando\)](#)

## Creazione di un AWS AppConfig ambiente (console)

Utilizzare la procedura seguente per creare un AWS AppConfig ambiente utilizzando la AWS Systems Manager console.

### Come creare un ambiente

1. Apri la AWS Systems Manager console all'indirizzo <https://console.aws.amazon.com/systems-manager/appconfig/>.
2. Nel riquadro di navigazione, scegli Applicazioni, quindi scegli il nome di un'applicazione per aprire la pagina dei dettagli.
3. Scegli la scheda Ambienti, quindi scegli Crea ambiente.
4. In Name (Nome), inserire un nome per l'ambiente.
5. In Description (Descrizione), immettere le informazioni sull'ambiente.

6. (Facoltativo) Nella sezione Monitor, scegli il campo del ruolo IAM, quindi scegli un ruolo IAM con l'autorizzazione a richiamare `cloudwatch:DescribeAlarms` le metriche che desideri monitorare per rilevare gli allarmi.
7. Nell'elenco degli CloudWatch allarmi, inserisci Amazon Resource Names (ARNs) una o più metriche da monitorare. AWS AppConfig ripristina la distribuzione della configurazione se uno di questi parametri entra in uno stato. ALARM Per informazioni sulle metriche consigliate, consulta [Monitoraggio delle implementazioni per il rollback automatico](#)
8. (Facoltativo) Nella sezione Associa estensioni, scegli un'estensione dall'elenco. Per ulteriori informazioni, consulta [Comprendere AWS AppConfig le estensioni](#).
9. (Facoltativo) Nella sezione Tag, inserisci una chiave e un valore opzionale. È possibile specificare un massimo di 50 tag per una risorsa.
10. Seleziona Create environment (Crea ambiente).

AWS AppConfig crea l'ambiente e quindi visualizza la pagina dei dettagli dell'ambiente. Passa a [Creazione di un profilo di configurazione in AWS AppConfig](#).

## Creazione di un AWS AppConfig ambiente (riga di comando)

La procedura seguente descrive come utilizzare AWS CLI (su Linux o Windows) o AWS Strumenti per PowerShell creare un AWS AppConfig ambiente.

Creare un ambiente passo dopo passo

1. Apri il AWS CLI.
2. Esegui il comando seguente per creare un ambiente.

Linux

```
aws appconfig create-environment \
--application-id The_application_ID \
--name A_name_for_the_environment \
--description A_description_of_the_environment \
--monitors
"AlarmArn=ARN_of_the_Amazon_CloudWatch_alarm,AlarmArnRole=ARN_of_the_IAM
role_for_AWS_AppConfig_to_monitor_AlarmArn" \
--tags User_defined_key_value_pair_metadata_of_the_environment
```

## Windows

```
aws appconfig create-environment ^
--application-id The_application_ID ^
--name A_name_for_the_environment ^
--description A_description_of_the_environment ^
--monitors
"AlarmArn=ARN_of_the_Amazon_CloudWatch_alarm,AlarmArnRole=ARN_of_the_IAM
role_for_AWS_AppConfig_to_monitor_AlarmArn" ^
--tags User_defined_key_value_pair_metadata_of_the_environment
```

## PowerShell

```
New-APPCEnvironment ` 
-Name Name_for_the_environment ` 
-ApplicationId The_application_ID ` 
-Description Description_of_the_environment ` 
-Monitors
@{` 
"AlarmArn=ARN_of_the_Amazon_CloudWatch_alarm,AlarmArnRole=ARN_of_the_IAM
role_for_AWS_AppConfig_to_monitor_AlarmArn"} ` 
-Tag Hashtable_type_user_defined_key_value_pair_metadata_of_the_environment
```

Il sistema restituisce informazioni simili alle seguenti.

## Linux

```
{
  "ApplicationId": "The application ID",
  "Id": "The_environment_ID",
  "Name": "Name of the environment",
  "State": "The state of the environment",
  "Description": "Description of the environment",

  "Monitors": [
    {
      "AlarmArn": "ARN of the Amazon CloudWatch alarm",
      "AlarmRoleArn": "ARN of the IAM role for AppConfig to monitor AlarmArn"
    }
  ]
}
```

## Windows

```
{  
    "ApplicationId": "The application ID",  
    "Id": "The environment ID",  
    "Name": "Name of the environment",  
    "State": "The state of the environment"  
    "Description": "Description of the environment",  
  
    "Monitors": [  
        {  
            "AlarmArn": "ARN of the Amazon CloudWatch alarm",  
            "AlarmRoleArn": "ARN of the IAM role for AppConfig to monitor AlarmArn"  
        }  
    ]  
}
```

## PowerShell

ApplicationId	: The application ID
ContentLength	: Runtime of the command
Description	: Description of the environment
HttpStatusCode	: HTTP Status of the runtime
Id	: The environment ID
Monitors	: {ARN of the Amazon CloudWatch alarm, ARN of the IAM role for AppConfig to monitor AlarmArn}
Name	: Name of the environment
Response Metadata	: Runtime Metadata
State	: State of the environment

Passa a [Creazione di un profilo di configurazione in AWS AppConfig](#).

## Creazione di un profilo di configurazione in AWS AppConfig

I dati di configurazione sono una raccolta di impostazioni che influenzano il comportamento dell'applicazione. Un profilo di configurazione include, tra le altre cose, un URI che consente di AWS AppConfig localizzare i dati di configurazione nella posizione archiviata e un tipo di configurazione. AWS AppConfig supporta i seguenti tipi di profili di configurazione:

- Contrassegni di funzionalità: è possibile utilizzare i flag di funzionalità per abilitare o disabilitare le funzionalità all'interno delle applicazioni o per configurare caratteristiche diverse delle funzionalità dell'applicazione utilizzando gli attributi dei flag. AWS AppConfig memorizza le configurazioni dei feature flag nell'archivio di configurazione AWS AppConfig ospitato in un formato di feature flag che contiene dati e metadati sui flag e sugli attributi dei flag. L'URI per le configurazioni dei feature flag è semplice. hosted
- Configurazioni in formato libero: una configurazione a forma libera può archiviare dati in uno dei seguenti strumenti e in uno qualsiasi degli strumenti di Systems Servizi AWS Manager seguenti:
  - AWS AppConfig archivio di configurazione ospitato
  - Amazon Simple Storage Service
  - AWS CodePipeline
  - Gestione dei segreti AWS
  - AWS Systems Manager (SSM) Parameter Store
  - Archivio documenti SSM

#### Note

Se possibile, consigliamo di ospitare i dati di configurazione nell'archivio di configurazione AWS AppConfig ospitato in quanto offre la maggior parte delle funzionalità e dei miglioramenti.

Ecco alcuni esempi di dati di configurazione per aiutarti a comprendere meglio i diversi tipi di dati di configurazione e come possono essere utilizzati in un feature flag o senza un profilo di configurazione.

#### Dati di configurazione dei flag di funzionalità

I seguenti dati di configurazione dei flag di funzionalità abilitano o disabilitano i pagamenti mobili e i pagamenti predefiniti in base alla regione.

#### JSON

```
{  
  "allow_mobile_payments": {  
    "enabled": false  
  },  
}
```

```
"default_payments_per_region": {  
    "enabled": true  
}  
}
```

## YAML

```
---  
allow_mobile_payments:  
    enabled: false  
default_payments_per_region:  
    enabled: true
```

## Dati di configurazione operativa

I seguenti dati di configurazione in formato libero impongono limiti al modo in cui un'applicazione elabora le richieste.

## JSON

```
{  
    "throttle-limits": {  
        "enabled": "true",  
        "throttles": [  
            {  
                "simultaneous_connections": 12  
            },  
            {  
                "tps_maximum": 5000  
            }  
        ],  
        "limit-background-tasks": [  
            true  
        ]  
    }  
}
```

## YAML

```
---  
throttle-limits:  
    enabled: 'true'
```

```
throttles:
- simultaneous_connections: 12
- tps_maximum: 5000
limit-background-tasks:
- true
```

## Accesso ai dati di configurazione della lista di controllo

I seguenti dati di configurazione in formato libero dell'elenco di controllo degli accessi specificano quali utenti o gruppi possono accedere a un'applicazione.

### JSON

```
{
  "allow-list": {
    "enabled": "true",
    "cohorts": [
      {
        "internal_employees": true
      },
      {
        "beta_group": false
      },
      {
        "recent_new_customers": false
      },
      {
        "user_name": "Jane_Doe"
      },
      {
        "user_name": "John_Doe"
      }
    ]
  }
}
```

### YAML

```
---
allow-list:
  enabled: 'true'
  cohorts:
```

```
- internal_employees: true
- beta_group: false
- recent_new_customers: false
- user_name: Jane_Doe
- user_name: Ashok_Kumar
```

## Argomenti

- [Creazione di un profilo di configurazione del feature flag in AWS AppConfig](#)
- [Creazione di un profilo di configurazione in formato libero in AWS AppConfig](#)
- [Creazione di un profilo di configurazione per fonti di dati non native](#)

## Creazione di un profilo di configurazione del feature flag in AWS AppConfig

È possibile utilizzare i flag di funzionalità per abilitare o disabilitare funzionalità all'interno delle applicazioni o per configurare caratteristiche diverse delle funzionalità dell'applicazione utilizzando gli attributi dei flag. AWS AppConfig archivia le configurazioni dei flag di funzionalità nell'archivio di configurazione AWS AppConfig ospitato in un formato di feature flag che contiene dati e metadati sui flag e sugli attributi dei flag.

### Note

Quando create un profilo di configurazione dei feature flag, potete creare un feature flag di base come parte del flusso di lavoro del profilo di configurazione. AWS AppConfig supporta anche i flag di funzionalità multivarianti. I flag di funzionalità multivarianti consentono di definire una serie di possibili valori di flag da restituire per una richiesta. Quando si richiede un flag configurato con varianti, l'applicazione fornisce un contesto che AWS AppConfig valuta in base a una serie di regole definite dall'utente. A seconda del contesto specificato nella richiesta e delle regole definite per la variante, AWS AppConfig restituisce valori di flag diversi all'applicazione.

Per creare flag di funzionalità multivarianti, create prima un profilo di configurazione, quindi modificate tutti i flag all'interno del profilo di configurazione per aggiungere varianti. Per ulteriori informazioni, consulta [Creazione di flag di funzionalità multivarianti](#).

## Argomenti

- [Comprendere degli attributi dei feature flag](#)

- [Creazione di un profilo di configurazione del feature flag \(console\)](#)
- [Creazione di un profilo di configurazione del feature flag \(riga di comando\)](#)
- [Creazione di flag di funzionalità multivarianti](#)
- [Comprendere del tipo di riferimento per AWS.AppConfig.FeatureFlags](#)
- [Salvataggio di una versione precedente di Feature Flag in una nuova versione](#)

## Comprendere degli attributi dei feature flag

Quando create un profilo di configurazione del feature flag o create un nuovo flag all'interno di un profilo di configurazione esistente, potete specificare gli attributi e i vincoli corrispondenti per il flag. Un attributo è un campo che si associa al feature flag per esprimere le proprietà relative al feature flag. Gli attributi vengono forniti all'applicazione con la chiave flag e il valore enable o del flag.

I vincoli assicurano che nell'applicazione non vengano distribuiti valori di attributo imprevisti. Un esempio è illustrato nell'immagine seguente.

### Define attributes

Key	Type	Constraint
currency	String	CAD,USD,MXN

Required  Regular expression  Enum

[Add new attribute](#)

---

### Attribute Values

Key	Key
currency	CAD

[Cancel](#) [Apply](#)

### Note

Notate le seguenti informazioni sugli attributi dei flag.

- Per i nomi degli attributi, la parola «abilitato» è riservata. Non è possibile creare un attributo di feature flag chiamato «enabled». Non ci sono altre parole riservate.
- Gli attributi di un flag di funzionalità sono inclusi nella `GetLatestConfiguration` risposta solo se tale flag è abilitato.
- Le chiavi degli attributi Flag per un determinato flag devono essere univoche.

AWS AppConfig supporta i seguenti tipi di attributi di bandiera e i vincoli corrispondenti.

Tipo	Vincolo	Description
Stringa	Espressione regolare	Modello Regex per la stringa
	Enum	Elenco di valori accettabili per la stringa
Numero	Minimo	Valore numerico minimo per l'attributo
	Massimo	Valore numerico massimo per l'attributo
Booleano	Nessuno	Nessuno
Matrice di stringhe	Espressione regolare	Modello Regex per gli elementi dell'array
	Enum	Elenco di valori accettabili per gli elementi dell'array
Matrice numerica	Minimo	Valore numerico minimo per gli elementi dell'array
	Massimo	Valore numerico massimo per gli elementi dell'array

## Creazione di un profilo di configurazione del feature flag (console)

Utilizzate la seguente procedura per creare un profilo di configurazione dei AWS AppConfig feature flag utilizzando la AWS AppConfig console. Al momento della creazione del profilo di configurazione, è anche possibile creare un feature flag di base.

### Come creare un profilo di configurazione

1. Apri la AWS Systems Manager console all'indirizzo <https://console.aws.amazon.com/systems-manager/appconfig/>.
2. Nel riquadro di navigazione, scegli Applicazioni, quindi scegli un'applicazione in cui hai creato. [Creazione di uno spazio dei nomi per l'applicazione in AWS AppConfig](#)
3. Nella scheda Profili di configurazione e bandiere di funzionalità, scegli Crea configurazione.
4. Nella sezione Opzioni di configurazione, scegliete Feature flag.
5. Nella sezione Profilo di configurazione, per Nome del profilo di configurazione, inserisci un nome.
6. (Facoltativo) Espandi Descrizione e inserisci una descrizione.
7. (Facoltativo) Espandi Opzioni aggiuntive e completa quanto segue, se necessario.
  - a. Nell'elenco Crittografia, scegliete una chiave AWS Key Management Service (AWS KMS) dall'elenco. Questa chiave gestita dal cliente consente di crittografare le nuove versioni dei dati di configurazione nell'archivio di configurazione AWS AppConfig ospitato. Per ulteriori informazioni su questa chiave, consulta AWS AppConfig Supporta le chiavi di Customer Manager in [Sicurezza in AWS AppConfig](#).
  - b. Nella sezione Tag, scegli Aggiungi nuovo tag, quindi specifica una chiave e un valore opzionale.
8. Scegli Next (Successivo).
9. Nella sezione Definizione del flag di funzionalità, per Nome bandiera, inserisci un nome.
10. Per la chiave Flag, inserisci un identificatore di bandiera per distinguere i flag all'interno dello stesso profilo di configurazione. I flag all'interno dello stesso profilo di configurazione non possono avere la stessa chiave. Dopo aver creato il flag, è possibile modificare il nome del flag, ma non la chiave del flag.
11. (Facoltativo) Espandi la descrizione e inserisci le informazioni su questo contrassegno.
12. Seleziona Questo è un contrassegno a breve termine e, facoltativamente, scegli una data in cui disattivare o eliminare il contrassegno. AWS AppConfig non disabilita il flag alla data di obsolescenza.

13. (Facoltativo) Nella sezione Attributi del flag Feature, scegliete Definisci attributo. Gli attributi consentono di fornire valori aggiuntivi all'interno della bandiera. Per ulteriori informazioni su attributi e vincoli, vedere [Comprensione degli attributi dei feature flag](#)
  - a. Per Key, specificate una chiave di bandiera e sceglietene il tipo dall'elenco Tipo. Per informazioni sulle opzioni supportate per i campi Valore e Vincoli, consultate la sezione precedentemente citata sugli attributi.
  - b. Seleziona Valore richiesto per specificare se il valore di un attributo è obbligatorio.
  - c. Scegliete Definisci attributo per aggiungere altri attributi.
14. Nella sezione Feature flag value, scegliete Enabled per abilitare il flag. Usa lo stesso interruttore per disabilitare un flag quando raggiunge una data di deprecazione specificata, se applicabile.
15. Scegli Next (Successivo).
16. Nella pagina Rivedi e salva, verifica i dettagli del flag, quindi Salva e continua la distribuzione.

Passa a [Distribuzione di flag di funzionalità e dati di configurazione in AWS AppConfig](#).

## Creazione di un profilo di configurazione del feature flag (riga di comando)

La procedura seguente descrive come utilizzare AWS Command Line Interface (su Linux o Windows) o Tools for Windows per PowerShell creare un profilo di configurazione dei AWS AppConfig feature flag. Al momento della creazione del profilo di configurazione, è anche possibile creare un flag di funzionalità di base.

Per creare una configurazione di feature flag

1. Aprire il AWS CLI.
2. Crea un profilo di configurazione del feature flag specificandone il tipo come `AWS.AppConfig.FeatureFlags`. Il profilo di configurazione deve utilizzare hosted l'URI della posizione.

### Linux

```
aws appconfig create-configuration-profile \
--application-id APPLICATION_ID \
--name CONFIGURATION_PROFILE_NAME \
--location-uri hosted \
--type AWS.AppConfig.FeatureFlags
```

## Windows

```
aws appconfig create-configuration-profile ^
--application-id APPLICATION_ID ^
--name CONFIGURATION_PROFILE_NAME ^
--location-uri hosted ^
--type AWS.AppConfig.FeatureFlags
```

## PowerShell

```
New-APPConfigurationProfile ` 
-Name CONFIGURATION_PROFILE_NAME ` 
-ApplicationId APPLICATION_ID ` 
-LocationUri hosted ` 
-Type AWS.AppConfig.FeatureFlags
```

3. Crea i dati di configurazione del tuo feature flag. I tuoi dati devono essere in formato JSON e conformi allo schema AWS.AppConfig.FeatureFlags JSON. Per ulteriori informazioni sullo schema, vedere. [Comprensione del tipo di riferimento per AWS.AppConfig.FeatureFlags](#)
4. Utilizza l'CreateHostedConfigurationVersionAPI per salvare i dati di configurazione del feature flag in AWS AppConfig.

## Linux

```
aws appconfig create-hosted-configuration-version \
--application-id APPLICATION_ID \
--configuration-profile-id CONFIGURATION_PROFILE_ID \
--content-type "application/json" \
--content file://path/to/feature_flag_configuration_data.json \
--cli-binary-format raw-in-base64-out
```

## Windows

```
aws appconfig create-hosted-configuration-version ^
--application-id APPLICATION_ID ^
--configuration-profile-id CONFIGURATION_PROFILE_ID ^
--content-type "application/json" ^
--content file://path/to/feature_flag_configuration_data.json ^
```

```
--cli-binary-format raw-in-base64-out
```

## PowerShell

```
New-APPCHostedConfigurationVersion ` 
-ApplicationId APPLICATION_ID ` 
-ConfigurationProfileId CONFIGURATION_PROFILE_ID ` 
-ContentType "application/json" ` 
-Content file://path/to/feature_flag_configuration_data.json
```

Il comando carica il contenuto specificato per il Content parametro dal disco. Il contenuto deve essere simile all'esempio seguente.

```
{
  "flags": {
    "ui_refresh": {
      "name": "UI Refresh"
    }
  },
  "values": {
    "ui_refresh": {
      "enabled": false,
      "attributeValues": {
        "dark_mode_support": true
      }
    }
  },
  "version": "1"
}
```

Il sistema restituisce informazioni simili alle seguenti.

## Linux

```
{
  "ApplicationId" : "ui_refresh",
  "ConfigurationProfileId" : "UI Refresh",
  "VersionNumber" : "1",
  "ContentType" : "application/json"
}
```

## Windows

```
{  
  "ApplicationId" : "ui_refresh",  
  "ConfigurationProfileId" : "UI Refresh",  
  "VersionNumber" : "1",  
  "ContentType" : "application/json"  
}
```

## PowerShell

```
ApplicationId      : ui_refresh  
ConfigurationProfileId : UI Refresh  
VersionNumber      : 1  
ContentType        : application/json
```

**service\_returned\_content\_file** Contiene i dati di configurazione che includono alcuni metadati AWS AppConfig generati.

### Note

Quando crei la versione di configurazione ospitata, AWS AppConfig verifica che i dati siano conformi allo `AWS.AppConfig.FeatureFlags` schema JSON. AWS AppConfig verifica inoltre che ogni attributo feature flag presente nei dati soddisfi i vincoli definiti per tali attributi.

## Creazione di flag di funzionalità multivarianti

Le varianti Feature Flag consentono di definire una serie di possibili valori di flag da restituire per una richiesta. È inoltre possibile configurare diversi stati (abilitati o disabilitati) per i flag multivarianti. Quando si richiede un flag configurato con varianti, l'applicazione fornisce un contesto che AWS AppConfig valuta in base a una serie di regole definite dall'utente. A seconda del contesto specificato nella richiesta e delle regole definite per la variante, AWS AppConfig restituisce valori di flag diversi all'applicazione.

La schermata seguente mostra un esempio di feature flag con tre varianti definite dall'utente e la variante predefinita.

Feature flag variants <small>Info</small>		<a href="#">Reorder variant up</a>	<a href="#">Reorder variant down</a>	<a href="#">Edit</a>	<a href="#">Create variant</a>
Name	Enabled value	Attribute values	Rule		
<input type="radio"/> beta testers	<input checked="" type="checkbox"/> ON	-	(or (eq \$userId "Alice") (eq \$userId "123456789012"))		
<input type="radio"/> EU demographic	<input checked="" type="checkbox"/> ON	-	(and (ends_with \$email "@example.com") (eq \$continent "EU"))		
<input type="radio"/> QA testing	<input checked="" type="checkbox"/> ON	-	(and (matches pattern: ".@example.com" in: \$email) (contains \$roles "Engineer") (gt \$tenure 5))		
<input type="radio"/> default	<input checked="" type="checkbox"/> ON	-	-		

i **Variant order is used for evaluation logic**  
 Variants are evaluated as an ordered list based on the order shown and any specified rules. The variant at the top of the list is evaluated first. If no rules match the supplied context, AWS AppConfig returns the default variant.

## Argomenti

- [Comprendere dei concetti dei feature flag multivarianti e dei casi d'uso comuni](#)
- [Comprendere delle regole del feature flag multivariante](#)
- [Creazione di un flag di funzionalità multivariante](#)

## Comprendere dei concetti dei feature flag multivarianti e dei casi d'uso comuni

Per aiutarti a comprendere meglio le varianti dei flag di funzionalità, questa sezione spiega i concetti relativi alle varianti dei flag e i casi d'uso più comuni.

### Concetti

- Feature flag: un tipo di AWS AppConfig configurazione utilizzato per controllare il comportamento di una funzionalità in un'applicazione. Un flag ha uno stato (abilitato o disabilitato) e un set opzionale di attributi contenenti valori arbitrari di stringa, numerici, booleani o matrici.
- Variante del flag di funzionalità: una combinazione specifica di valori di status e di attributo appartenenti a un flag di funzionalità. Un feature flag può avere più varianti.
- Regola di variante: espressione definita dall'utente utilizzata per selezionare una variante del feature flag. Ogni variante ha una propria regola che AWS AppConfig valuta se restituirla o meno.
- Variante predefinita: una variante speciale che viene restituita quando non viene selezionata nessun'altra variante. Tutti i flag di funzionalità multivarianti hanno una variante predefinita.

Nota, la variante predefinita deve essere l'ultima nell'ordine delle varianti e non può avere regole associate. Se non è definita per ultima, AWS AppConfig restituisce a `BadRequestException` quando si tenta di creare il flag multivariante.

- Contesto: chiavi e valori definiti dall'utente passati al AWS AppConfig momento del recupero della configurazione. I valori di contesto vengono utilizzati durante la valutazione delle regole per selezionare la variante del feature flag da restituire.

### Note

AWS AppConfig l'agente valuta le regole delle varianti e determina quale regola si applica alla richiesta in base al contesto fornito. Per ulteriori informazioni sul recupero dei flag di funzionalità multivarianti, vedere. [Recupero dei flag delle funzionalità di base e multivarianti](#)

## Casi d'uso comuni

Questa sezione descrive due casi d'uso comuni per le varianti dei feature flag.

### Segmentazione degli utenti

La segmentazione degli utenti è il processo di divisione degli utenti in base a determinati attributi. Ad esempio, puoi utilizzare le varianti flag per esporre una funzionalità ad alcuni utenti ma non ad altri in base all'ID utente, alla posizione geografica, al tipo di dispositivo o alla frequenza di acquisto.

Utilizzando l'esempio della frequenza di acquisto, supponiamo che l'applicazione di commercio supporti una funzionalità per aumentare la fidelizzazione dei clienti. Puoi utilizzare le varianti di bandiera per configurare diversi tipi di incentivi da mostrare a un utente in base all'ultima volta che ha acquistato qualcosa. A un nuovo utente potrebbe essere offerto un piccolo sconto per incoraggiarlo a diventare cliente, mentre a un cliente abituale potrebbe essere concesso uno sconto maggiore se acquista qualcosa da una nuova categoria.

### Suddivisione del traffico

La suddivisione del traffico è il processo di selezione di una variante di bandiera casuale ma coerente in base a un valore di contesto definito dall'utente. Ad esempio, potresti voler eseguire un esperimento in cui una piccola percentuale dei tuoi utenti (identificati dal loro ID utente) vede una particolare variante. In alternativa, potresti voler eseguire un'implementazione graduale delle funzionalità in cui una funzionalità venga esposta prima al 5% degli utenti, poi al 15%, quindi al 40%, quindi al 100%, mantenendo un'esperienza utente coerente durante l'implementazione.

Utilizzando l'esempio di sperimentazione, potreste utilizzare le varianti dei flag per testare un nuovo stile di pulsante per l'azione principale sulla home page dell'applicazione e vedere se genera più clic.

Per il vostro esperimento, potreste creare una variante di bandiera con una regola di suddivisione del traffico che indichi il 5% degli utenti a visualizzare il nuovo stile, mentre la variante predefinita indica gli utenti che dovrebbero continuare a vedere lo stile esistente. Se l'esperimento ha esito positivo, puoi aumentare il valore percentuale o persino impostare la variante come predefinita.

### Comprensione delle regole del feature flag multivariante

Quando crei una variante del feature flag, specifichi una regola per essa. Le regole sono espressioni che accettano valori di contesto come input e producono un risultato booleano come output. Ad esempio, è possibile definire una regola per selezionare una variante di bandiera per gli utenti beta, identificata dall'ID dell'account, testando un aggiornamento dell'interfaccia utente. In questo scenario, procedi come segue:

1. Create un nuovo profilo di configurazione del feature flag chiamato UI Refresh.
2. Crea un nuovo flag di funzionalità chiamato ui\_refresh.
3. Modifica il flag della funzionalità dopo averlo creato per aggiungere varianti.
4. Crea e abilita una nuova variante chiamata BetaUsers.
5. Definisci una regola per selezionare BetaUsers la variante se l'ID dell'account dal contesto della richiesta si trova in un elenco di account IDs approvati per visualizzare la nuova esperienza beta.
6. Verifica che lo stato della variante predefinita sia impostato su Disabilitato.

#### Note

Le varianti vengono valutate come un elenco ordinato in base all'ordine in cui sono definite nella console. La variante in cima all'elenco viene valutata per prima. Se nessuna regola corrisponde al contesto fornito, AWS AppConfig restituisce la variante predefinita.

Quando AWS AppConfig elabora la richiesta del feature flag, confronta prima il contesto fornito, che include l'AccountID (in questo esempio) con BetaUsers la variante. Se il contesto corrisponde alla regola per BetaUsers, AWS AppConfig restituisce i dati di configurazione per l'esperienza beta. Se il contesto non include un ID account o se l'ID dell'account termina con un valore diverso da 123, AWS AppConfig restituisce i dati di configurazione per la regola predefinita, il che significa che l'utente visualizza l'esperienza corrente in produzione.

### Note

Per informazioni sul recupero dei flag di funzionalità multivarianti, consulta. [Recupero dei flag delle funzionalità di base e multivarianti](#)

## Definizione delle regole per i flag di funzionalità multivarianti

Una regola variante è un'espressione composta da uno o più operandi e un operatore. Un operando è un valore specifico utilizzato durante la valutazione di una regola. I valori degli operandi possono essere statici, ad esempio un numero letterale o una stringa, o variabili, ad esempio il valore trovato in un contesto o il risultato di un'altra espressione. Un operatore, ad esempio «maggiore di», è un test o un'azione applicata ai relativi operandi che produce un valore. Un'espressione di regola variante deve produrre un «vero» o un «falso» per essere valida.

### Operandi

Tipo	Description	Esempio
Stringa	Una sequenza di caratteri UTF-8, racchiusa tra virgolette doppie.	"apple", "##ë# ##š##"
Número intero	Un valore intero a 64 bit.	-7, 42
Float	Un valore a virgola mobile IEEE-754 a 64 bit.	3.14, 1.234e-5
Time stamp	Un momento temporale specifico, come descritto dalla nota del <a href="#">W3C</a> sui formati di data e ora.	2012-03-04T05:06:0 7-08:00, 2024-01
Booleano	Un valore vero o falso.	true, false
Valore di contesto	Un valore parametrizzato sotto forma di <code>\$ key</code> che viene recuperato dal contesto	\$country, \$userId

Tipo	Description	Esempio
	durante la valutazione della regola.	

## Operatori di confronto

Operatore	Description	Esempio
eq	Determina se un valore di contesto è uguale a un determinato valore.	(eq \$state "Virginia")
gt	Determina se un valore di contesto è maggiore di un determinato valore.	(gt \$age 65)
gte	Determina se un valore di contesto è maggiore o uguale a un determinato valore.	(gte \$age 65)
lt	Determina se un valore di contesto è inferiore a un determinato valore.	(lt \$age 65)
lte	Determina se un valore di contesto è minore o uguale a un determinato valore.	(lte \$age 65)

## Operatori logici

Operatore	Description	Esempio
and	Determina se entrambi gli operandi sono veri.	<pre>(and   (eq \$state "Virginia")   (gt \$age 65))</pre>

Operatore	Description	Esempio
		)
or	Determina se almeno uno degli operandi è vero.	(or (eq \$state "Virginia") (gt \$age 65) )
not	Inverte il valore di un'espresione.	(not (eq \$state "Virginia"))

## Operatori personalizzati

Operatore	Description	Esempio
inizia_con	Determina se un valore di contesto inizia con un determinato prefisso.	(begins_with \$state "A")
termina_con	Determina se un valore di contesto termina con un determinato prefisso.	(ends_with \$email "amazon.com")
contiene	Determina se un valore di contesto contiene una determinata sottocorona.	(contains \$promoCode "WIN")
in	Determina se un valore di contesto è contenuto in un elenco di costanti.	(in \$userId ["123", "456"])
fiammiferi	Determina se un valore di contesto corrisponde a un determinato modello di espressione regolare.	(matches in::\$greeting pattern::"h.*y")

Operatore	Description	Esempio
exists	Determina se è stato fornito un valore per una chiave di contesto.	(exists key::"country")
dividi	<p>Restituisce una determinata percentuale di traffico in base a un hash coerente dei valori di contesto forniti.</p> <p>true Per una spiegazione dettagliata di come split funziona, consultate la sezione successiva di questo argomento, <a href="#">Comprendere dell'operatore split</a></p> <p>Si noti che seed si tratta di una proprietà facoltativa. Se non lo specifichi seed, l'hash è coerente a livello locale, il che significa che il traffico verrà suddiviso in modo coerente per quel flag, ma altri flag che ricevono lo stesso valore di contesto potrebbero suddividere il traffico in modo diverso.</p> <p>Se fornito, seed è garantito che ogni valore univoco suddividerà il traffico in modo uniforme tra feature flag, profili di configurazione e Account AWS</p>	(split pct::10 by::\$userId seed::"abc")

## Comprendere l'operatore split

La sezione seguente descrive come si comporta l'operatore split quando viene utilizzato in diversi scenari. Come promemoria, split calcola una determinata percentuale di traffico sulla base di un hash coerente del valore di contesto fornito. true Per capirlo meglio, considera il seguente scenario di base che utilizza la divisione con due varianti:

```
A: (split by::$uniqueId pct::20)
C: <no rule>
```

Come previsto, fornendo un insieme casuale di uniqueId valori si ottiene una distribuzione che è approssimativamente:

```
A: 20%
C: 80%
```

Se aggiungi una terza variante, ma usi la stessa percentuale di frazionamento in questo modo:

```
A: (split by::$uniqueId pct::20)
B: (split by::$uniqueId pct::20)
C: <default>
```

Si ottiene la seguente distribuzione:

```
A: 20%
B: 0%
C: 80%
```

Questa distribuzione potenzialmente inaspettata si verifica perché ogni regola di variante viene valutata in ordine e la prima corrispondenza determina la variante restituita. Quando viene valutata la regola A, il 20% dei uniqueId valori corrisponde, quindi viene restituita la prima variante. Successivamente, viene valutata la regola B. Tuttavia, tutti i uniqueId valori che avrebbero dovuto corrispondere alla seconda istruzione split corrispondevano già alla regola della variante A, quindi nessun valore corrisponde a B. Viene invece restituita la variante predefinita.

Consideriamo ora un terzo esempio.

```
A: (split by::$uniqueId pct::20)
B: (split by::$uniqueId pct::25)
C: <default>
```

Come nell'esempio precedente, il primo 20% dei `uniqueId` valori corrisponde alla regola A. Per la regola della variante B, il 25% di tutti i `uniqueId` valori corrisponderebbe, ma la maggior parte di quelli precedentemente corrispondenti alla regola A. Ciò lascia il 5% del totale per la variante B, mentre il resto riceve la variante C. La distribuzione sarebbe simile alla seguente:

```
A: 20%
B: 5%
C: 75%
```

## Utilizzo della proprietà `seed`

È possibile utilizzare la `seed` proprietà per garantire che il traffico venga suddiviso in modo coerente per un determinato valore di contesto indipendentemente da dove viene utilizzato l'operatore di divisione. Se non lo specifichiseed, l'hash è coerente a livello locale, il che significa che il traffico verrà suddiviso in modo coerente per quel flag, ma altri flag che ricevono lo stesso valore di contesto potrebbero suddividere il traffico in modo diverso. Se fornito, `seed` è garantito che ogni valore univoco suddividerà il traffico in modo uniforme tra feature flag, profili di configurazione e Account AWS

In genere, i clienti utilizzano lo stesso `seed` valore tra le varianti all'interno di un flag quando suddividono il traffico sulla stessa proprietà di contesto. Tuttavia, a volte può avere senso utilizzare un valore iniziale diverso. Ecco un esempio che utilizza semi diversi per le regole A e B:

```
A: (split by::$uniqueId pct::20 seed::"seed_one")
B: (split by::$uniqueId pct::25 seed::"seed_two")
C: <default>
```

Come in precedenza, il 20% dei `uniqueId` valori corrispondenti corrisponde alla regola A. Ciò significa che l'80% dei valori soddisfa e viene testato in base alla regola variante B. Poiché il seme è diverso, non c'è correlazione tra i valori che corrispondono ad A e i valori che corrispondono alla regola B. Tuttavia, vi sono solo l'80% in più di `uniqueId` valori da dividere, il 25% di quel numero corrisponde alla regola B e il 75% no. Ciò si traduce nella seguente distribuzione:

```
A: 20%
B: 20% (25% of what falls through from A, or 25% of 80%)
```

C: 60%

## Creazione di un flag di funzionalità multivariante

Utilizzate le procedure descritte in questa sezione per creare varianti di un feature flag.

### Prima di iniziare

Prendi nota delle seguenti informazioni importanti.

- È possibile creare varianti di feature flag esistenti modificandole. Non è possibile creare varianti di un nuovo feature flag quando si crea un nuovo profilo di configurazione. È necessario prima completare il flusso di lavoro di creazione del nuovo profilo di configurazione. Dopo aver creato il profilo di configurazione, è possibile aggiungere varianti a qualsiasi flag all'interno del profilo di configurazione. Per informazioni su come creare un nuovo profilo di configurazione, vedere [Creazione di un profilo di configurazione del feature flag in AWS AppConfig](#).
- Per recuperare i dati delle varianti di feature flag per le piattaforme di calcolo Amazon EC2, Amazon ECS e Amazon EKS, devi utilizzare la versione AWS AppConfig dell'agente 2.0.4416 o successiva.
- Per motivi di prestazioni AWS CLI e le chiamate SDK per non recuperare i dati delle varianti. AWS AppConfig Per ulteriori informazioni su AWS AppConfig Agent, consulta. [Come utilizzare AWS AppConfig Agent per recuperare i dati di configurazione](#)
- Quando create una variante del feature flag, specificate una regola per essa. Le regole sono espressioni che prendono il contesto della richiesta come input e producono un risultato booleano come output. Prima di creare varianti, esamina gli operandi e gli operatori supportati per le regole delle varianti di bandiera. È possibile creare regole prima di creare varianti. Per ulteriori informazioni, consulta [Comprensione delle regole del feature flag multivariante](#).

### Argomenti

- [Creazione di un flag di funzionalità multivariante \(console\)](#)
- [Creazione di un flag di funzionalità multivariante \(riga di comando\)](#)

### Creazione di un flag di funzionalità multivariante (console)

La procedura seguente descrive come creare un flag di funzionalità multivariante per un profilo di configurazione esistente utilizzando la AWS AppConfig console. È inoltre possibile modificare i flag di funzionalità esistenti per creare varianti.

## Per creare un flag di funzionalità multivariante

1. Apri la AWS Systems Manager console all'indirizzo <https://console.aws.amazon.com/systems-manager/appconfig/>.
2. Nel riquadro di navigazione, scegli Applicazioni, quindi scegli un'applicazione.
3. Nella scheda Profili di configurazione e flag di funzionalità, scegliete un profilo di configurazione dei feature flag esistente.
4. Nella sezione Bandiere, scegli Aggiungi nuova bandiera.
5. Nella sezione Definizione della bandiera della caratteristica, per Nome bandiera, inserisci un nome.
6. Per la chiave Flag, inserisci un identificatore di bandiera per distinguere i flag all'interno dello stesso profilo di configurazione. I flag all'interno dello stesso profilo di configurazione non possono avere la stessa chiave. Dopo aver creato il flag, è possibile modificare il nome del flag, ma non la chiave del flag.
7. (Facoltativo) Nel campo Descrizione, inserisci le informazioni su questo contrassegno.
8. Nella sezione Varianti, scegli Contrassegno multivariante.
9. (Facoltativo) Nella sezione Attributi del flag Feature, scegliete Definisci attributo. Gli attributi consentono di fornire valori aggiuntivi all'interno della bandiera. Per ulteriori informazioni su attributi e vincoli, vedere. [Comprensione degli attributi dei feature flag](#)
  - a. Per Key, specificate una chiave di bandiera e sceglietene il tipo dall'elenco Tipo. Per informazioni sulle opzioni supportate per i campi Valore e Vincoli, consultate la sezione precedentemente citata sugli attributi.
  - b. Seleziona Valore richiesto per specificare se il valore di un attributo è obbligatorio.
  - c. Scegliete Definisci attributo per aggiungere altri attributi.
  - d. Scegliete Applica per salvare le modifiche agli attributi.
10. Nella sezione Varianti del Feature flag, scegliete Crea variante.
  - a. Per Nome variante, inserisci un nome.
  - b. Utilizzate l'interruttore Valore abilitato per abilitare la variante.
  - c. Nella casella di testo Regola, inserisci una regola.
  - d. Utilizzate le opzioni Crea variante > Crea variante sopra o Crea variante sotto per creare varianti aggiuntive per questo contrassegno.

- e. Nella sezione Variante predefinita, utilizzate l'interruttore Valore abilitato per abilitare la variante predefinita. Facoltativamente, fornite i valori per gli attributi definiti nel passaggio 10.
  - f. Scegli Applica.
11. Verifica i dettagli del contrassegno e delle sue varianti e scegli Crea bandiera.

Per informazioni sulla distribuzione del nuovo flag di funzionalità con varianti, consulta [Distribuzione di flag di funzionalità e dati di configurazione in AWS AppConfig](#).

Creazione di un flag di funzionalità multivariante (riga di comando)

La procedura seguente descrive come utilizzare AWS Command Line Interface (su Linux o Windows) o Tools for Windows per PowerShell creare un flag di funzionalità multivariante per un profilo di configurazione esistente. È inoltre possibile modificare i flag di funzionalità esistenti per creare varianti.

Prima di iniziare

Completate le seguenti attività prima di creare un feature flag multivariante utilizzando il AWS CLI

- Create un profilo di configurazione del feature flag. Per ulteriori informazioni, consulta [Creazione di un profilo di configurazione del feature flag in AWS AppConfig](#).
- Effettua l'aggiornamento alla versione più recente di AWS CLI. Per ulteriori informazioni, consulta [Installare o aggiornare alla versione più recente di AWS CLI nella Guida per l'AWS Command Line Interface utente](#).

Per creare un contrassegno di funzionalità multivariante

1. Create un file di configurazione sul computer locale che specifichi i dettagli del flag multivariante che desiderate creare. Salvate il file con un'.json estensione. Il file deve aderire allo schema [AWS AppConfig FeatureFlags JSON](#). Il contenuto dello schema del file di configurazione sarà simile al seguente.

```
{  
  "flags": {  
    "FLAG_NAME": {  
      "attributes": {  
        "ATTRIBUTE_NAME": {  
          "constraints": {  
            "type": "CONSTRAINT_TYPE"  
          }  
        }  
      }  
    }  
  }  
}
```

```
        }
    },
},
"description": "FLAG_DESCRIPTION",
"name": "VARIANT_NAME"
}
},
"values": {
"VARIANT_VALUE_NAME": {
"_variants": [
{
"attributeValues": {
"ATTRIBUTE_NAME": BOOLEAN
},
"enabled": BOOLEAN,
"name": "VARIANT_NAME",
"rule": "VARIANT_RULE"
},
{
"attributeValues": {
"ATTRIBUTE_NAME": BOOLEAN
},
"enabled": BOOLEAN,
"name": "VARIANT_NAME",
"rule": "VARIANT_RULE"
},
{
"attributeValues": {
"ATTRIBUTE_NAME": BOOLEAN
},
"enabled": BOOLEAN,
"name": "VARIANT_NAME",
"rule": "VARIANT_RULE"
},
{
"attributeValues": {
"ATTRIBUTE_NAME": BOOLEAN
},
"enabled": BOOLEAN,
"name": "VARIANT_NAME",
"rule": "VARIANT_RULE"
}
]
}
}
```

```
},
"version": "VERSION_NUMBER"
}
```

Ecco un esempio con tre varianti e la variante predefinita.

```
{
  "flags": {
    "ui_refresh": {
      "attributes": {
        "dark_mode_support": {
          "constraints": {
            "type": "boolean"
          }
        }
      },
      "description": "A release flag used to release a new UI",
      "name": "UI Refresh"
    }
  },
  "values": {
    "ui_refresh": {
      "_variants": [
        {
          "attributeValues": {
            "dark_mode_support": true
          },
          "enabled": true,
          "name": "QA",
          "rule": "(ends_with $email \"qa-testers.mycompany.com\")"
        },
        {
          "attributeValues": {
            "dark_mode_support": true
          },
          "enabled": true,
          "name": "Beta Testers",
          "rule": "(exists key::\"opted_in_to_beta\")"
        },
        {
          "attributeValues": {
            "dark_mode_support": false
          },

```

```
        "enabled": true,
        "name": "Sample Population",
        "rule": "(split pct::10 by::$email)"
    },
    {
        "attributeValues": {
            "dark_mode_support": false
        },
        "enabled": false,
        "name": "Default Variant"
    }
]
},
"version": "1"
}
```

2. Utilizza l'CreateHostedConfigurationVersionAPI per salvare i dati di configurazione del feature flag in AWS AppConfig.

### Linux

```
aws appconfig create-hosted-configuration-version \
--application-id APPLICATION_ID \
--configuration-profile-id CONFIGURATION_PROFILE_ID \
--content-type "application/json" \
--content file://path/to/feature_flag_configuration_data.json \
--cli-binary-format raw-in-base64-out \
outfile
```

### Windows

```
aws appconfig create-hosted-configuration-version ^
--application-id APPLICATION_ID ^
--configuration-profile-id CONFIGURATION_PROFILE_ID ^
--content-type "application/json" ^
--content file://path/to/feature_flag_configuration_data.json ^
--cli-binary-format raw-in-base64-out ^
outfile
```

## PowerShell

```
New-APPCHostedConfigurationVersion  
-ApplicationId APPLICATION_ID  
-ConfigurationProfileId CONFIGURATION_PROFILE_ID  
-ContentType "application/json"  
-Content file://path/to/feature_flag_configuration_data.json  
-Raw
```

**service\_returned\_content\_file** Contiene i dati di configurazione che includono alcuni metadati AWS AppConfig generati.

### Note

Quando crei la versione di configurazione ospitata, AWS AppConfig verifica che i dati siano conformi allo [AWS.AppConfig.FeatureFlags](#) schema JSON. AWS AppConfig verifica inoltre che ogni attributo feature flag presente nei dati soddisfi i vincoli definiti per tali attributi.

## Comprendere il tipo di riferimento per AWS.AppConfig.FeatureFlags

Utilizzate lo schema `AWS.AppConfig.FeatureFlags` JSON come riferimento per creare i dati di configurazione dei feature flag.

```
{  
  "$schema": "http://json-schema.org/draft-07/schema#",  
  "definitions": {  
    "flagSetDefinition": {  
      "type": "object",  
      "properties": {  
        "version": {  
          "$ref": "#/definitions/flagSchemaVersions"  
        },  
        "flags": {  
          "$ref": "#/definitions/flagDefinitions"  
        },  
        "values": {  
          "$ref": "#/definitions/flagValues"  
        }  
      }  
    }  
  }  
}
```

```
        }
    },
    "required": ["version"],
    "additionalProperties": false
},
"flagDefinitions": {
    "type": "object",
    "patternProperties": {
        "^[a-z][a-zA-Z\\d_-]{0,63}$": {
            "$ref": "#/definitions/flagDefinition"
        }
    },
    "additionalProperties": false
},
"flagDefinition": {
    "type": "object",
    "properties": {
        "name": {
            "$ref": "#/definitions/customerDefinedName"
        },
        "description": {
            "$ref": "#/definitions/customerDefinedDescription"
        },
        "_createdAt": {
            "type": "string"
        },
        "_updatedAt": {
            "type": "string"
        },
        "_deprecation": {
            "type": "object",
            "properties": {
                "status": {
                    "type": "string",
                    "enum": ["planned"]
                },
                "date": {
                    "type": "string",
                    "format": "date"
                }
            },
            "additionalProperties": false
        },
        "attributes": {

```

```
        "$ref": "#/definitions/attributeDefinitions"
    }
},
"additionalProperties": false
},
"attributeDefinitions": {
    "type": "object",
    "patternProperties": {
        "^[a-z][a-zA-Z\d_-]{0,63}$": {
            "$ref": "#/definitions/attributeDefinition"
        }
    },
    "maxProperties": 25,
    "additionalProperties": false
},
"attributeDefinition": {
    "type": "object",
    "properties": {
        "description": {
            "$ref": "#/definitions/customerDefinedDescription"
        },
        "constraints": {
            "oneOf": [
                { "$ref": "#/definitions/numberConstraints" },
                { "$ref": "#/definitions/stringConstraints" },
                { "$ref": "#/definitions/arrayConstraints" },
                { "$ref": "#/definitions/boolConstraints" }
            ]
        }
    },
    "additionalProperties": false
},
"flagValues": {
    "type": "object",
    "patternProperties": {
        "^[a-z][a-zA-Z\d_-]{0,63}$": {
            "$ref": "#/definitions/flagValue"
        }
    },
    "additionalProperties": false
},
"flagValue": {
    "type": "object",
    "properties": {
```

```
  "enabled": {
    "type": "boolean"
  },
  "_createdAt": {
    "type": "string"
  },
  "_updatedAt": {
    "type": "string"
  },
  "_variants": {
    "type": "array",
    "maxLength": 32,
    "items": {
      "$ref": "#/definitions/variant"
    }
  }
},
"patternProperties": {
  "^[a-z][a-zA-Z\d_-]{0,63}$": {
    "$ref": "#/definitions/attributeValue",
    "maxProperties": 25
  }
},
"additionalProperties": false
},
"attributeValue": {
  "oneOf": [
    { "type": "string", "maxLength": 1024 },
    { "type": "number" },
    { "type": "boolean" },
    {
      "type": "array",
      "oneOf": [
        {
          "items": {
            "type": "string",
            "maxLength": 1024
          }
        },
        {
          "items": {
            "type": "number"
          }
        }
      ]
    }
  ]
}
```

```
        ]
    }
],
"additionalProperties": false
},
"stringConstraints": {
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": ["string"]
        },
        "required": {
            "type": "boolean"
        },
        "pattern": {
            "type": "string",
            "maxLength": 1024
        },
        "enum": {
            "type": "array",
            "maxLength": 100,
            "items": {
                "oneOf": [
                    {
                        "type": "string",
                        "maxLength": 1024
                    },
                    {
                        "type": "integer"
                    }
                ]
            }
        }
    }
},
"required": ["type"],
"not": {
    "required": ["pattern", "enum"]
},
"additionalProperties": false
},
"numberConstraints": {
    "type": "object",
    "properties": {
```

```
"type": {
    "type": "string",
    "enum": ["number"]
},
"required": {
    "type": "boolean"
},
"minimum": {
    "type": "integer"
},
"maximum": {
    "type": "integer"
}
},
"required": ["type"],
"additionalProperties": false
},
"arrayConstraints": {
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": ["array"]
        },
        "required": {
            "type": "boolean"
        },
        "elements": {
            "$ref": "#/definitions/elementConstraints"
        }
    },
    "required": ["type"],
    "additionalProperties": false
},
"boolConstraints": {
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": ["boolean"]
        },
        "required": {
            "type": "boolean"
        }
    }
}
```

```
  },
  "required": ["type"],
  "additionalProperties": false
},
"elementConstraints": {
  "oneOf": [
    { "$ref": "#/definitions/numberConstraints" },
    { "$ref": "#/definitions/stringConstraints" }
  ]
},
"variant": {
  "type": "object",
  "properties": {
    "enabled": {
      "type": "boolean"
    },
    "name": {
      "$ref": "#/definitions/customerDefinedName"
    },
    "rule": {
      "type": "string",
      "maxLength": 16384
    },
    "attributeValues": {
      "type": "object",
      "patternProperties": {
        "^[a-z][a-zA-Z\\d_-]{0,63}$": {
          "$ref": "#/definitions/attributeValue"
        }
      },
      "maxProperties": 25,
      "additionalProperties": false
    }
  },
  "required": ["name", "enabled"],
  "additionalProperties": false
},
"customerDefinedName": {
  "type": "string",
  "pattern": "^[^\\n]{1,64}$"
},
"customerDefinedDescription": {
  "type": "string",
  "maxLength": 1024
}
```

```
},
  "flagSchemaVersions": {
    "type": "string",
    "enum": ["1"]
  },
  "type": "object",
  "$ref": "#/definitions/flagSetDefinition",
  "additionalProperties": false
}
```

### Important

Per recuperare i dati di configurazione dei feature flag, l'applicazione deve chiamare l'GetLatestConfigurationAPI. Non è possibile recuperare i dati di configurazione dei feature flag chiamandoGetConfiguration, il che è obsoleto. Per ulteriori informazioni, consulta [GetLatestConfiguration](#) nella documentazione di riferimento dell'API AWS AppConfig

Quando l'applicazione chiama [GetLatestConfiguration](#) e riceve una configurazione appena distribuita, le informazioni che definiscono i flag e gli attributi delle funzionalità vengono rimosse. Il JSON semplificato contiene una mappa di chiavi che corrispondono a ciascuna delle chiavi flag specificate. Il JSON semplificato contiene anche i valori mappati di `true` o per l'attributo. `false` enabled Se un flag è enabled impostato a `true`, saranno presenti anche tutti gli attributi del flag. Lo schema JSON seguente descrive il formato dell'output JSON.

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "patternProperties": {
    "^[a-z][a-zA-Z\\d_-]{0,63}$": {
      "$ref": "#/definitions/attributeValuesMap"
    }
  },
  "additionalProperties": false,
  "definitions": {
    "attributeValuesMap": {
      "type": "object",
      "properties": {
        "enabled": {

```

```
        "type": "boolean"
    }
},
"required": ["enabled"],
"patternProperties": {
    "^[a-z][a-zA-Z\d_-]{0,63}$": {
        "$ref": "#/definitions/attributeValue"
    }
},
"maxProperties": 25,
"additionalProperties": false
},
"attributeValue": {
    "oneOf": [
        { "type": "string", "maxLength": 1024 },
        { "type": "number" },
        { "type": "boolean" },
        {
            "type": "array",
            "oneOf": [
                {
                    "items": {
                        "oneOf": [
                            {
                                "type": "string",
                                "maxLength": 1024
                            }
                        ]
                    }
                },
                {
                    "items": {
                        "oneOf": [
                            {
                                "type": "number"
                            }
                        ]
                    }
                }
            ]
        }
    ],
    "additionalProperties": false
}
```

```
 }  
 }
```

## Salvataggio di una versione precedente di Feature Flag in una nuova versione

Quando aggiorni un feature flag, salva AWS AppConfig automaticamente le modifiche in una nuova versione. Se desiderate utilizzare una versione precedente del feature flag, dovete copiarla in una versione bozza e salvarla. Non potete modificare e salvare le modifiche a una versione precedente del flag senza salvarla in una nuova versione.

Per modificare una versione precedente del Feature Flag e salvarla in una nuova versione

1. Apri la AWS Systems Manager console all'indirizzo <https://console.aws.amazon.com/systems-manager/appconfig/>.
2. Nel pannello di navigazione, scegli Applicazioni, quindi scegli l'applicazione con il contrassegno di funzionalità che desideri modificare e salvare in una nuova versione.
3. Nella scheda Profili di configurazione e bandiere di funzionalità, scegliete il profilo di configurazione con il flag di funzionalità che desiderate modificare e salvatelo in una nuova versione.
4. Nella scheda Feature flags, utilizzate l'elenco Versione per scegliere la versione da modificare e salvare in una nuova versione.
5. Scegliete Copia nella versione bozza.
6. Nel campo Etichetta della versione, inserisci una nuova etichetta (facoltativa, ma consigliata).
7. Nel campo Descrizione della versione, inserisci una nuova descrizione (facoltativa, ma consigliata).
8. Scegli Salva versione.
9. Scegli Avvia distribuzione per distribuire la nuova versione.

## Creazione di un profilo di configurazione in formato libero in AWS AppConfig

I dati di configurazione sono una raccolta di impostazioni che influenzano il comportamento dell'applicazione. Un profilo di configurazione include, tra le altre cose, un URI che consente di AWS AppConfig localizzare i dati di configurazione nella posizione archiviata e un tipo di configurazione. Con i profili di configurazione in formato libero, è possibile archiviare i dati

nell'archivio di configurazione AWS AppConfig ospitato o in uno dei seguenti strumenti Servizi AWS e negli strumenti di Systems Manager:

Location (Ubicazione)	Tipi di file supportati
AWS AppConfig archivio di configurazione ospitato	YAML, JSON e testo se aggiunti utilizzando la <a href="#">Console di gestione AWS</a> Qualsiasi tipo di file se aggiunto utilizzando l' <a href="#">AWS AppConfig CreateHostedConfigurationVersionazione API</a> .
<a href="#">Amazon Simple Storage Service (Amazon S3)</a>	Qualsiasi
<a href="#">AWS CodePipeline</a>	Pipeline (come definita dal servizio)
<a href="#">Gestione dei segreti AWS</a>	Segreto (come definito dal servizio)
<a href="#">AWS Systems Manager Archivio parametri</a>	Parametri di stringa standard e sicuri (come definiti da Parameter Store)
<a href="#">AWS Systems Manager archivio documenti (documenti SSM)</a>	YAML, JSON, testo

Un profilo di configurazione può anche includere validatori opzionali per garantire che i dati di configurazione siano corretti dal punto di vista sintattico e semantico. AWS AppConfig esegue un controllo utilizzando i validatori quando si avvia una distribuzione. Se vengono rilevati errori, la distribuzione si interrompe prima di apportare modifiche ai target di configurazione.

#### Note

Se possibile, consigliamo di ospitare i dati di configurazione nell'archivio di configurazione AWS AppConfig ospitato in quanto offre la maggior parte delle funzionalità e dei miglioramenti.

Per le configurazioni in formato libero archiviate nell'archivio di configurazione AWS AppConfig ospitato o nei documenti SSM, è possibile creare la configurazione in formato libero utilizzando la console Systems Manager al momento della creazione di un profilo di configurazione. Il processo è descritto più avanti in questo argomento.

Per le configurazioni in formato libero archiviate in Parameter Store, Secrets Manager o Amazon S3, devi prima creare il parametro, il segreto o l'oggetto e archiviarlo nell'archivio di configurazione pertinente. Dopo aver archiviato i dati di configurazione, utilizza la procedura descritta in questo argomento per creare il profilo di configurazione.

## Argomenti

- [Comprendere i validatori](#)
- [Comprensione delle quote e delle limitazioni dell'archivio di configurazione](#)
- [Informazioni sull'archivio di configurazione AWS AppConfig ospitato](#)
- [Informazioni sulle configurazioni archiviate in Amazon S3](#)
- [Creazione di un profilo AWS AppConfig di configurazione in formato libero \(console\)](#)
- [Creazione di un profilo di configurazione in AWS AppConfig formato libero \(riga di comando\)](#)

## Comprendere i validatori

Quando crei un profilo di configurazione, hai la possibilità di specificare fino a due validatori. Un validatore assicura che i dati di configurazione siano sintatticamente e semanticamente corretti. Se prevedi di utilizzare un validatore, devi crearlo prima di creare il profilo di configurazione. AWS AppConfig supporta i seguenti tipi di validatori:

- AWS Lambda funzioni: supportate per i contrassegni delle funzionalità e le configurazioni in formato libero.
- Schema JSON: supportato per configurazioni in formato libero. (convalida AWS AppConfig automaticamente i flag delle funzionalità rispetto a uno schema JSON.)

## Argomenti

- [AWS Lambda validatori di funzioni](#)
- [validatori dello schema JSON](#)

## AWS Lambda validatori di funzioni

I validatori delle funzioni Lambda devono essere configurati con il seguente schema di eventi. AWS AppConfig utilizza questo schema per richiamare la funzione Lambda. Il contenuto è una stringa con codifica base64 e l'uri è una stringa.

```
{  
  "applicationId": "The application ID of the configuration profile being validated",  
  "configurationProfileId": "The ID of the configuration profile being validated",  
  "configurationVersion": "The version of the configuration profile being validated",  
  "content": "Base64EncodedByteString",  
  "uri": "The configuration uri"  
}
```

AWS AppConfig verifica che l'intestazione X-Amz-Function-Error Lambda sia impostata nella risposta. Lambda imposta questa intestazione se la funzione genera un'eccezione. Per ulteriori informazioni in merito *X-Amz-Function-Error*, consulta la sezione [Gestione degli errori e tentativi automatici AWS Lambda nella Guida](#) per gli sviluppatori AWS Lambda

Ecco un semplice esempio di codice di risposta Lambda per una convalida corretta.

```
import json  
  
def handler(event, context):  
    #Add your validation logic here  
    print("We passed!")
```

Ecco un semplice esempio di codice di risposta Lambda per una convalida non riuscita.

```
def handler(event, context):  
    #Add your validation logic here  
    raise Exception("Failure!")
```

Ecco un altro esempio che convalida solo se il parametro di configurazione è un numero primo.

```
function isPrime(value) {  
  if (value < 2) {  
    return false;  
  }  
  
  for (i = 2; i < value; i++) {  
    if (value % i === 0) {  
      return false;  
    }  
  }  
}
```

```
    return true;
}

exports.handler = async function(event, context) {
  console.log('EVENT: ' + JSON.stringify(event, null, 2));
  const input = parseInt(Buffer.from(event.content, 'base64').toString('ascii'));
  const prime = isPrime(input);
  console.log('RESULT: ' + input + (prime ? ' is' : ' is not') + ' prime');
  if (!prime) {
    throw input + "is not prime";
  }
}
```

AWS AppConfig chiama la tua Lambda di convalida quando richiama `StartDeployment` le operazioni `ValidateConfigurationActivity` e API. Devi fornire `appconfig.amazonaws.com` le autorizzazioni per richiamare la tua Lambda. Per ulteriori informazioni, consulta [Concessione dell'accesso alle funzioni ai servizi](#). AWS AppConfig limita il tempo di esecuzione Lambda di convalida a 15 secondi, inclusa la latenza di avvio.

### validatori dello schema JSON

Se crei una configurazione in un documento SSM, devi specificare o creare uno schema JSON per la configurazione. Uno schema JSON definisce le proprietà consentite per ogni impostazione di configurazione dell'applicazione. Lo schema JSON funziona come un set di regole per garantire che le impostazioni di configurazione nuove o aggiornate siano conformi alle best practice richieste dall'applicazione. Ecco un esempio.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "$id$",
  "description": "BasicFeatureToggle-1",
  "type": "object",
  "additionalProperties": false,
  "patternProperties": {
    "[^\\s]+$": {
      "type": "boolean"
    }
  },
  "minProperties": 1
}
```

Quando si crea una configurazione da un documento SSM, il sistema verifica automaticamente che la configurazione sia conforme ai requisiti dello schema. In caso contrario, AWS AppConfig restituisce un errore di convalida.

### Important

Nota le seguenti informazioni importanti sui validatori dello schema JSON:

- I dati di configurazione archiviati nei documenti SSM devono essere convalidati rispetto a uno schema JSON associato prima di poter aggiungere la configurazione al sistema. I parametri SSM non richiedono un metodo di convalida, ma si consiglia di creare un controllo di convalida per le configurazioni dei parametri SSM nuove o aggiornate utilizzando AWS Lambda
- Una configurazione in un documento SSM utilizza il tipo di documento `ApplicationConfiguration`. Lo schema JSON corrispondente utilizza il tipo di documento `ApplicationConfigurationSchema`.
- AWS AppConfig supporta lo schema JSON versione 4.X per lo schema in linea. Se la configurazione dell'applicazione richiede una versione diversa di JSON Schema, devi creare un validatore Lambda.

## Comprensione delle quote e delle limitazioni dell'archivio di configurazione

Gli archivi di configurazione supportati da AWS AppConfig hanno le seguenti quote e limitazioni.

	AWS AppConfig archivio di configurazione ospitato	Simple Storage Service (Amazon S3)	Systems Manager Parameter Store	Gestione dei segreti AWS	Systems Manager, archivio documenti	AWS CodePipeline
Limite di dimensione della configurazione	2 MB per impostazione predefinita	2 MB Applicato da AWS	4 KB (piano gratuito) /8 KB	64 KB	64 KB	2 MB Applicata, non AWS

	AWS AppConfig archivio di configurazione ospitato	Simple Storage Service (Amazon S3)	Systems Manager Parameter Store	Gestione dei segreti AWS	Systems Manager, archivio documenti	AWS CodePipeline
	ta, 4 MB massimi	AppConfig, non da S3	(parametri avanzati)			AppConfig CodePipeline
Limite di storage delle risorse	1 GB	Illimitato	10.000 parametri (piano gratuito) /100.000 parametri (parametri avanzati)	500.000	500 documenti	Limitato dal numero di profili di configurazione per applicazione (100 profili per applicazione)
Crittografia lato server	Sì	<a href="#">SSE-S3, SSE-KMS</a>	Sì	Sì	No	Sì
CloudFormation supporto	Sì	Non per la creazione o l'aggiornamento dei dati	Sì	Sì	No	Sì
Prezzi	Gratuito	<a href="#">Vedi i prezzi di Amazon S3</a>	<a href="#">Vedi i prezzi AWS Systems Manager</a>	<a href="#">Vedi Gestione dei segreti AWS i prezzi</a>	Gratuito	<a href="#">Vedi AWS CodePipeline i prezzi</a>

## Informazioni sull'archivio di configurazione AWS AppConfig ospitato

AWS AppConfig include un archivio di configurazione interno o ospitato. Le configurazioni devono avere una dimensione pari o inferiore a 2 MB. L'archivio di configurazione AWS AppConfig ospitato offre i seguenti vantaggi rispetto ad altre opzioni dell'archivio di configurazione.

- Non è necessario impostare e configurare altri servizi come Amazon Simple Storage Service (Amazon S3) o Archivio parametri.
- Non è necessario configurare le autorizzazioni AWS Identity and Access Management (IAM) per utilizzare l'archivio di configurazione.
- Puoi memorizzare le configurazioni in YAML, JSON o come documenti di testo.
- L'utilizzo dell'archivio è gratuito.
- Puoi creare una configurazione e aggiungerla all'archivio quando crei un profilo di configurazione.

## Informazioni sulle configurazioni archiviate in Amazon S3

Puoi archiviare le configurazioni in un bucket Amazon Simple Storage Service (Amazon S3). Quando si crea il profilo di configurazione, si specifica l'URI di un singolo oggetto S3 in un bucket. È inoltre necessario specificare l'Amazon Resource Name (ARN) di un ruolo AWS Identity and Access Management (IAM) che AWS AppConfig autorizza a ottenere l'oggetto. Prima di creare un profilo di configurazione per un oggetto Amazon S3, tieni presente le seguenti restrizioni.

Limitazione	Informazioni
Dimensione	Le configurazioni archiviate come oggetti S3 possono avere una dimensione massima di 1 MB.
Crittografia degli oggetti	Un profilo di configurazione può avere come target oggetti crittografati SSE-S3 e SSE-KMS.
Classi di archiviazione	AWS AppConfig supporta le seguenti classi di storage S3:,,, e. STANDARD INTELLIGENT_TIERING REDUCED_REDUNDANCY STANDARD_IA ONEZONE_IA Le seguenti classi non sono supportate: tutte le classi S3 Glacier (GLACIER e DEEP_ARCHIVE ).

Limitazione	Informazioni
Controllo delle versioni	AWS AppConfig richiede che l'oggetto S3 utilizzi il controllo delle versioni.

Configurazione delle autorizzazioni per una configurazione archiviata come oggetto Amazon S3

Quando crei un profilo di configurazione per una configurazione archiviata come oggetto S3, devi specificare un ARN per un ruolo IAM che AWS AppConfig dia il permesso di ottenere l'oggetto. Il ruolo deve includere le seguenti autorizzazioni:

Autorizzazioni per accedere all'oggetto S3

- s3: GetObject
- s3: GetObjectVersion

Autorizzazioni per elencare i bucket S3

s3: ListAllMyBuckets

Autorizzazioni per accedere al bucket S3 in cui è archiviato l'oggetto

- s3: GetBucketLocation
- s3: GetBucketVersioning
- s3: ListBucket
- s3: ListBucketVersions

Completa la seguente procedura per creare un ruolo che AWS AppConfig consenta di archiviare una configurazione in un oggetto S3.

Creazione della policy IAM per l'accesso a un oggetto S3

Utilizza la seguente procedura per creare una policy IAM che AWS AppConfig consenta di archiviare una configurazione in un oggetto S3.

Per creare una policy IAM per l'accesso a un oggetto S3

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.

2. Nel pannello di navigazione, scegli Policy e Crea policy.
3. Nella pagina Crea policy, scegli la scheda JSON.
4. Aggiornare la policy di esempio riportata di seguito con informazioni sul bucket S3 e sull'oggetto di configurazione. Quindi incollare il criterio nel campo di testo nella scheda JSON. Sostituisci *placeholder values* con le informazioni appropriate.

### JSON

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetObject",  
                "s3:GetObjectVersion"  
            ],  
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/my-configurations/my-  
configuration.json"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetBucketLocation",  
                "s3:GetBucketVersioning",  
                "s3>ListBucketVersions",  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::amzn-s3-demo-bucket"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": "s3>ListAllMyBuckets",  
            "Resource": "*"  
        }  
    ]  
}
```

5. Scegliere Esamina policy.
6. Nella pagina Review (Revisione) immettere un nome nella casella Name (Nome), quindi digitare una descrizione.
7. Scegli Crea policy. Il sistema visualizza di nuovo la pagina Ruoli.

## Creazione del ruolo IAM per l'accesso a un oggetto S3

Utilizza la seguente procedura per creare un ruolo IAM che AWS AppConfig consenta di archiviare una configurazione in un oggetto S3.

### Per creare un ruolo IAM per l'accesso a un oggetto Amazon S3

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione, scegli Ruoli e quindi Crea ruolo.
3. Nella sezione Seleziona il tipo di entità affidabile, scegli il AWS servizio.
4. Nella sezione Scegli un caso d'uso, in Casi d'uso comuni, scegli EC2, quindi scegli Avanti: Autorizzazioni.
5. Nella pagina Allega criteri autorizzazioni, nella casella di ricerca, immettere il nome del criterio creato nella procedura precedente.
6. Scegliere questa policy, quindi selezionare Next: Tags.
7. Nella pagina Aggiungi tag (opzionale), inserisci una chiave e un valore opzionale, quindi scegli Avanti: revisione.
8. Nella pagina Review (Revisione) immettere un nome nella casella Role name (Nome ruolo), quindi digitare una descrizione.
9. Scegliere Create role (Crea ruolo). Il sistema visualizza di nuovo la pagina Roles (Ruoli).
10. Nella pagina Ruoli, scegli il ruolo appena creato per aprire la pagina Riepilogo. Annotare i valori per Role Name (Nome ruolo) e Role ARN (ARN ruolo). Verrà specificato il ruolo ARN quando si crea il profilo di configurazione più avanti in questo argomento.

## Creazione di una relazione di trust

Utilizza la procedura seguente per configurare il ruolo appena creato per considerare attendibile AWS AppConfig.

Per aggiungere una relazione di trust

1. Nella pagina Summary (Riepilogo) per il ruolo creato in precedenza, scegliere la scheda Trust Relationships (Relazioni di trust), quindi scegliere Edit Trust Relationship (Modifica relazione di trust).
2. Eliminare "ec2.amazonaws.com" e aggiungere "appconfig.amazonaws.com" come mostrato nell'esempio seguente.

JSON

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

3. Scegli Update Trust Policy (Aggiorna policy di trust).

Creazione di un profilo AWS AppConfig di configurazione in formato libero (console)

Utilizzate la procedura seguente per creare un profilo di configurazione in AWS AppConfig formato libero e (facoltativamente) una configurazione in formato libero utilizzando la console. AWS Systems Manager

Per creare un profilo di configurazione a forma libera

1. Apri la AWS Systems Manager console all'indirizzo <https://console.aws.amazon.com/systems-manager/appconfig/>.
2. Nel riquadro di navigazione, scegli Applicazioni, quindi scegli un'applicazione in cui hai creato. [Creazione di uno spazio dei nomi per l'applicazione in AWS AppConfig](#)
3. Scegliete la scheda Profili di configurazione e bandiere di funzionalità, quindi scegliete Crea configurazione.

4. Nella sezione Opzioni di configurazione, scegli Configurazione in formato libero.
5. Per Nome del profilo di configurazione, inserisci un nome per il profilo di configurazione.
6. (Facoltativo) Espandi Descrizione e inserisci una descrizione.
7. (Facoltativo) Espandi Opzioni aggiuntive e completa quanto segue, se necessario.
  - a. Nella sezione Associa estensioni, scegliete un'estensione dall'elenco.
  - b. Nella sezione Tag, scegli Aggiungi nuovo tag, quindi specifica una chiave e un valore opzionale.
8. Scegli Next (Successivo).
9. Nella pagina Specificare i dati di configurazione, nella sezione Definizione della configurazione, scegli un'opzione.
10. Completate i campi per l'opzione selezionata, come descritto nella tabella seguente.

Opzione selezionata	Informazioni
AWS AppConfig configurazione ospitata	Scegli Text, JSON o YAML e inserisci la tua configurazione nel campo. Vai al passaggio 12 di questa procedura.
Oggetto Amazon S3	Inserisci l'URI dell'oggetto nel campo di origine dell'oggetto S3 e vai al passaggio 11 di questa procedura.
AWS CodePipeline	Scegli Avanti e vai al passo 12 di questa procedura.
Segreto di Secrets Manager	Scegli il segreto dall'elenco, vai al passaggio 11 di questa procedura.
AWS Systems Manager parameter	Scegli il parametro dall'elenco e vai al Passaggio 11 di questa procedura.
AWS Systems Manager documento	<ol style="list-style-type: none"><li>1. Scegliete un documento dall'elenco o scegliete Crea nuovo documento.</li><li>2. Se scegli Crea nuovo documento, per Nome documento, inserisci un nome. Facoltativamente, espandi Nome della</li></ol>

Opzione selezionata	Informazioni
	<p>versione e inserisci un nome per la versione del documento.</p> <p>3. Per lo schema di configurazione dell'applicazione, scegli lo schema JSON dall'elenco o scegli Crea schema. Se si sceglie Crea schema, Systems Manager apre la pagina Crea schema. Immettete i dettagli dello schema, quindi scegliete Crea schema di configurazione dell'applicazione.</p> <p>4. Nella sezione Contenuto scegliere YAML o JSON e quindi immettere i dati di configurazione nel campo.</p>

11. Nella sezione Ruolo di servizio, scegli Nuovo ruolo di servizio per AWS AppConfig creare il ruolo IAM che fornisce l'accesso ai dati di configurazione. AWS AppConfig compila automaticamente il campo Nome del ruolo in base al nome inserito in precedenza. In alternativa, scegli Ruolo di servizio esistente. Scegliere il ruolo utilizzando l'elenco Role ARN (Arn ruolo).
12. Facoltativamente, nella pagina Aggiungi validatori, scegli Schema JSON o. AWS Lambda Se si sceglie JSON Scheme (Schema JSON), immettere lo schema JSON nel campo. Se scegli AWS Lambda, scegli la funzione Amazon Resource Name (ARN) e la versione dall'elenco.

 **Important**

I dati di configurazione archiviati nei documenti SSM devono essere convalidati rispetto a uno schema JSON associato prima di poter aggiungere la configurazione al sistema. I parametri SSM non richiedono un metodo di convalida, ma ti consigliamo di creare un controllo di convalida per configurazioni di parametri SSM nuove o aggiornate utilizzando AWS Lambda

13. Scegli Next (Successivo).
14. Nella pagina Rivedi e salva, scegli Salva e continua la distribuzione.

### ⚠️ Important

Se hai creato un profilo di configurazione per AWS CodePipeline, devi creare una pipeline in CodePipeline cui sia specificato AWS AppConfig come provider di distribuzione. Non è necessario eseguire [Distribuzione di flag di funzionalità e dati di configurazione in AWS AppConfig](#). Tuttavia, è necessario configurare un client per ricevere gli aggiornamenti della configurazione dell'applicazione come descritto in [Recupero dei dati di configurazione senza Agent AWS AppConfig](#). Per informazioni sulla creazione di una pipeline che viene specificata AWS AppConfig come provider di distribuzione, vedi [Tutorial: Create a Pipeline that Uses AWS AppConfig as a Deployment Provider](#) nella Guida per l'utente AWS CodePipeline.

Passa a [Distribuzione di flag di funzionalità e dati di configurazione in AWS AppConfig](#).

Creazione di un profilo di configurazione in AWS AppConfig formato libero (riga di comando)

La procedura seguente descrive come utilizzare AWS CLI (su Linux o Windows) o AWS Strumenti per PowerShell creare un profilo di configurazione in formato libero. AWS AppConfig Se preferisci, puoi usare AWS CloudShell per eseguire i comandi elencati di seguito. Per ulteriori informazioni, consulta [Che cos'è AWS CloudShell?](#) nella Guida per l'utente di AWS CloudShell .

### ℹ️ Note

Per le configurazioni in formato libero ospitate nell'archivio di configurazione AWS AppConfig ospitato, è necessario specificare l'hostedURI di posizione.

Per creare un profilo di configurazione utilizzando AWS CLI

1. Aprire il AWS CLI.
2. Esegui il comando seguente per creare un profilo di configurazione in formato libero.

Linux

```
aws appconfig create-configuration-profile \
--application-id APPLICATION_ID \
--name NAME \
--description CONFIGURATION_PROFILE_DESCRIPTION \
```

```
--location-uri CONFIGURATION_URI or hosted \
--retrieval-role-arn IAM_ROLE_ARN \
--tags TAGS \
--validators "Content=SCHEMA_CONTENT or LAMBDA_FUNCTION_ARN,Type=JSON_SCHEMA
or LAMBDA"
```

## Windows

```
aws appconfig create-configuration-profile ^
--application-id APPLICATION_ID ^
--name NAME ^
--description CONFIGURATION_PROFILE_DESCRIPTION ^
--location-uri CONFIGURATION_URI or hosted ^
--retrieval-role-arn IAM_ROLE_ARN ^
--tags TAGS ^
--validators "Content=SCHEMA_CONTENT or LAMBDA_FUNCTION_ARN,Type=JSON_SCHEMA
or LAMBDA"
```

## PowerShell

```
New-APPConfigurationProfile ` 
-Name NAME ` 
-ApplicationId APPLICATION_ID ` 
-Description CONFIGURATION_PROFILE_DESCRIPTION ` 
-LocationUri CONFIGURATION_URI or hosted ` 
-RetrievalRoleArn IAM_ROLE_ARN ` 
-Tag TAGS ` 
-Validators "Content=SCHEMA_CONTENT or LAMBDA_FUNCTION_ARN,Type=JSON_SCHEMA
or LAMBDA"
```

### ⚠ Important

Prendi nota delle seguenti informazioni importanti.

- Se hai creato un profilo di configurazione per AWS CodePipeline, devi creare una pipeline in CodePipeline cui sia specificato AWS AppConfig come provider di distribuzione. Non è necessario eseguire [Distribuzione di flag di funzionalità e dati di configurazione in AWS AppConfig](#). Tuttavia, è necessario configurare un client per ricevere gli aggiornamenti della configurazione dell'applicazione come descritto in [Recupero dei dati di configurazione senza Agent AWS AppConfig](#). Per informazioni sulla creazione di una pipeline che viene

specificata AWS AppConfig come provider di distribuzione, vedi [Tutorial: Create a Pipeline that Uses AWS AppConfig as a Deployment Provider](#) nella Guida per l'utente AWS CodePipeline

- Se hai creato una configurazione nell'archivio di configurazione AWS AppConfig ospitato, puoi creare nuove versioni della configurazione utilizzando le operazioni API [CreateHostedConfigurationVersion](#). Per visualizzare AWS CLI i dettagli e i comandi di esempio per questa operazione API, consulta [create-hosted-configuration-version](#) la sezione AWS CLI Command Reference.

Passa a [Distribuzione di flag di funzionalità e dati di configurazione in AWS AppConfig](#).

## Creazione di un profilo di configurazione per fonti di dati non native

AWS AppConfig supporta la distribuzione dei dati di configurazione dalla maggior parte degli archivi di dati. A livello nativo, AWS AppConfig supporta la distribuzione dei dati di configurazione archiviati nei seguenti servizi:

- L'archivio di configurazione AWS AppConfig ospitato
- Simple Storage Service (Amazon S3)
- Gestione dei segreti AWS
- AWS Systems Manager Parameter Store
- Systems Manager Document Store
- AWS CodePipeline

Se i dati di configurazione sono archiviati in una posizione non supportata nativamente da AWS AppConfig, puoi creare un'[AWS AppConfig estensione](#) per recuperare i dati dalla fonte. Ad esempio, utilizzando un' AWS AppConfig estensione, puoi recuperare i dati di configurazione archiviati in Amazon Relational Database Service (Amazon RDS), Amazon DynamoDB (DynamoDB GitHub) o un repository locale GitLab, solo per citarne alcuni. Implementando un'estensione, puoi sfruttare la AWS AppConfig sicurezza e i miglioramenti per le tue applicazioni e DevOps il tuo ambiente informatico. È inoltre possibile utilizzare questo metodo per migrare i dati di configurazione da sistemi legacy a AWS AppConfig.

La creazione di un profilo di configurazione per fonti di dati non supportate nativamente AWS AppConfig implica i seguenti processi o azioni:

1. Crea una [AWS Lambda funzione](#) che recuperi i dati dalla tua fonte di dati. Finché una funzione Lambda può accedere all'origine dati, AWS AppConfig l'estensione sarà in grado di recuperare i dati.
2. Crea un' AWS AppConfig estensione personalizzata che richiami la tua funzione Lambda. Per ulteriori informazioni, consulta [Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig](#).
3. Crea un profilo di configurazione in formato AWS AppConfig libero. In particolare, crea un profilo di configurazione che utilizzi la definizione di configurazione AWS AppConfig ospitata. Il profilo di configurazione funge da archivio dati temporaneo dopo che la funzione Lambda ha recuperato la configurazione dall'origine. L'applicazione recupererà i dati di configurazione dall'archivio di configurazione AWS AppConfig ospitato. Per ulteriori informazioni, consulta [Creazione di un profilo di configurazione in formato libero in AWS AppConfig](#).
4. Crea un'associazione di estensioni che si attiva utilizzando il punto di `PRE_CREATE_HOSTED_CONFIGURATION_VERSION` azione. Per ulteriori informazioni, consulta [Fase 4: Creare un'associazione di estensioni per un'estensione personalizzata AWS AppConfig](#).

Una volta configurata, quando l'applicazione richiede una nuova versione dei dati di configurazione, Lambda recupera i dati di configurazione e li inserisce nel profilo di configurazione. AWS AppConfig quindi salva il profilo di configurazione e i dati di terze parti.

Quando sei pronto, puoi distribuire il profilo di configurazione nelle tue applicazioni, proprio come qualsiasi altro tipo di dati di configurazione.

 Note

Puoi scegliere di inserire dati di terze parti in linea con i dati di configurazione esistenti o fare in modo che l'intero contenuto dei dati di configurazione contenga solo dati di terze parti. Se desideri che i dati siano in linea con altri dati esistenti, tale logica deve far parte della funzione Lambda che importa i dati dalla fonte di terze parti.

## Migrazione AWS AppConfig da servizi di configurazione preesistenti e creati internamente

Se avete iniziato a utilizzare AWS AppConfig e avete ancora dati di configurazione o feature flag preesistenti in un altro sistema, potete utilizzare la procedura descritta in precedenza in questo argomento per migrare dal sistema esistente a un altro sistema. AWS AppConfig È possibile creare

un'estensione che estragga i dati dal sistema legacy e li distribuisca. AWS AppConfig L'utilizzo AWS AppConfig in questo modo ti offre tutti i controlli e i vantaggi del guardrail di sicurezza pur continuando a utilizzare i tuoi archivi di dati legacy.

# Distribuzione di flag di funzionalità e dati di configurazione in AWS AppConfig

Dopo aver [creato gli elementi necessari per l'utilizzo](#) dei flag di funzionalità e dei dati di configurazione in formato libero, è possibile creare una nuova distribuzione. Quando si crea una nuova distribuzione, si specificano le seguenti informazioni:

- Un ID dell'applicazione
- Un ID del profilo di configurazione
- Una versione di configurazione
- Un ID di ambiente in cui si desidera distribuire i dati di configurazione
- Un ID della strategia di implementazione che definisce la velocità con cui desiderate che le modifiche abbiano effetto
- Un ID chiave AWS Key Management Service (AWS KMS) per crittografare i dati utilizzando una chiave gestita dal cliente.

Quando richiami l'azione [StartDeployment](#) API, AWS AppConfig esegue le seguenti attività:

1. Recupera i dati di configurazione dal data store sottostante utilizzando l'URI di posizione nel profilo di configurazione.
2. Verifica che i dati di configurazione siano corretti dal punto di vista sintattico e semantico utilizzando i validatori specificati al momento della creazione del profilo di configurazione.
3. Memorizza nella cache una copia dei dati in modo che sia pronta per essere recuperata dall'applicazione. Questa copia memorizzata nella cache è denominata dati distribuiti.

Puoi mitigare le situazioni in cui la distribuzione dei dati di configurazione causa errori nell'applicazione utilizzando una combinazione di strategie di AWS AppConfig distribuzione e rollback automatici basati sugli allarmi di Amazon CloudWatch. Una strategia di distribuzione consente di rilasciare lentamente le modifiche agli ambienti di produzione nell'arco di minuti o ore. Una volta configurato, se uno o più CloudWatch allarmi entrano nello stato di allarme durante una distribuzione, ripristina AWS AppConfig automaticamente i dati di configurazione alla versione precedente. Per ulteriori informazioni sulle strategie di distribuzione, consulta [Utilizzo delle strategie di distribuzione](#). Per ulteriori informazioni sui rollback automatici, vedere [Monitoraggio delle implementazioni per il rollback automatico](#).

## Argomenti

- [Utilizzo delle strategie di distribuzione](#)
- [Distribuzione di una configurazione](#)
- [Distribuzione delle configurazioni utilizzando AWS AppConfig CodePipeline](#)
- [Ripristino di una configurazione](#)

## Utilizzo delle strategie di distribuzione

Una strategia di implementazione consente di rilasciare lentamente le modifiche agli ambienti di produzione nell'arco di minuti o ore. Una strategia AWS AppConfig di implementazione definisce i seguenti aspetti importanti di una distribuzione di configurazione.

Impostazione	Descrizione
Il tipo di distribuzione	<p>Il tipo di distribuzione definisce il modo in cui la configurazione viene distribuita o implementata. AWS AppConfig supporta i tipi di distribuzione lineare ed esponenziale.</p> <ul style="list-style-type: none"><li>• Lineare: per questo tipo, AWS AppConfig elabora l'implementazione mediante incrementi del fattore di crescita distribuiti uniformemente nell'implementazione. Ecco un esempio di cronologia per un'implementazione di 10 ore che utilizza una crescita lineare del 20%:</li></ul>
Tempo trascorso	Avanzamento della distribuzione
0 ore	0%
2 ore	20%
4 ore	40%

Impostazione	Descrizione	
	Tempo trascorso	Avanzamento della distribuzione
	6 ore	60%
	8 ore	80%
	10 ore	100%

- Esponenziale: per questo tipo, AWS AppConfig elabora la distribuzione in modo esponenziale utilizzando la seguente formula:  $G * (2^N)$ . In questa formula, G è la percentuale di fasi specificata dall'utente e N è il numero di fasi fino a quando la configurazione viene distribuita a tutte le destinazioni. Ad esempio, se si specifica un fattore di crescita di 2, il sistema esegue la configurazione come segue:

$$2 * (2^0)$$

$$2 * (2^1)$$

$$2 * (2^2)$$

Espressa numericamente, la distribuzione viene eseguita come segue: 2% degli obiettivi, 4% degli obiettivi, 8% degli obiettivi e continua fino a quando la configurazione non è stata distribuita su tutti i target.

Impostazione	Descrizione
Percentuale di fasi (fattore di crescita)	<p>Questa impostazione specifica la percentuale di chiamanti da destinare durante ogni fase della distribuzione</p> <p> <b>Note</b> Nell'SDK e nella <a href="#">Documentazione di riferimento delle API AWS AppConfig</a>, <code>step percentage</code> viene chiamato <code>growth factor</code>.</p>
Tempo di distribuzione	<p>Questa impostazione specifica il periodo di tempo durante il quale viene eseguita la AWS AppConfig distribuzione sugli host. Questo non è un valore di timeout. Si tratta di una finestra temporale durante la quale la distribuzione viene elaborata a intervalli.</p>
Tempo	<p>Questa impostazione specifica la quantità di tempo di AWS AppConfig monitoraggio degli CloudWatch allarmi Amazon dopo che la configurazione è stata distribuita al 100% dei suoi obiettivi, prima di considerare la distribuzione completa. Se durante questo periodo viene attivato un avviso, eseguire il rollback AWS AppConfig della distribuzione. È necessario configurare le autorizzazioni per eseguire il rollback in base AWS AppConfig agli allarmi. CloudWatch Per ulteriori informazioni, consulta <a href="#">Configura le autorizzazioni per il rollback automatico</a>.</p>

Puoi scegliere una strategia predefinita inclusa AWS AppConfig o crearne una personalizzata.

## Argomenti

- [Utilizzo di strategie di distribuzione predefinite](#)
- [Creazione di una strategia di distribuzione](#)

## Utilizzo di strategie di distribuzione predefinite

AWS AppConfig include strategie di distribuzione predefinite per aiutarti a implementare rapidamente una configurazione. Anziché creare strategie personalizzate, puoi scegliere una delle seguenti opzioni quando distribuisci una configurazione.

Strategia di distribuzione	Descrizione
AppConfig.Lineare 20 6 minuti PercentEvery	<p>AWS consigliato:</p> <p>Questa strategia implementa la configurazione sul 20% di tutti gli obiettivi ogni sei minuti per un'implementazione di 30 minuti. Il sistema monitora gli CloudWatch allarmi Amazon per 30 minuti. Se non vengono ricevuti allarmi in questo periodo, la distribuzione è completata. Se viene attivato un allarme durante questo periodo, ripristina la AWS AppConfig distribuzione.</p> <p>Consigliamo di utilizzare questa strategia per le implementazioni di produzione perché è in linea con le AWS migliori pratiche e include un'enfasi aggiuntiva sulla sicurezza dell'implementazione grazie alla sua lunga durata e ai tempi di cottura.</p>
AppConfig. 10 percento e 20 minuti	<p>AWS consigliato:</p> <p>questa strategia elabora la distribuzione in modo esponenziale utilizzando un fattore di crescita del 10% in 20 minuti. Il sistema monitora la presenza di CloudWatch allarmi</p>

Strategia di distribuzione	Descrizione
	<p>per 10 minuti. Se non vengono ricevuti allarmi in questo periodo, la distribuzione è completata. Se viene attivato un allarme durante questo periodo, ripristina la AWS AppConfig distribuzione.</p> <p>Consigliamo di utilizzare questa strategia per le implementazioni di produzione perché è in linea con le AWS migliori pratiche per le implementazioni di configurazione.</p>
AppConfig.AllAtOnce	<p>Rapidità:</p> <p>questa strategia distribuisce immediatamente la configurazione a tutte le destinazioni. Il sistema monitora gli allarmi per 10 minuti. CloudWatch Se non vengono ricevuti allarmi in questo periodo, la distribuzione è completata. Se durante questo periodo viene attivato un avviso, eseguire il rollback AWS AppConfig della distribuzione.</p>

Strategia di distribuzione	Descrizione
AppConfig. Lineare 50 30 secondi PercentEvery	<p>Test/dimostrazione:</p> <p>questa strategia distribuisce la configurazione a metà di tutte le destinazioni ogni 30 secondi per una distribuzione di un minuto. Il sistema monitora gli CloudWatch allarmi Amazon per 1 minuto. Se non vengono ricevuti allarmi in questo periodo, la distribuzione è completata. Se viene attivato un allarme durante questo periodo, ripristina la AWS AppConfig distribuzione.</p> <p>Ti consigliamo di utilizzare questa strategia solo a scopo di test o dimostrazione perché ha una breve durata e tempo di bake.</p>

## Creazione di una strategia di distribuzione

Se non desideri utilizzare una delle strategie di distribuzione predefinite, puoi crearne una personalizzata. È possibile creare un massimo di 20 strategie di distribuzione. Quando si distribuisce una configurazione, è possibile scegliere la strategia di distribuzione più adatta all'applicazione e all'ambiente.

### Creazione di una strategia AWS AppConfig di implementazione (console)

Utilizzare la procedura seguente per creare una strategia di AWS AppConfig distribuzione utilizzando la AWS Systems Manager console.

Per creare una strategia di distribuzione

1. Apri la AWS Systems Manager console all'indirizzo <https://console.aws.amazon.com/systems-manager/appconfig/>.
2. Nel riquadro di navigazione, scegli Strategie di distribuzione, quindi scegli Crea strategia di distribuzione.
3. In Name (Nome), immettere un nome per la strategia di distribuzione.

4. In Description (Descrizione), immettere informazioni sulla strategia di distribuzione.
5. In Tipo di distribuzione, scegliere un tipo.
6. In Step percentage (Percentuale fase), scegliere la percentuale di chiamanti da destinare durante ogni passaggio della distribuzione.
7. In Deployment time (Tempo di distribuzione), immettere la durata totale della distribuzione espressa in minuti o ore.
8. Per Bake time, inserisci il tempo totale, in minuti o ore, per monitorare gli CloudWatch allarmi di Amazon prima di procedere alla fase successiva di una distribuzione o prima di considerare che la distribuzione sia completa.
9. Nella sezione Tags (Tag) immettere una chiave e un valore facoltativo. È possibile specificare un massimo di 50 tag per una risorsa.
10. Scegliere Create deployment strategy (Crea strategia di distribuzione).

 **Important**

Se hai creato un profilo di configurazione per AWS CodePipeline, devi creare una pipeline CodePipeline che specifichi AWS AppConfig come provider di distribuzione. Non è necessario eseguire [Distribuzione di una configurazione](#). Tuttavia, è necessario configurare un client per ricevere gli aggiornamenti della configurazione dell'applicazione come descritto in [Recupero dei dati di configurazione senza Agent AWS AppConfig](#). Per informazioni sulla creazione di una pipeline che viene specificata AWS AppConfig come provider di distribuzione, vedi [Tutorial: Create a Pipeline that Uses AWS AppConfig as a Deployment Provider](#) nella Guida per l'utente AWS CodePipeline

Passa a [Distribuzione di una configurazione](#).

## Creazione di una strategia di AWS AppConfig distribuzione (riga di comando)

La procedura seguente descrive come utilizzare AWS CLI (su Linux o Windows) o AWS Strumenti per PowerShell creare una strategia di AWS AppConfig distribuzione.

Per creare una strategia di distribuzione passo dopo passo

1. Aprire il AWS CLI.
2. Esegui il comando seguente per creare una strategia di distribuzione.

## Linux

```
aws appconfig create-deployment-strategy \
--name A_name_for_the_deployment_strategy \
--description A_description_of_the_deployment_strategy \
--deployment-duration-in-minutes Total_amount_of_time_for_a_deployment_to_last
\
--final-bake-time-in-minutes Amount_of_time_AWS
AppConfig_monitors_for_alarms_before_considering_the_deployment_to_be_complete
\
--growth-
factor The_percentage_of_targets_to_receive_a_deployed_configuration_during_each_interval
\
--growth-
type The_linear_or_exponential_algorithm_used_to_define_how_percentage_grows_over_time
\
--replicate-
to To_save_the_deployment_strategy_to_a_Systems_Manager_(SSM)_document \
--tags User_defined_key_value_pair_metadata_of_the_deployment_strategy
```

## Windows

```
aws appconfig create-deployment-strategy ^
--name A_name_for_the_deployment_strategy ^
--description A_description_of_the_deployment_strategy ^
--deployment-duration-in-minutes Total_amount_of_time_for_a_deployment_to_last
 ^
--final-bake-time-in-minutes Amount_of_time_AWS
AppConfig_monitors_for_alarms_before_considering_the_deployment_to_be_complete
 ^
--growth-
factor The_percentage_of_targets_to_receive_a_deployed_configuration_during_each_interval
 ^
--growth-
type The_linear_or_exponential_algorithm_used_to_define_how_percentage_grows_over_time
 ^
--name A_name_for_the_deployment_strategy ^
--replicate-
to To_save_the_deployment_strategy_to_a_Systems_Manager_(SSM)_document ^
--tags User_defined_key_value_pair_metadata_of_the_deployment_strategy
```

## PowerShell

```
New-APPCDeploymentStrategy
  --Name A_name_for_the_deployment_strategy
  --Description A_description_of_the_deployment_strategy
  --DeploymentDurationInMinutes Total_amount_of_time_for_a_deployment_to_last
  --FinalBakeTimeInMinutes Amount_of_time_AWS
  AppConfig_monitors_for_alarms_before_considering_the_deployment_to_be_complete
  --
  --
  GrowthFactor The_percentage_of_targets_to_receive_a_deployed_configuration_during_each_interval
  --
  --
  GrowthType The_linear_or_exponential_algorithm_used_to_define_how_percentage_grows_over_time
  --
  --
  ReplicateTo To_save_the_deployment_strategy_to_a_Systems_Manager_(SSM)_document
  --
  --
  Tag Hashtable_type_User_defined_key_value_pair_metadata_of_the_deployment_strategy
```

Il sistema restituisce informazioni simili alle seguenti.

## Linux

```
{
  "Id": "Id of the deployment strategy",
  "Name": "Name of the deployment strategy",
  "Description": "Description of the deployment strategy",
  "DeploymentDurationInMinutes": "Total amount of time the deployment lasted",
  "GrowthType": "The linear or exponential algorithm used to define how
percentage grew over time",
  "GrowthFactor": "The percentage of targets that received a deployed
configuration during each interval",
  "FinalBakeTimeInMinutes": "The amount of time AWS AppConfig monitored for
alarms before considering the deployment to be complete",
  "ReplicateTo": "The Systems Manager (SSM) document where the deployment
strategy is saved"
}
```

## Windows

```
{  
  "Id": "Id of the deployment strategy",  
  "Name": "Name of the deployment strategy",  
  "Description": "Description of the deployment strategy",  
  "DeploymentDurationInMinutes": "Total amount of time the deployment lasted",  
  "GrowthType": "The linear or exponential algorithm used to define how  
percentage grew over time",  
  "GrowthFactor": "The percentage of targets that received a deployed  
configuration during each interval",  
  "FinalBakeTimeInMinutes": "The amount of time AWS AppConfig monitored for  
alarms before considering the deployment to be complete",  
  "ReplicateTo": "The Systems Manager (SSM) document where the deployment  
strategy is saved"  
}
```

## PowerShell

```
ContentLength          : Runtime of the command  
DeploymentDurationInMinutes : Total amount of time the deployment lasted  
Description          : Description of the deployment strategy  
FinalBakeTimeInMinutes : The amount of time AWS AppConfig monitored for  
alarms before considering the deployment to be complete  
GrowthFactor          : The percentage of targets that received a deployed  
configuration during each interval  
GrowthType            : The linear or exponential algorithm used to define  
how percentage grew over time  
HttpStatuscode        : HTTP Status of the runtime  
Id                   : The deployment strategy ID  
Name                 : Name of the deployment strategy  
ReplicateTo          : The Systems Manager (SSM) document where the  
deployment strategy is saved  
ResponseMetadata      : Runtime Metadata
```

## Distribuzione di una configurazione

Dopo aver [creato gli elementi necessari per](#) lavorare con i flag di funzionalità e i dati di configurazione in formato libero, puoi creare una nuova distribuzione utilizzando l' Console di gestione AWS, l'AWS CLI o l'SDK AWS . L'avvio di una distribuzione richiede l'operazione dell'API AWS AppConfig .

[StartDeployment](#) Questa chiamata include l' AWS AppConfig applicazione, l'ambiente, il profilo IDs di configurazione e (facoltativamente) la versione dei dati di configurazione da distribuire. La chiamata include anche l'ID della strategia di distribuzione da utilizzare, che determina la modalità di distribuzione dei dati di configurazione.

Se distribuisci segreti archiviati in Gestione dei segreti AWS oggetti Amazon Simple Storage Service (Amazon S3) crittografati con una chiave gestita dal cliente o parametri di stringa sicuri archiviati in AWS Systems Manager Parameter Store crittografati con una chiave gestita dal cliente, devi specificare un valore per il parametro `KmsKeyId`. Se la configurazione non è crittografata o è crittografata con un Chiave gestita da AWS, non è necessario specificare un valore per il `KmsKeyId` parametro.

 Note

Il valore specificato `KmsKeyId` deve essere una chiave gestita dal cliente.

Non deve essere necessariamente la stessa chiave che hai usato per crittografare la configurazione.

Quando si avvia una distribuzione con `UnKmsKeyId`, la politica di autorizzazione allegata al principale AWS Identity and Access Management (IAM) deve consentire l'`kms:GenerateDataKey` operazione.

AWS AppConfig monitora la distribuzione su tutti gli host e riporta lo stato. Se una distribuzione fallisce, AWS AppConfig ripristina la configurazione.

 Note

È possibile distribuire solo una configurazione alla volta in un ambiente. Tuttavia, è possibile implementare una configurazione ciascuna in ambienti diversi contemporaneamente.

## Implementa una configurazione (console)

Utilizzare la procedura seguente per distribuire una AWS AppConfig configurazione utilizzando la AWS Systems Manager console.

## Per distribuire una configurazione utilizzando la console

1. Apri la AWS Systems Manager console all'indirizzo <https://console.aws.amazon.com/systems-manager/appconfig/>.
2. Nel riquadro di navigazione, scegli Applicazioni, quindi scegli un'applicazione in cui hai creato. [Creazione di uno spazio dei nomi per l'applicazione in AWS AppConfig](#)
3. Nella scheda Ambienti, compila il pulsante di opzione corrispondente a un ambiente, quindi scegli Visualizza dettagli.
4. Selezionare Start deployment (Avvia distribuzione).
5. Per Configuration (Configurazione), scegliere una configurazione dall'elenco.
6. A seconda dell'origine della configurazione, utilizza l'elenco delle versioni per scegliere la versione che desideri distribuire.
7. In Deployment strategy (Strategia di distribuzione), scegliere una strategia dall'elenco.
8. (Facoltativo) Inserisci una descrizione in Descrizione distribuzione.
9. Per opzioni di crittografia aggiuntive, scegli una AWS Key Management Service chiave dall'elenco.
10. (Facoltativo) Nella sezione Tag, scegli Aggiungi nuovo tag e inserisci una chiave e un valore opzionale. È possibile specificare un massimo di 50 tag per una risorsa.
11. Selezionare Start deployment (Avvia distribuzione).

## Implementa una configurazione (riga di comando)

La procedura seguente descrive come utilizzare AWS CLI (su Linux o Windows) o AWS Strumenti per PowerShell distribuire una configurazione. AWS AppConfig

### Per distribuire una configurazione passo dopo passo

1. Aprire il. AWS CLI
2. Esegui il comando seguente per distribuire una configurazione.

#### Linux

```
aws appconfig start-deployment \
  --application-id The_application_ID \
  --environment-id The_environment_ID \
  --deployment-strategy-id The_deployment_strategy_ID \
```

```
--configuration-profile-id The_configuration_profile_ID \
--configuration-version The_configuration_version_to_deploy \
--description A_description_of_the_deployment \
--tags User_defined_key_value_pair_metadata_of_the_deployment
```

## Windows

```
aws appconfig start-deployment ^
--application-id The_application_ID ^
--environment-id The_environment_ID ^
--deployment-strategy-id The_deployment_strategy_ID ^
--configuration-profile-id The_configuration_profile_ID ^
--configuration-version The_configuration_version_to_deploy ^
--description A_description_of_the_deployment ^
--tags User_defined_key_value_pair_metadata_of_the_deployment
```

## PowerShell

```
Start-APPCDeployment
-ApplicationId The_application_ID
-ConfigurationProfileId The_configuration_profile_ID
-ConfigurationVersion The_configuration_version_to_deploy
-DeploymentStrategyId The_deployment_strategy_ID
-Description A_description_of_the_deployment
-EnvironmentId The_environment_ID
-Tag Hashtable_type_user_defined_key_value_pair_metadata_of_the_deployment
```

Il sistema restituisce informazioni simili alle seguenti.

## Linux

```
{
  "ApplicationId": "The ID of the application that was deployed",
  "EnvironmentId" : "The ID of the environment",
  "DeploymentStrategyId": "The ID of the deployment strategy that was
deployed",
  "ConfigurationProfileId": "The ID of the configuration profile that was
deployed",
  "DeploymentNumber": The sequence number of the deployment,
  "ConfigurationName": "The name of the configuration",
```

```
"ConfigurationLocationUri": "Information about the source location of the configuration",
  "ConfigurationVersion": "The configuration version that was deployed",
  "Description": "The description of the deployment",
  "DeploymentDurationInMinutes": Total amount of time the deployment lasted,
  "GrowthType": "The linear or exponential algorithm used to define how percentage grew over time",
  "GrowthFactor": The percentage of targets to receive a deployed configuration during each interval,
  "FinalBakeTimeInMinutes": Time AWS AppConfig monitored for alarms before considering the deployment to be complete,
  "State": "The state of the deployment",

  "EventLog": [
    {
      "Description": "A description of the deployment event",
      "EventType": "The type of deployment event",
      "OccurredAt": The date and time the event occurred,
      "TriggeredBy": "The entity that triggered the deployment event"
    }
  ],
  "PercentageComplete": The percentage of targets for which the deployment is available,
  "StartedAt": The time the deployment started,
  "CompletedAt": The time the deployment completed
}
```

## Windows

```
{
  "ApplicationId": "The ID of the application that was deployed",
  "EnvironmentId": "The ID of the environment",
  "DeploymentStrategyId": "The ID of the deployment strategy that was deployed",
  "ConfigurationProfileId": "The ID of the configuration profile that was deployed",
  "DeploymentNumber": The sequence number of the deployment,
  "ConfigurationName": "The name of the configuration",
  "ConfigurationLocationUri": "Information about the source location of the configuration",
  "ConfigurationVersion": "The configuration version that was deployed",
  "Description": "The description of the deployment",
```

```

    "DeploymentDurationInMinutes": Total amount of time the deployment lasted,
    "GrowthType": "The linear or exponential algorithm used to define how
percentage grew over time",
    "GrowthFactor": The percentage of targets to receive a deployed configuration
during each interval,
    "FinalBakeTimeInMinutes": Time AWS AppConfig monitored for alarms before
considering the deployment to be complete,
    "State": "The state of the deployment",

    "EventLog": [
        {
            "Description": "A description of the deployment event",
            "EventType": "The type of deployment event",
            "OccurredAt": The date and time the event occurred,
            "TriggeredBy": "The entity that triggered the deployment event"
        }
    ],
    "PercentageComplete": The percentage of targets for which the deployment is
available,
    "StartedAt": The time the deployment started,
    "CompletedAt": The time the deployment completed
}

```

## PowerShell

ApplicationId	: The ID of the application that was deployed
CompletedAt	: The time the deployment completed
ConfigurationLocationUri	: Information about the source location of the configuration
ConfigurationName	: The name of the configuration
ConfigurationProfileId	: The ID of the configuration profile that was deployed
ConfigurationVersion	: The configuration version that was deployed
ContentLength	: Runtime of the deployment
DeploymentDurationInMinutes	: Total amount of time the deployment lasted
DeploymentNumber	: The sequence number of the deployment
DeploymentStrategyId	: The ID of the deployment strategy that was deployed
Description	: The description of the deployment
EnvironmentId	: The ID of the environment that was deployed

```
EventLog           : {Description : A description of the deployment
event, EventType : The type of deployment event, OccurredAt : The date and time
the event occurred,
                    TriggeredBy : The entity that triggered the deployment event}
FinalBakeTimeInMinutes : Time AWS AppConfig monitored for alarms before
considering the deployment to be complete
GrowthFactor       : The percentage of targets to receive a deployed
configuration during each interval
GrowthType         : The linear or exponential algorithm used to define
how percentage grew over time
HttpStatusCode     : HTTP Status of the runtime
PercentageComplete : The percentage of targets for which the deployment
is available
ResponseMetadata   : Runtime Metadata
StartedAt          : The time the deployment started
State              : The state of the deployment
```

## Distribuzione delle configurazioni utilizzando AWS AppConfig CodePipeline

AWS AppConfig è un'azione di implementazione integrata per (. AWS CodePipeline CodePipeline CodePipeline è un servizio di distribuzione continua completamente gestito che consente di automatizzare le pipeline di rilascio per aggiornamenti rapidi e affidabili di applicazioni e infrastrutture. CodePipeline automatizza le fasi di compilazione, test e distribuzione del processo di rilascio ogni volta che viene apportata una modifica al codice, in base al modello di rilascio definito dall'utente. Per ulteriori informazioni, consulta [What is AWS CodePipeline?](#)

L'integrazione di AWS AppConfig with CodePipeline offre i seguenti vantaggi:

- I clienti che CodePipeline gestivano l'orchestrazione ora dispongono di un mezzo leggero per implementare le modifiche alla configurazione delle loro applicazioni senza dover implementare l'intera base di codice.
- I clienti che desiderano AWS AppConfig gestire le implementazioni di configurazione ma sono limitati perché AWS AppConfig non supportano il codice o l'archivio di configurazione correnti, ora dispongono di opzioni aggiuntive. CodePipeline supporta AWS CodeCommit GitHub, e BitBucket (solo per citarne alcuni).

### Note

AWS AppConfig l'integrazione con CodePipeline è supportata solo Regioni AWS dove CodePipeline è [disponibile](#).

## Come funziona l'integrazione

Si inizia con l'impostazione e la configurazione CodePipeline. Ciò include l'aggiunta della configurazione a un archivio CodePipeline di codice supportato. Successivamente, configuri il tuo AWS AppConfig ambiente eseguendo le seguenti attività:

- [Crea uno spazio dei nomi e un profilo di configurazione](#)
- [Scegli una strategia di implementazione predefinita o creane una personalizzata](#)

Dopo aver completato queste attività, crei una pipeline CodePipeline che specifica AWS AppConfig come provider di distribuzione. Puoi quindi apportare una modifica alla configurazione e caricarla nel tuo CodePipeline code store. Il caricamento della nuova configurazione avvia automaticamente una nuova distribuzione in CodePipeline. Una volta completata la distribuzione, puoi verificare le modifiche. Per informazioni sulla creazione di una pipeline che viene specificata AWS AppConfig come provider di distribuzione, vedi [Tutorial: Create a Pipeline That Uses AWS AppConfig as a Deployment Provider](#) nella Guida per l'utente AWS CodePipeline

## Ripristino di una configurazione

Durante una distribuzione, puoi mitigare le situazioni in cui dati di configurazione errati o non corretti causano errori nell'applicazione utilizzando i rollback automatici (se si attiva un allarme durante una distribuzione) o ripristinando i dati di configurazione alla versione precedente (se una distribuzione è stata completata con successo).

Per i rollback automatici, puoi utilizzare una combinazione di [strategie di AWS AppConfig distribuzione](#) e CloudWatch allarmi Amazon. Una volta configurato, se uno o più CloudWatch allarmi entrano in funzione durante una distribuzione, ripristina AWS AppConfig automaticamente i dati di configurazione alla versione precedente, evitando così interruzioni o errori delle applicazioni. ALARM Per iniziare, consulta [Configura le autorizzazioni per il rollback automatico](#).

**Note**

È inoltre possibile ripristinare una configurazione richiamando l'operazione [StopDeployment](#) API mentre una distribuzione è ancora in corso.

Per le distribuzioni che vengono completate correttamente, supporta AWS AppConfig anche il ripristino dei dati di configurazione a una versione precedente utilizzando il `AllowRevert` parametro con l'[StopDeployment](#) operazione API. Per alcuni clienti, il ripristino di una configurazione precedente dopo una corretta implementazione garantisce che i dati rimangano gli stessi di prima della distribuzione. Il ripristino ignora anche i monitor degli allarmi, il che può impedire il proseguimento del rollforward durante un'emergenza dell'applicazione.

**⚠ Important**

Se si chiama `StopDeployment` con il `AllowRevert` parametro abilitato, AWS AppConfig ripristinerà la distribuzione solo se l'implementazione è riuscita nelle ultime 72 ore. Dopo 72 ore, la distribuzione non può più essere ripristinata. È necessario creare una nuova distribuzione.

Ecco una suddivisione delle `StopDeployment` funzionalità in base a diverse situazioni.

1. Se `StopDeployment` viene richiamato su una distribuzione in corso, lo stato di distribuzione risultante sarà `ROLLED_BACK`
2. Se `StopDeployment (withAllowRevert)` viene chiamato su una distribuzione in corso, lo stato di distribuzione risultante sarà `ROLLED_BACK`
3. Se `StopDeployment` viene richiamato su una distribuzione completata, `BadRequestException` verrà generato un.
4. Se `StopDeployment (withAllowRevert)` viene chiamato su una distribuzione completata, lo stato di distribuzione risultante sarà `REVERTED`.
5. Se `StopDeployment (withAllowRevert)` viene richiamato su una distribuzione completata dopo 72 ore, `BadRequestException` verrà generato un.

È possibile utilizzare il AWS CLI per chiamare l'[StopDeployment](#) operazione con il `AllowRevert` parametro. Ecco un esempio di AWS CLI comando che include il `AllowRevert` parametro.

```
aws appconfig stop-deployment \
--application-id 339ohji \
--environment-id 54j1r29 \
--deployment-number 2 \
--allow-revert
```

# Recupero dei flag delle funzionalità e dei dati di configurazione in AWS AppConfig

L'applicazione recupera i flag di funzionalità e i dati di configurazione in formato libero stabilendo una sessione di configurazione utilizzando il servizio Data. AWS AppConfig Si consiglia di utilizzare AWS AppConfig Agent per recuperare i dati di configurazione. L'agente (o l'estensione AWS AppConfig Agent Lambda per gli ambienti di calcolo Lambda) gestisce una serie di chiamate API e token di sessione per tuo conto. A un livello elevato, il processo funziona come segue:

1. AWS AppConfig L'agente viene configurato come host locale e l'agente esegue il sondaggio AWS AppConfig per gli aggiornamenti della configurazione.
2. L'agente richiama le azioni [StartConfigurationSession](#) [GetLatestConfiguration](#) API e memorizza nella cache i dati di configurazione localmente.
3. Per recuperare i dati, l'applicazione effettua una chiamata HTTP al server localhost. AWS AppConfig L'agente supporta diversi casi d'uso, come descritto in. [Come utilizzare AWS AppConfig Agent per recuperare i dati di configurazione](#)

Se preferisci, puoi richiamare manualmente queste azioni API per recuperare una configurazione. Il processo API funziona come segue:

1. L'applicazione stabilisce una sessione di configurazione utilizzando l'azione `StartConfigurationSession` API. Il client della sessione effettua quindi chiamate periodiche per `GetLatestConfiguration` verificare e recuperare i dati più recenti disponibili.
2. Durante la chiamata `StartConfigurationSession`, il codice invia gli identificatori (ID o nome) di un' AWS AppConfig applicazione, di un ambiente e di un profilo di configurazione monitorati dalla sessione.
3. In risposta, AWS AppConfig fornisce un `InitialConfigurationToken` codice da fornire al client della sessione e da utilizzare la prima volta che richiama `GetLatestConfiguration` quella sessione.
4. Durante la chiamata `GetLatestConfiguration`, il codice client invia il `ConfigurationToken` valore più recente a sua disposizione e riceve in risposta:
  - `NextPollConfigurationToken`: il `ConfigurationToken` valore da utilizzare nella chiamata successiva a `GetLatestConfiguration`.

- La configurazione: i dati più recenti destinati alla sessione. Questo campo può essere vuoto se il client dispone già dell'ultima versione della configurazione.

### Note

Il recupero dei dati di configurazione da un file separato Account AWS non è supportato.

## Indice

- [Che cos'è Agent AWS AppConfig ?](#)
- [Come utilizzare AWS AppConfig Agent per recuperare i dati di configurazione](#)
- [AWS AppConfig considerazioni sull'uso del browser e dei dispositivi mobili](#)
- [Recupero dei dati di configurazione senza Agent AWS AppConfig](#)

## Che cos'è Agent AWS AppConfig ?

AWS AppConfig Agent è un processo sviluppato e gestito da Amazon per il recupero dei dati di configurazione da AWS AppConfig. Con l'agente, puoi memorizzare nella cache i dati di configurazione localmente e interrogare in modo asincrono il servizio Data Plane per verificare la presenza di aggiornamenti. AWS AppConfig Questo caching/polling processo garantisce che i dati di configurazione siano sempre disponibili per l'applicazione, riducendo al minimo latenza e costi. L'agente non è l'unico modo per recuperare i dati di configurazione AWS AppConfig, ma è il metodo consigliato. L'agente migliora l'elaborazione e la gestione delle applicazioni nei seguenti modi:

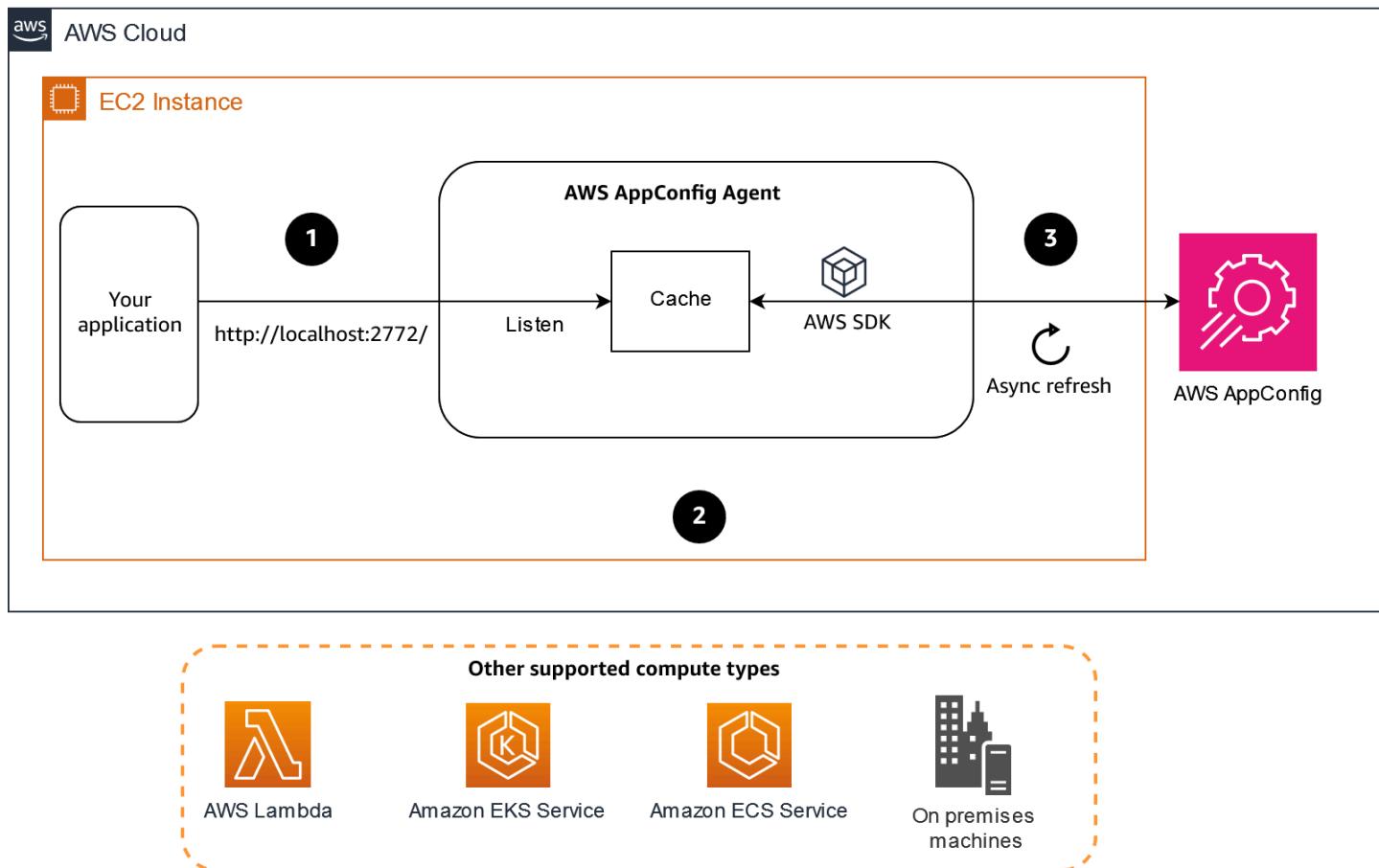
- L'agente chiama AWS AppConfig per conto dell'utente utilizzando un principale AWS Identity and Access Management (IAM) e gestendo una cache locale di dati di configurazione. Recuperando i dati di configurazione dalla cache locale, l'applicazione richiede meno aggiornamenti del codice per gestire i dati di configurazione, recupera i dati di configurazione in millisecondi e non è interessata da problemi di rete che possono interrompere le chiamate per tali dati.
- L'agente offre un'esperienza nativa per il recupero e la risoluzione dei flag di funzionalità. AWS AppConfig
- Immediatamente, l'agente fornisce le migliori pratiche per le strategie di memorizzazione nella cache, gli intervalli di polling e la disponibilità dei dati di configurazione locali, tenendo traccia dei token di configurazione necessari per le successive chiamate di servizio.

- Durante l'esecuzione in background, l'agente esegue periodicamente il polling del servizio AWS AppConfig Data Plane per gli aggiornamenti dei dati di configurazione. L'applicazione può recuperare i dati connettendosi a localhost sulla porta 2772 (un valore di porta predefinito personalizzabile) e chiamando HTTP GET per recuperare i dati.

 Note

AWS AppConfig L'agente memorizza i dati nella cache la prima volta che il servizio recupera i dati di configurazione. Per questo motivo, la prima chiamata per recuperare i dati è più lenta delle chiamate successive.

Il diagramma seguente mostra come funziona AWS AppConfig Agent.



1. L'applicazione richiede i dati di configurazione all'agente.
2. L'agente restituisce i dati da una cache in memoria.

3. L'agente esegue il polling asincrono del AWS AppConfig servizio alla ricerca dei dati di configurazione più recenti con una cadenza predefinita. I dati di configurazione più recenti vengono sempre archiviati in una cache in memoria.

## Come utilizzare AWS AppConfig Agent per recuperare i dati di configurazione

L' AWS AppConfig Agent è il metodo consigliato per recuperare i flag di AWS AppConfig funzionalità o i dati di configurazione in formato libero. L'agente è supportato su tutte le forme di AWS elaborazione, tra cui Amazon EC2, Amazon ECS, Amazon EKS e Lambda. Dopo aver completato la configurazione iniziale dell'agente, utilizzare l'agente per recuperare i dati di configurazione è più semplice rispetto alla chiamata diretta. AWS AppConfig APIs L'agente implementa automaticamente le migliori pratiche e può ridurre i costi di utilizzo grazie AWS AppConfig al minor numero di chiamate API per il recupero delle configurazioni.

### Note

Il recupero dei dati di configurazione da un dispositivo separato non è supportato. Account AWS

## Argomenti

- [Utilizzo di Agent con AWS AppConfigAWS Lambda](#)
- [Utilizzo di AWS AppConfig Agent con Amazon EC2 e macchine locali](#)
- [Utilizzo di AWS AppConfig Agent con Amazon ECS e Amazon EKS](#)
- [Recupero dei flag delle funzionalità di base e multivarianti](#)
- [Utilizzo di un manifesto per abilitare funzionalità di recupero aggiuntive](#)
- [Configurazione AWS AppConfig dell'agente per recuperare le configurazioni da più account](#)
- [Configurazione AWS AppConfig dell'agente per scrivere copie di configurazione su disco](#)
- [Generazione di un client utilizzando la specifica OpenAPI](#)
- [Utilizzo della modalità di sviluppo locale di Agent AWS AppConfig](#)

## Utilizzo di Agent con AWS AppConfigAWS Lambda

Un' AWS Lambda estensione è un processo complementare che aumenta le funzionalità di una funzione Lambda. Un'estensione può iniziare prima che una funzione venga richiamata, essere eseguita in parallelo con una funzione e continuare a funzionare dopo l'elaborazione della chiamata di una funzione. In sostanza, un'estensione Lambda è come un client che viene eseguito in parallelo a una chiamata Lambda. Questo client parallelo può interrarsi con la funzione in qualsiasi momento del suo ciclo di vita.

Se utilizzi i flag di AWS AppConfig funzionalità o altri dati di configurazione dinamici in una funzione Lambda, ti consigliamo di aggiungere l'estensione AWS AppConfig Agent Lambda come livello alla tua funzione Lambda. Ciò semplifica la chiamata ai flag delle funzionalità e l'estensione stessa include le migliori pratiche che ne semplificano l'utilizzo riducendo al contempo i costi. AWS AppConfig I costi ridotti derivano da un minor numero di chiamate API al AWS AppConfig servizio e da tempi di elaborazione delle funzioni Lambda più brevi. Per ulteriori informazioni sulle estensioni Lambda, consulta le estensioni [Lambda nella Developer Guide](#).

### Note

AWS AppConfig è una funzionalità di AWS Systems Manager AWS AppConfig il [prezzo](#) si basa sul numero di volte in cui una configurazione viene chiamata e ricevuta. I costi aumentano se la Lambda esegue più partenze a freddo e recupera frequentemente nuovi dati di configurazione.

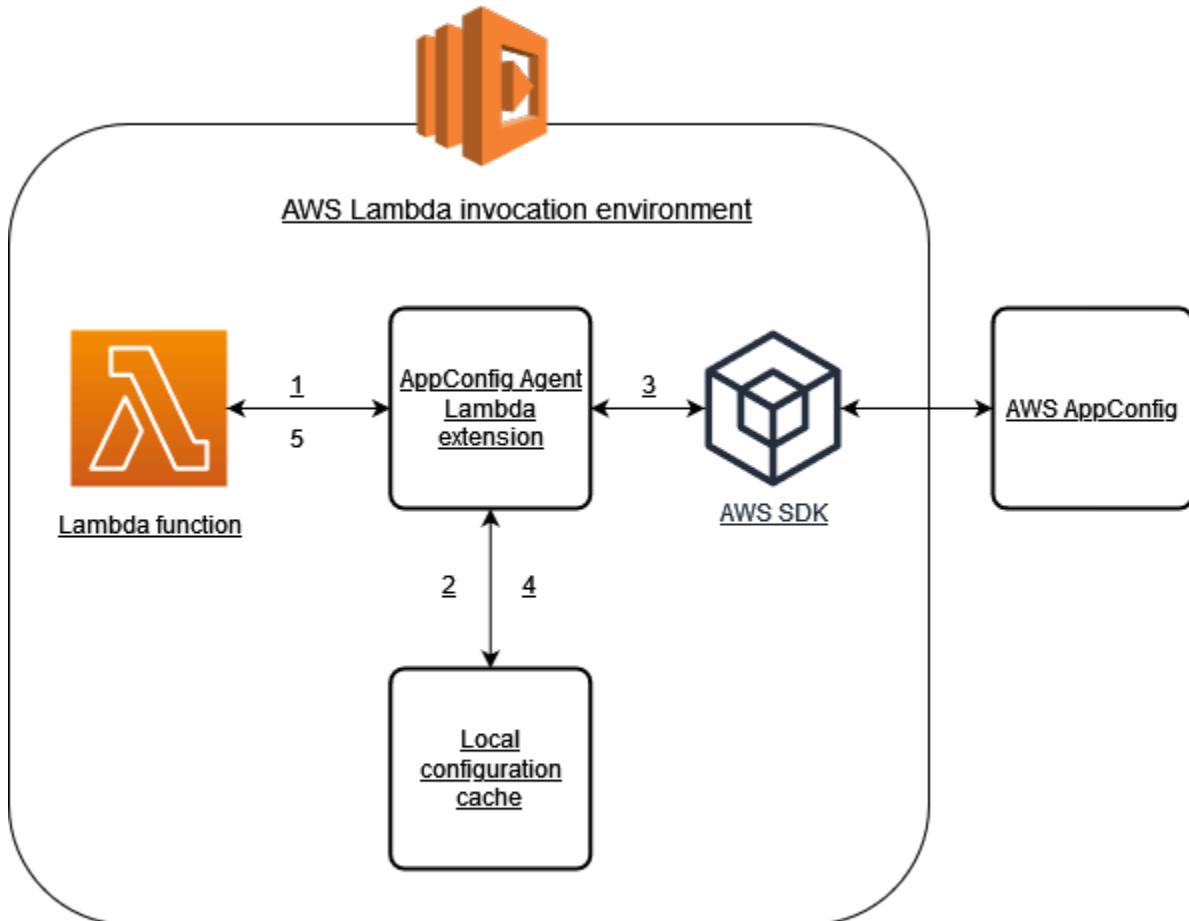
### Argomenti

- [Informazioni sul funzionamento dell' AWS AppConfig estensione Agent Lambda](#)
- [Aggiungere l'estensione AWS AppConfig Agent Lambda](#)
- [Configurazione dell'estensione AWS AppConfig Agent Lambda](#)
- [Informazioni sulle versioni disponibili dell'estensione AWS AppConfig Agent Lambda](#)

### Informazioni sul funzionamento dell' AWS AppConfig estensione Agent Lambda

Se utilizzi AWS AppConfig per gestire le configurazioni per una funzione Lambda senza estensioni Lambda, devi configurare la funzione Lambda per ricevere aggiornamenti di configurazione mediante l'integrazione con le azioni e API. [StartConfigurationSession](#)[GetLatestConfiguration](#)

L'integrazione dell'estensione AWS AppConfig Agent Lambda con la funzione Lambda semplifica questo processo. L'estensione si occupa della chiamata al AWS AppConfig servizio, della gestione di una cache locale dei dati recuperati, del tracciamento dei token di configurazione necessari per le successive chiamate di servizio e del controllo periodico degli aggiornamenti di configurazione in background. Il diagramma seguente mostra come funziona.



1. L'estensione AWS AppConfig Agent Lambda viene configurata come livello della funzione Lambda.
2. Per accedere ai dati di configurazione, la funzione chiama l' AWS AppConfig estensione su un endpoint HTTP in esecuzione. `localhost:2772`
3. L'estensione mantiene una cache locale dei dati di configurazione. Se i dati non sono nella cache, l'estensione chiama AWS AppConfig per ottenere i dati di configurazione.
4. Dopo aver ricevuto la configurazione dal servizio, l'estensione la memorizza nella cache locale e la passa alla funzione Lambda.

5. AWS AppConfig L'estensione Agent Lambda verifica periodicamente la presenza di aggiornamenti dei dati di configurazione in background. Ogni volta che viene richiamata la funzione Lambda, l'estensione controlla il tempo trascorso da quando ha recuperato una configurazione. Se il tempo trascorso è maggiore dell'intervallo di sondaggio configurato, l'estensione chiama AWS AppConfig per verificare la presenza di nuovi dati distribuiti, aggiorna la cache locale in caso di modifiche e reimposta il tempo trascorso.

 Note

- Lambda crea istanze separate corrispondenti al livello di simultaneità richiesto dalla funzione. Ogni istanza è isolata e mantiene la propria cache locale dei dati di configurazione. Per ulteriori informazioni sulle istanze Lambda e sulla concorrenza, vedere [Gestione della concorrenza per una funzione Lambda](#).
- Il tempo necessario affinché una modifica alla configurazione appaia in una funzione Lambda, dopo aver distribuito una configurazione aggiornata da AWS AppConfig, dipende dalla strategia di distribuzione utilizzata per la distribuzione e dall'intervallo di polling configurato per l'estensione.

## Aggiungere l'estensione AWS AppConfig Agent Lambda

Per utilizzare l'estensione AWS AppConfig Agent Lambda, devi aggiungere l'estensione alla tua Lambda. Questo può essere fatto aggiungendo l'estensione AWS AppConfig Agent Lambda alla funzione Lambda come livello o abilitando l'estensione su una funzione Lambda come immagine contenitore.

 Note

L'AWS AppConfig estensione è indipendente dal runtime e supporta tutti i runtime.

## Prima di iniziare

Prima di abilitare l'estensione AWS AppConfig Agent Lambda, procedi come segue:

- Organizza le configurazioni nella tua funzione Lambda in modo da poterle esternalizzare. AWS AppConfig

- Crea AWS AppConfig artefatti e dati di configurazione, inclusi flag di funzionalità o dati di configurazione in formato libero. Per ulteriori informazioni, consulta [Creazione di flag di funzionalità e dati di configurazione in formato libero in AWS AppConfig](#).
- Aggiungi appconfig:StartConfigurationSession e appconfig:GetLatestConfiguration alla policy AWS Identity and Access Management (IAM) utilizzata dal ruolo di esecuzione della funzione Lambda. Per ulteriori informazioni, consulta [Ruolo di esecuzione di AWS Lambda](#) nella Guida per gli sviluppatori di AWS Lambda . Per ulteriori informazioni sulle AWS AppConfig autorizzazioni, consulta [Azioni, risorse e chiavi di condizione AWS AppConfig](#) nel Service Authorization Reference.

Aggiungere l'estensione AWS AppConfig Agent Lambda utilizzando un layer e un ARN

Per utilizzare l'estensione AWS AppConfig Agent Lambda, aggiungete l'estensione alla funzione Lambda come livello. Per informazioni su come aggiungere un livello alla funzione, consulta [Configuring extensions](#) nella Developer Guide.AWS Lambda Il nome dell'estensione nella AWS Lambda console è AWS- AppConfig -Extension. Tieni inoltre presente che quando aggiungi l'estensione come layer a Lambda, devi specificare un Amazon Resource Name (ARN). Scegli un ARN da uno dei seguenti elenchi che corrisponde alla piattaforma e Regione AWS dove hai creato la Lambda.

- [piattaforma x86-64](#)
- [ARM64piattaforma](#)

Se desideri testare l'estensione prima di aggiungerla alla tua funzione, puoi verificarne il funzionamento utilizzando il seguente esempio di codice.

```
import urllib.request

def lambda_handler(event, context):
    url = f'http://localhost:2772/applications/{application_name}/
environments/{environment_name}/configurations/{configuration_name}'
    config = urllib.request.urlopen(url).read()
    return config
```

Per testarlo, crea una nuova funzione Lambda per Python, aggiungi l'estensione e quindi esegui la funzione Lambda. Dopo aver eseguito la funzione Lambda, la funzione AWS AppConfig Lambda restituisce la configurazione specificata per il percorso http://localhost:2772. Per informazioni sulla

creazione di una funzione Lambda, consulta [Create a Lambda function with the console](#) nella Developer Guide AWS Lambda

### Important

È possibile visualizzare i dati di registro per l'estensione AWS AppConfig Agent Lambda nei AWS Lambda registri. Le voci di registro sono precedute da `appconfig agent`. Ecco un esempio:

```
[appconfig agent] 2024/05/07 04:19:01 ERROR retrieve failure for
'SourceEventConfig:SourceEventConfigEnvironment:SourceEventConfigProfile':
StartConfigurationSession: api error AccessDenied: User:
arn:aws:sts::0123456789:assumed-role/us-east-1-LambdaRole/
extension1 is not authorized to perform: sts:AssumeRole on resource:
arn:aws:iam::0123456789:role/test1 (retry in 60s)
```

## Configurazione dell'estensione AWS AppConfig Agent Lambda

È possibile configurare l'estensione modificando le seguenti variabili di AWS Lambda ambiente. Per ulteriori informazioni, consulta [Uso delle variabili di AWS Lambda ambiente](#) nella Guida per gli AWS Lambda sviluppatori.

### Preacquisizione dei dati di configurazione

La variabile di ambiente `AWS_APPCONFIG_EXTENSION_PREFETCH_LIST` può migliorare il tempo di avvio della funzione. Quando l'estensione AWS AppConfig Agent Lambda viene inizializzata, recupera la configurazione specificata da prima che AWS AppConfig Lambda inizi a inizializzare la funzione e a richiamare il gestore. In alcuni casi, i dati di configurazione sono già disponibili nella cache locale prima che la funzione li richieda.

Per utilizzare la funzionalità di prefetch, impostate il valore della variabile di ambiente sul percorso corrispondente ai dati di configurazione. Ad esempio, se la configurazione corrisponde a un'applicazione, un ambiente e un profilo di configurazione denominati rispettivamente «`my_application`», «`my_environment`» e «`my_configuration_data`», il percorso sarebbe `./applications/my_application/environments/my_environment/configurations/my_configuration_data`. Puoi specificare più elementi di configurazione elencandoli come elenco separato da virgolette (se hai un nome di risorsa che include una virgola, usa il valore ID della risorsa anziché il suo nome).

## Accesso ai dati di configurazione da un altro account

L'estensione AWS AppConfig Agent Lambda può recuperare i dati di configurazione da un altro account specificando un ruolo IAM che concede le autorizzazioni ai dati. Per configuralo, segui questi passaggi:

1. Nell'account in cui AWS AppConfig viene utilizzato per gestire i dati di configurazione, crea un ruolo con una policy di fiducia che conceda all'account che esegue la funzione Lambda l'accesso alle `appconfig:StartConfigurationSession` azioni `appconfig:GetLatestConfiguration` e, insieme alle AWS AppConfig risorse di configurazione, parziali o ARNs complete.
2. Nell'account che esegue la funzione Lambda, aggiungi la variabile di `AWS_APPCONFIG_EXTENSION_ROLE_ARN` ambiente alla funzione Lambda con l'ARN del ruolo creato nel passaggio 1.
3. (Facoltativo) Se necessario, è possibile specificare un [ID esterno](#) utilizzando la `AWS_APPCONFIG_EXTENSION_ROLE_EXTERNAL_ID` variabile di ambiente. Allo stesso modo, è possibile configurare un nome di sessione utilizzando la variabile di `AWS_APPCONFIG_EXTENSION_ROLE_SESSION_NAME` ambiente.

### Note

Osservare le seguenti informazioni.

- L'estensione AWS AppConfig Agent Lambda può recuperare i dati da un solo account. Se specifichi un ruolo IAM, l'estensione non sarà in grado di recuperare i dati di configurazione dall'account in cui è in esecuzione la funzione Lambda.
- AWS Lambda registra le informazioni sull'estensione AWS AppConfig Agent Lambda e sulla funzione Lambda utilizzando Amazon Logs, CloudWatch
- La tabella seguente include una colonna di valori di esempio. A seconda della risoluzione del monitor, potrebbe essere necessario scorrere fino alla fine della tabella e quindi scorrere verso destra per visualizzare la colonna.

Variabile di ambiente	Informazioni	Valore predefinito	Valori di esempio
AWS_APPCO NFIG_EXTE NSION_HTT P_PORT	Questa variabile di ambiente specifica la porta su cui viene eseguito il server HTTP locale che ospita l'estensione.	2772	2772
AWS_APPCO NFIG_EXTE NSION_LOG _LEVEL	Questa variabile di ambiente specifica il livello di dettaglio registrato dall'agente. Ogni livello include il livello corrente e tutti i livelli superiori. Il valore non fa distinzione tra maiuscole e minuscole. Dal più dettagliato al meno dettagliato, i livelli di registro sono: <code>tracedebug,,info,warning,error,fatal,none.</code> Il trace registro include informazioni dettagliate, incluse informazioni sulla tempistica, sull'agente.	Info	trace debug Info avviso error fatali nessuno
AWS_APPCO NFIG_EXTE NSION_MAX _CONNECTIONS	Questa variabile di ambiente configura il numero massimo di connessioni da cui l'estensione utilizza per recuperare le	3	3

Variabile di ambiente	Informazioni	Valore predefinito	Valori di esempio
	configurazioni. AWS AppConfig		
AWS_APPCO		45	45
NFIG_EXTE			45 secondi
NSION_POL			
L_INTERVA			5 min
L_SECONDS	Questa variabile di ambiente controlla la frequenza con cui l'agente richiede dati di configurazione aggiornati. AWS AppConfig È possibile specificare un numero di secondi per l'intervallo. È inoltre possibile specificare un numero con un'unità di tempo: s per secondi, m per minuti e h per ore. Se non viene specificata un'unità, l'agente utilizza come impostazione predefinita i secondi. Ad esempio, 60, 60 e 1 m generano lo stesso intervallo di sondaggio.		1 ora

Variabile di ambiente	Informazioni	Valore predefinito	Valori di esempio
AWS_APPCO_NFIG_EXTENSION_POL_L_TIMEOUT_MILLIS	<p>Questa variabile di ambiente controlla il periodo di tempo massimo, in millisecondi, in cui l'estensione attende una risposta AWS AppConfig durante l'aggiornamento dei dati nella cache. Se AWS AppConfig non risponde nel periodo di tempo specificato, l'estensione salta questo intervallo di sondaggio e restituisce i dati memorizzati nella cache precedentemente aggiornati.</p>	3000 ms	3000 ms 300 ms 5 secondi

Variabile di ambiente	Informazioni	Valore predefinito	Valori di esempio
AWS_APPCO NFIG_EXTE NSION_PRE FETCH_LIST	<p>Questa variabile di ambiente specifica i dati di configurazione richiesti dall'agente non AWS AppConfig appena viene avviato. È possibile fornire più identificatori di configurazione in un elenco separato da virgole. La preacquisizione dei dati di configurazione da AWS AppConfig può ridurre in modo significativo il tempo di avvio a freddo della funzione.</p>	Nessuno	<p>MyApp:MyEnv:MyConfig</p> <p>abcd123:efgh456:ijkl789</p> <p>MyApp::Config1, ::Config2</p> <p>MyEnv MyApp MyEnv</p>
AWS_APPCO NFIG_EXTE NSION_PRO XY_HEADERS	<p>Questa variabile di ambiente specifica le intestazioni richieste dal proxy a cui fa riferimento la variabile di ambiente.</p> <p>AWS_APPCO NFIG_EXTE NSION_PRO XY_URL Il valore è un elenco di intestazioni separate da virgole.</p>	Nessuno	<p>intestazione: valore</p> <p>h1: v1, h2: v2</p>

Variabile di ambiente	Informazioni	Valore predefinito	Valori di esempio
AWS_APPCO NFIG_EXTE NSION_PRO XY_URL	Questa variabile di ambiente specifica l'URL del proxy da utilizzare per le connessioni dall'estensione a. AWS AppConfig Servizi AWSHTTPSe HTTP URLs sono supportati.	Nessuno	<a href="http://localhost:7474">http://localhost:7474</a> <a href="https://my-proxy.example.com">https://my-proxy.example.com</a>
AWS_APPCO NFIG_EXTE NSION_ROLE_ARN	Questa variabile di ambiente specifica l'ARN del ruolo IAM corrispondente a un ruolo che deve essere assunto dall'estensione per recuperare AWS AppConfig la configurazione.	Nessuno	<code>arn:aws:iam::123456789012:role/ MyRole</code>
AWS_APPCO NFIG_EXTE NSION_ROL E_EXTERNAL_ID	Questa variabile di ambiente specifica l'id esterno da utilizzare insieme al ruolo ARN assunto.	Nessuno	MyExternalId
AWS_APPCO NFIG_EXTE NSION_ROL E_SESSION_NAME	Questa variabile di ambiente specifica il nome della sessione da associare alle credenziali per il ruolo IAM assunto.	Nessuno	AWSAppConfigAgentSession

Variabile di ambiente	Informazioni	Valore predefinito	Valori di esempio
AWS_APPCO NFIG_EXTE NSION_SER VICE_REGION	<p>Questa variabile di ambiente specifica una regione alternativa che l'estensione deve utilizzare per chiamare il servizio AWS AppConfig. Se non è definita, l'estensione utilizza l'endpoint nella regione corrente.</p>	Nessuno	us-east-1 eu-west-1
AWS_APPCO NFIG_EXTE NSION_MANIFEST	<p>Questa variabile di ambiente configura l'AWS AppConfig agente per sfruttare funzionalità aggiuntive relative alla configurazione, come il recupero di più account e il salvataggio della configurazione su disco. Per ulteriori informazioni su queste caratteristiche, consultare <a href="#">Utilizzo di un manifesto per abilitare funzionalità di recupero aggiuntive</a>.</p>	Nessuno	<p>Quando si utilizza la configurazione come manifesto:..</p> <p>AWS AppConfig MyApp:MyEnv:MyManifestConfig</p> <p>Quando si carica il manifesto dal disco: file:/path/to/manifest.json</p>

Variabile di ambiente	Informazioni	Valore predefinito	Valori di esempio
AWS_APPCO NFIG_EXTE NSION_WAI T_ON_MANIFEST	Questa variabile di ambiente configura l' AWS AppConfig agente in modo che attenda l'elaborazione del manifesto prima di completare l'avvio.	true	true false

## Informazioni sulle versioni disponibili dell'estensione AWS AppConfig Agent Lambda

Questo argomento include informazioni sulle versioni delle estensioni di AWS AppConfig Agent Lambda. L'estensione AWS AppConfig Agent Lambda supporta le funzioni Lambda sviluppate per le piattaforme x86-64 e (Graviton2). ARM64 Per funzionare correttamente, la funzione Lambda deve essere configurata per utilizzare lo specifico Amazon Resource Name (ARN) per il Regione AWS luogo in cui è attualmente ospitata. È possibile visualizzare Regione AWS i dettagli dell'ARN più avanti in questa sezione.

### Important

Nota i seguenti dettagli importanti sull'estensione AWS AppConfig Agent Lambda.

- L'azione `GetConfiguration` API è stata dichiarata obsoleta il 28 gennaio 2022. Le chiamate per ricevere i dati di configurazione devono invece utilizzare il `StartConfigurationSession` comando and `GetLatestConfiguration` APIs Se utilizzi una versione dell'estensione AWS AppConfig Agent Lambda creata dopo il 28 gennaio 2022, devi configurare le autorizzazioni per la nuova. APIs Per ulteriori informazioni, consulta [Recupero dei dati di configurazione senza Agent AWS AppConfig](#).
- AWS AppConfig supporta tutte le versioni elencate in. [Versioni di estensione precedenti](#) Ti consigliamo di eseguire periodicamente l'aggiornamento alla versione più recente per sfruttare i miglioramenti delle estensioni.

## Argomenti

- [AWS AppConfig Note sulla versione di Agent Lambda Extension](#)

- [Individuazione del numero di versione dell'estensione Lambda](#)
- [piattaforma x86-64](#)
- [ARM64piattaforma](#)
- [Versioni di estensione precedenti](#)

### AWS AppConfig Note sulla versione di Agent Lambda Extension

La tabella seguente descrive le modifiche apportate alle versioni recenti dell'estensione AWS AppConfig Lambda.

Versione	Data di lancio	Note
2.0.8693	20/11/2025	<p>Supporto ambientale migliorato, miglioramenti minori e correzioni di bug. È stato aggiunto il supporto per quanto segue: Regioni AWS</p> <ul style="list-style-type: none"><li>• Asia Pacifico (Taipei), ap-east-2</li><li>• Asia Pacifico (Nuova Zelanda), ap-southeast-6</li><li>• Asia Pacifico (Tailandia), ap-southeast-7</li><li>• Messico (Centrale), mx-central-1</li></ul>
2.0.2037	05/12/2025	/pingPercorso aggiunto, che espone un semplice controllo dello stato che restituisce la versione dell'agente. Include anche miglioramenti minori e correzioni di bug.
2.0.1079	12/12/2024	Miglioramenti e correzioni di bug minori.

Versione	Data di lancio	Note
2.0.719	08/08/2024	Miglioramenti e correzioni di bug minori.
2.0.678	23/07/2024	Miglioramenti al supporto degli obiettivi, delle varianti e delle suddivisioni dei feature flag. Per ulteriori informazioni, consulta <a href="#">Creazione di flag di funzionalità multivarianti</a> .
2.0.501	07/01/2024	Miglioramenti e correzioni di bug minori.
2.0.358	12/01/2023	<p><u><a href="#">È stato aggiunto il supporto per le seguenti funzionalità di recupero:</a></u></p> <ul style="list-style-type: none"> <li>• Recupero di più account: utilizza AWS AppConfig Agent da un sistema primario o di recupero Account AWS per recuperare i dati di configurazione da più account fornitore.</li> <li>• Scrittura della copia della configurazione su disco: utilizza AWS AppConfig Agent per scrivere i dati di configurazione su disco. Questa funzionalità consente l'integrazione con i clienti con applicazioni che leggono i dati di configurazione dal disco AWS AppConfig.</li> </ul>

Versione	Data di lancio	Note
2.0.181	14/08/2023	Aggiunto il supporto per Israele (Tel Aviv) Regione AWS il-central-1.
2.0.165	21/02/2023	<p>Correzioni di bug minori. Non si limita più l'uso dell'estensione a specifiche versioni di runtime tramite la console.</p> <p>AWS Lambda È stato aggiunto il supporto per quanto segue: Regioni AWS</p> <ul style="list-style-type: none"> <li>• Medio Oriente (EAU), me-central-1</li> <li>• Asia Pacifico (Hyderabad), ap-south-2</li> <li>• Asia Pacifico (Melbourne), ap-southeast-4</li> <li>• Europa (Spagna), eu-south-2</li> <li>• Europa (Zurigo), eu-central-2</li> </ul>

Versione	Data di lancio	Note
2.0.122	23/08/2022	È stato aggiunto il supporto per un proxy di tunneling, che può essere configurato con le variabili di ambiente. AWS_APPCONFIG_EXTENSION_PROXY_URL AWS_APPCONFIG_EXTENSION_PROXY_HEADER Sono Aggiunto.NET 6 come runtime. Per ulteriori informazioni sulle variabili di ambiente, vedere <a href="#">Configurazione dell'estensione AWS AppConfig Agent Lambda</a> .
2.0.58	05/03/2022	Supporto migliorato per i processori Graviton2 (ARM64) in Lambda.

Versione	Data di lancio	Note
2.0.45	15/03/2022	Aggiunto il supporto per la chiamata di un singolo flag di funzionalità. In precedenza, i clienti chiamavano i flag delle funzionalità raggruppati in un profilo di configurazione e dovevano analizzare la risposta lato client. Con questa versione, i clienti possono utilizzare un <code>flag=&lt;flag-name&gt;</code> parametro quando chiamano l'endpoint HTTP <code>localhost</code> per ottenere il valore di un singolo flag. È stato inoltre aggiunto il supporto iniziale per i processori Graviton2 () . ARM64

## Individuazione del numero di versione dell'estensione Lambda

Utilizzare la procedura seguente per individuare il numero di versione dell'estensione AWS AppConfig Agent Lambda attualmente configurata. Per funzionare correttamente, la funzione Lambda deve essere configurata per utilizzare lo specifico Amazon Resource Name (ARN) per il Regione AWS luogo in cui è attualmente ospitata.

1. Accedi a Console di gestione AWS e apri la AWS Lambda console all'indirizzo. <https://console.aws.amazon.com/lambda/>
2. Scegliete la funzione Lambda in cui desiderate aggiungere il AWS- AppConfig- Extension layer.
3. Nella sezione Livelli, scegli Aggiungi un livello.
4. Nella sezione Scegli un livello, scegli AWS- AppConfig -Estensione dall'elenco dei AWS livelli.
5. Utilizzate l'elenco delle versioni per scegliere un numero di versione.
6. Scegliere Aggiungi.

7. Utilizzate la scheda Test per testare la funzione.
8. Al termine del test, visualizza l'output del registro. Individua la versione dell'estensione AWS AppConfig Agent Lambda nella sezione Dettagli dell'esecuzione. Questa versione deve corrispondere a quella richiesta URLs per quella versione.

piattaforma x86-64

Quando aggiungi l'estensione come layer alla tua Lambda, devi specificare un ARN. Scegli un ARN dalla tabella seguente che corrisponda al Regione AWS luogo in cui hai creato la Lambda. Si ARNs tratta di funzioni Lambda sviluppate per la piattaforma x86-64.

Versione 2.0.8693

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	<code>arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:279</code>
Stati Uniti orientali (Ohio)	<code>arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:235</code>
Stati Uniti occidentali (California settentrionale)	<code>arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:348</code>
Stati Uniti occidentali (Oregon)	<code>arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:335</code>
Canada (Centrale)	<code>arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:228</code>
Canada occidentale (Calgary)	<code>arn:aws:lambda:ca-west-1:436199621743:layer:AWS-AppConfig-Extension:130</code>

Region	ARN
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:261
Europa (Zurigo)	arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension:178
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:261
Europe (Londra)	arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:207
Europe (Paris)	arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:235
Europe (Stockholm)	arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:333
Europa (Milano)	arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:215
Europa (Spagna)	arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension:176
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:205

Region	ARN
Cina (Ningxia)	<code>arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:203</code>
Asia Pacifico (Hong Kong)	<code>arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:217</code>
Asia Pacifico (Tokyo)	<code>arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:228</code>
Asia Pacifico (Seoul)	<code>arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:239</code>
Asia Pacifico (Osaka)	<code>arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:234</code>
Asia Pacifico (Singapore)	<code>arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:224</code>
Asia Pacifico (Sydney)	<code>arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:272</code>
Asia Pacifico (Giacarta)	<code>arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:222</code>
Asia Pacifico (Melbourne)	<code>arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension:152</code>

Region	ARN
Asia Pacifico (Malesia)	arn:aws:lambda:ap-southeast-5:631746059939:layer:AWS-AppConfig-Extension:127
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:248
Asia Pacifico (Hyderabad)	arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension:179
Asia Pacifico (Nuova Zelanda)	arn:aws:lambda:ap-southeast-6:381491832265:layer:AWS-AppConfig-Extension:41
Asia Pacifico (Thailandia)	arn:aws:lambda:ap-southeast-7:851725616657:layer:AWS-AppConfig-Extension:98
Asia Pacifico (Taipei)	arn:aws:lambda:ap-east-2:730335625313:layer:AWS-AppConfig-Extension:100
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:00010852771:layer:AWS-AppConfig-Extension:288
Messico (centrale)	arn:aws:lambda:mx-central-1:891376990304:layer:AWS-AppConfig-Extension:98
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:225

Region	ARN
Israele (Tel Aviv)	<code>arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension:155</code>
Medio Oriente (Emirati Arabi Uniti)	<code>arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension:195</code>
Medio Oriente (Bahrein)	<code>arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:227</code>
AWS GovCloud (Stati Uniti orientali)	<code>arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:184</code>
AWS GovCloud (Stati Uniti occidentali)	<code>arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:182</code>

## ARM64 piattaforma

Quando aggiungi l'estensione come layer alla tua Lambda, devi specificare un ARN. Scegli un ARN dalla tabella seguente che corrisponda al Regione AWS luogo in cui hai creato la Lambda. Si ARNs tratta di funzioni Lambda sviluppate per la ARM64 piattaforma.

## Versione 2.0.8693

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	<code>arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:212</code>

Region	ARN
Stati Uniti orientali (Ohio)	<code>arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:187</code>
Stati Uniti occidentali (California settentrionale)	<code>arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension-Arm64:225</code>
Stati Uniti occidentali (Oregon)	<code>arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:237</code>
Canada (Centrale)	<code>arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension-Arm64:148</code>
Canada occidentale (Calgary)	<code>arn:aws:lambda:ca-west-1:436199621743:layer:AWS-AppConfig-Extension-Arm64:120</code>
Europa (Francoforte)	<code>arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:204</code>
Europa (Zurigo)	<code>arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension-Arm64:136</code>
Europa (Irlanda)	<code>arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:199</code>
Europe (Londra)	<code>arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:159</code>

Region	ARN
Europe (Paris)	arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension-Arm64:154
Europe (Stockholm)	arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension-Arm64:192
Europa (Milano)	arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension-Arm64:143
Europa (Spagna)	arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension-Arm64:137
Asia Pacifico (Hong Kong)	arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension-Arm64:145
Asia Pacifico (Taipei)	arn:aws:lambda:ap-east-2:730335625313:layer:AWS-AppConfig-Extension-Arm64:74
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:181
Asia Pacifico (Seoul)	arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension-Arm64:147
Asia Pacifico (Osaka)	arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension-Arm64:149

Region	ARN
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:176
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:215
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension-Arm64:159
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension-Arm64:137
Asia Pacifico (Malesia)	arn:aws:lambda:ap-southeast-5:631746059939:layer:AWS-AppConfig-Extension-Arm64:102
Asia Pacifico (Nuova Zelanda)	arn:aws:lambda:ap-southeast-6:381491832265:layer:AWS-AppConfig-Extension-Arm64:31
Asia Pacifico (Thailandia)	arn:aws:lambda:ap-southeast-7:851725616657:layer:AWS-AppConfig-Extension-Arm64:97
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension-Arm64:190
Asia Pacifico (Hyderabad)	arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension-Arm64:137

Region	ARN
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:00010852771:layer:AWS-AppConfig-Extension-Arm64:176
Messico (centrale)	arn:aws:lambda:mx-central-1:891376990304:layer:AWS-AppConfig-Extension-Arm64:97
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension-Arm64:153
Medio Oriente (Emirati Arabi Uniti)	arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension-Arm64:151
Medio Oriente (Bahrein)	arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension-Arm64:155
Israele (Tel Aviv)	arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension-Arm64:138
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension-Arm64:127
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension-Arm64:125
AWS GovCloud (Stati Uniti orientali)	arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension-Arm64:130

Region	ARN
AWS GovCloud (Stati Uniti occidentali)	arn:aws:us-gov:lambda:us-gov-west-1:946746059096:layer:AWS- AppConfig-Extension-Arm64:128

## Versioni di estensione precedenti

Questa sezione elenca gli ARNs e Regioni AWS per le versioni precedenti dell'estensione AWS AppConfig Lambda. Questo elenco non contiene informazioni per tutte le versioni precedenti dell'estensione AWS AppConfig Agent Lambda, ma verrà aggiornato quando verranno rilasciate nuove versioni.

### Argomenti

- [Versioni di estensione precedenti \(piattaforma x86-64\)](#)
- [Versioni di estensione precedenti \(ARM64 piattaforma\)](#)

## Versioni di estensione precedenti (piattaforma x86-64)

Le tabelle seguenti ARNs elencano le versioni precedenti dell'estensione AWS AppConfig Agent Lambda sviluppata per la piattaforma x86-64. Regioni AWS

Data sostituita dall'estensione più recente: 20/11/2025

Versione 2.0.2037

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS- AppConfig- Extension:207
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:728743619870:layer:AWS- AppConfig- Extension:162

Region	ARN
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:258
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:262
Canada (Centrale)	arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:152
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:436199621743:layer:AWS-AppConfig-Extension:57
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:189
Europa (Zurigo)	arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension:106
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:189
Europe (Londra)	arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:133
Europe (Paris)	arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:162

Region	ARN
Europe (Stockholm)	<code>arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:259</code>
Europa (Milano)	<code>arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:140</code>
Europa (Spagna)	<code>arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension:102</code>
Cina (Pechino)	<code>arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:133</code>
Cina (Ningxia)	<code>arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:131</code>
Asia Pacifico (Hong Kong)	<code>arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:142</code>
Asia Pacifico (Tokyo)	<code>arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:155</code>
Asia Pacifico (Seoul)	<code>arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:165</code>
Asia Pacifico (Osaka)	<code>arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:159</code>

Region	ARN
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:156
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:199
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:150
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension:78
Asia Pacifico (Malesia)	arn:aws:lambda:ap-southeast-5:631746059939:layer:AWS-AppConfig-Extension:55
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:55480029851:layer:AWS-AppConfig-Extension:175
Asia Pacifico (Hyderabad)	arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension:104
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension:215
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:152

Region	ARN
Israele (Tel Aviv)	<code>arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension:81</code>
Medio Oriente (Emirati Arabi Uniti)	<code>arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension:120</code>
Medio Oriente (Bahrein)	<code>arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:154</code>
AWS GovCloud (Stati Uniti orientali)	<code>arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:110</code>
AWS GovCloud (Stati Uniti occidentali)	<code>arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:109</code>

Data sostituita dalla nuova estensione: 20/05/2025

Versione 2.0.1079

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	<code>arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:174</code>
Stati Uniti orientali (Ohio)	<code>arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:133</code>

Region	ARN
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:223
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:230
Canada (Centrale)	arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:123
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:436199621743:layer:AWS-AppConfig-Extension:27
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:159
Europa (Zurigo)	arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension:77
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:160
Europe (Londra)	arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:121
Europe (Paris)	arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:133

Region	ARN
Europe (Stockholm)	<code>arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:225</code>
Europa (Milano)	<code>arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:111</code>
Europa (Spagna)	<code>arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension:74</code>
Cina (Pechino)	<code>arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:106</code>
Cina (Ningxia)	<code>arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:104</code>
Asia Pacifico (Hong Kong)	<code>arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:113</code>
Asia Pacifico (Tokyo)	<code>arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:126</code>
Asia Pacifico (Seoul)	<code>arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:136</code>
Asia Pacifico (Osaka)	<code>arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:130</code>

Region	ARN
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:134
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:165
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:121
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension:49
Asia Pacifico (Malesia)	arn:aws:lambda:ap-southeast-5:631746059939:layer:AWS-AppConfig-Extension:26
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:55480029851:layer:AWS-AppConfig-Extension:146
Asia Pacifico (Hyderabad)	arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension:75
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension:179
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:123

Region	ARN
Israele (Tel Aviv)	<code>arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension:52</code>
Medio Oriente (Emirati Arabi Uniti)	<code>arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension:91</code>
Medio Oriente (Bahrein)	<code>arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:125</code>
AWS GovCloud (Stati Uniti orientali)	<code>arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:80</code>
AWS GovCloud (Stati Uniti occidentali)	<code>arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:80</code>

Data sostituita dalla nuova estensione: 12/12/2024

Versione 2.0.719

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	<code>arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:173</code>
Stati Uniti orientali (Ohio)	<code>arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:132</code>

Region	ARN
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:221
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:229
Canada (Centrale)	arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:121
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:436199621743:layer:AWS-AppConfig-Extension:27
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:158
Europa (Zurigo)	arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension:75
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:159
Europe (Londra)	arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:120
Europe (Paris)	arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:132

Region	ARN
Europe (Stockholm)	<code>arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:224</code>
Europa (Milano)	<code>arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:110</code>
Europa (Spagna)	<code>arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension:72</code>
Cina (Pechino)	<code>arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:104</code>
Cina (Ningxia)	<code>arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:102</code>
Asia Pacifico (Hong Kong)	<code>arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:112</code>
Asia Pacifico (Tokyo)	<code>arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:125</code>
Asia Pacifico (Seoul)	<code>arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:135</code>
Asia Pacifico (Osaka)	<code>arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:129</code>

Region	ARN
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:132
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:164
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:120
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension:48
Asia Pacifico (Malesia)	arn:aws:lambda:ap-southeast-5:631746059939:layer:AWS-AppConfig-Extension:25
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:55480029851:layer:AWS-AppConfig-Extension:145
Asia Pacifico (Hyderabad)	arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension:74
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension:178
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:122

Region	ARN
Israele (Tel Aviv)	<code>arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension:50</code>
Medio Oriente (Emirati Arabi Uniti)	<code>arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension:90</code>
Medio Oriente (Bahrein)	<code>arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:124</code>
AWS GovCloud (Stati Uniti orientali)	<code>arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:79</code>
AWS GovCloud (Stati Uniti occidentali)	<code>arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:79</code>

Data sostituita dalla nuova estensione: 08/08/2024

Versione 2.0.678

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	<code>arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:167</code>
Stati Uniti orientali (Ohio)	<code>arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:126</code>

Region	ARN
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:213
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:223
Canada (Centrale)	arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:116
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:436199621743:layer:AWS-AppConfig-Extension:21
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:152
Europa (Zurigo)	arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension:70
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:153
Europe (Londra)	arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:114
Europe (Paris)	arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:126

Region	ARN
Europe (Stockholm)	<code>arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:218</code>
Europa (Milano)	<code>arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:104</code>
Europa (Spagna)	<code>arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension:67</code>
Cina (Pechino)	<code>arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:99</code>
Cina (Ningxia)	<code>arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:97</code>
Asia Pacifico (Hong Kong)	<code>arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:106</code>
Asia Pacifico (Tokyo)	<code>arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:119</code>
Asia Pacifico (Seoul)	<code>arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:129</code>
Asia Pacifico (Osaka)	<code>arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:123</code>

Region	ARN
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:127
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:158
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:114
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension:42
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:139
Asia Pacifico (Hyderabad)	arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension:68
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:00010852771:layer:AWS-AppConfig-Extension:172
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:116
Israele (Tel Aviv)	arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension:45

Region	ARN
Medio Oriente (Emirati Arabi Uniti)	<code>arn:aws:lambda:me-central-1:662846165436:layer:AWS- AppConfig-Extension:84</code>
Medio Oriente (Bahrein)	<code>arn:aws:lambda:me-south-1:559955524753:layer:AWS- AppConfig-Extension:118</code>
AWS GovCloud (Stati Uniti orientali)	<code>arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS- AppConfig-Extension:73</code>
AWS GovCloud (Stati Uniti occidentali)	<code>arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS- AppConfig-Extension:73</code>

Data sostituita dalla nuova estensione: 23/07/2024

Versione 2.0.501

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	<code>arn:aws:lambda:us-east-1:027255383542:layer:AWS- AppConfig-Extension:153</code>
Stati Uniti orientali (Ohio)	<code>arn:aws:lambda:us-east-2:728743619870:layer:AWS- AppConfig-Extension:112</code>
Stati Uniti occidentali (California settentrionale)	<code>arn:aws:lambda:us-west-1:958113053741:layer:AWS- AppConfig-Extension:195</code>

Region	ARN
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:210
Canada (Centrale)	arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:101
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:136
Europa (Zurigo)	arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension:53
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:144
Europe (Londra)	arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:99
Europe (Parigi)	arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:111
Europe (Stoccolma)	arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:201
Europa (Milano)	arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:89

Region	ARN
Europa (Spagna)	arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension:50
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:85
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:83
Asia Pacifico (Hong Kong)	arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:91
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:104
Asia Pacifico (Seoul)	arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:114
Asia Pacifico (Osaka)	arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:107
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:112
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:142

Region	ARN
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:98
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension:26
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:125
Asia Pacifico (Hyderabad)	arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension:53
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:00010852771:layer:AWS-AppConfig-Extension:155
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:102
Israele (Tel Aviv)	arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension:28
Medio Oriente (Emirati Arabi Uniti)	arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension:68
Medio Oriente (Bahrein)	arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:103

Region	ARN
AWS GovCloud (Stati Uniti orientali)	arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:59
AWS GovCloud (Stati Uniti occidentali)	arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:59

Data sostituita dalla nuova estensione: 07/01/2024

Versione 2.0.358

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:128
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:93
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:141
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:161
Canada (Centrale)	arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:93

Region	ARN
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:106
Europa (Zurigo)	arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension:47
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:125
Europe (Londra)	arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:93
Europe (Paris)	arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:98
Europe (Stockholm)	arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:159
Europa (Milano)	arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:83
Europa (Spagna)	arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension:44
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:76

Region	ARN
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:76
Asia Pacifico (Hong Kong)	arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:83
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:98
Asia Pacifico (Seoul)	arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:108
Asia Pacifico (Osaka)	arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:101
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:106
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:106
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:79
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension:20

Region	ARN
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:107
Asia Pacifico (Hyderabad)	arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension:47
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:00010852771:layer:AWS-AppConfig-Extension:128
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:83
Israele (Tel Aviv)	arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension:22
Medio Oriente (Emirati Arabi Uniti)	arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension:49
Medio Oriente (Bahrein)	arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:85
AWS GovCloud (Stati Uniti orientali)	arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:54
AWS GovCloud (Stati Uniti occidentali)	arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:54

Data sostituita dalla nuova estensione: 12/01/2023

Versione 2.0.181

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:113
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:81
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:124
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:146
Canada (Centrale)	arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:81
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:93
Europa (Zurigo)	arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension:32
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:110

Region	ARN
Europe (Londra)	arn:aws:lambda:eu-west-2:28260088358:layer:AWS-AppConfig-Extension:81
Europe (Parigi)	arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:82
Europe (Stoccolma)	arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:142
Europa (Milano)	arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:73
Europa (Spagna)	arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension:29
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:68
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:68
Asia Pacifico (Hong Kong)	arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:73
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:84

Region	ARN
Asia Pacifico (Seoul)	arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:93
Asia Pacifico (Osaka)	arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:86
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:91
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:93
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:64
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension:5
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:94
Asia Pacifico (Hyderabad)	arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension:32
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:00010852771:layer:AWS-AppConfig-Extension:113

Region	ARN
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:73
Israele (Tel Aviv)	arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension:7
Medio Oriente (Emirati Arabi Uniti)	arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension:34
Medio Oriente (Bahrein)	arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:73
AWS GovCloud (Stati Uniti orientali)	arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:46
AWS GovCloud (Stati Uniti occidentali)	arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:46

Data sostituita dalla nuova estensione: 14/08/2023

Versione 2.0.165

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:110

Region	ARN
Stati Uniti orientali (Ohio)	<code>arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:79</code>
Stati Uniti occidentali (California settentrionale)	<code>arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:121</code>
Stati Uniti occidentali (Oregon)	<code>arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:143</code>
Canada (Centrale)	<code>arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:79</code>
Europa (Francoforte)	<code>arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:91</code>
Europa (Zurigo)	<code>arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension:29</code>
Europa (Irlanda)	<code>arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:108</code>
Europe (Londra)	<code>arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:79</code>
Europe (Paris)	<code>arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:80</code>

Region	ARN
Europe (Stockholm)	arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:139
Europa (Milano)	arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:71
Europa (Spagna)	arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension:26
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:66
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:66
Asia Pacifico (Hong Kong)	arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:71
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:82
Asia Pacifico (Seoul)	arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:91
Asia Pacifico (Osaka)	arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:84

Region	ARN
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:89
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:91
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:60
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension:2
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:92
Asia Pacifico (Hyderabad)	arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension:29
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:00010852771:layer:AWS-AppConfig-Extension:110
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:71
Medio Oriente (Emirati Arabi Uniti)	arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension:31

Region	ARN
Medio Oriente (Bahrein)	arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:71
AWS GovCloud (Stati Uniti orientali)	arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:44
AWS GovCloud (Stati Uniti occidentali)	arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:44

Data sostituita dalla nuova estensione: 21/02/2023

Versione 2.0.122

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:82
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:59
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:93
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:114

Region	ARN
Canada (Centrale)	arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:59
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:70
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:82
Europe (Londra)	arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:59
Europe (Parigi)	arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:60
Europe (Stoccolma)	arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:111
Europa (Milano)	arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:54
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:52
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:52

Region	ARN
Asia Pacifico (Hong Kong)	arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:54
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:62
Asia Pacifico (Seoul)	arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:70
Asia Pacifico (Osaka)	arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:59
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:64
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:70
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:37
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:71
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:0010852771:layer:AWS-AppConfig-Extension:82

Region	ARN
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:54
Medio Oriente (Bahrein)	arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:54
AWS GovCloud (Stati Uniti orientali)	arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:29
AWS GovCloud (Stati Uniti occidentali)	arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:29

Data sostituita dalla nuova estensione: 23/08/2022

Versione 2.0.58

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:69
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:50
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:78

Region	ARN
Stati Uniti occidentali (Oregon)	<code>arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:101</code>
Canada (Centrale)	<code>arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:50</code>
Europa (Francoforte)	<code>arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:59</code>
Europa (Irlanda)	<code>arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:69</code>
Europe (Londra)	<code>arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:50</code>
Europe (Paris)	<code>arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:51</code>
Europe (Stockholm)	<code>arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:98</code>
Europa (Milano)	<code>arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:47</code>
Cina (Pechino)	<code>arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:46</code>

Region	ARN
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:46
Asia Pacifico (Hong Kong)	arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:47
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:49
Asia Pacifico (Seoul)	arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:59
Asia Pacifico (Osaka)	arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:46
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:51
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:59
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:24
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:60

Region	ARN
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:00010852771:layer:AWS-AppConfig-Extension:69
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:47
Medio Oriente (Bahrein)	arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:47
AWS GovCloud (Stati Uniti orientali)	arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:23
AWS GovCloud (Stati Uniti occidentali)	arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:23

Data sostituita dalla nuova estensione: 21/04/2022

Versione 2.0.45

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:68
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:49

Region	ARN
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:77
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:100
Canada (Centrale)	arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:49
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:58
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:68
Europe (Londra)	arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:49
Europe (Parigi)	arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:50
Europe (Stoccolma)	arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:97
Europa (Milano)	arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:46

Region	ARN
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:45
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:45
Asia Pacifico (Hong Kong)	arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:46
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:48
Asia Pacifico (Seoul)	arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:58
Asia Pacifico (Osaka)	arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:45
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:50
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:58
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:23

Region	ARN
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:59
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension:68
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:46
Medio Oriente (Bahrein)	arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:46
AWS GovCloud (Stati Uniti orientali)	arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:22
AWS GovCloud (Stati Uniti occidentali)	arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:22

Data sostituita dalla nuova estensione: 15/03/2022

Versione 2.0.30

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:61

Region	ARN
Stati Uniti orientali (Ohio)	<code>arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:47</code>
Stati Uniti occidentali (California settentrionale)	<code>arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:61</code>
Stati Uniti occidentali (Oregon)	<code>arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:89</code>
Canada (Centrale)	<code>arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:47</code>
Europa (Francoforte)	<code>arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:54</code>
Europa (Irlanda)	<code>arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:59</code>
Europe (Londra)	<code>arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:47</code>
Europe (Paris)	<code>arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:48</code>
Europe (Stockholm)	<code>arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:86</code>

Region	ARN
Europa (Milano)	arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:44
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:43
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:43
Asia Pacifico (Hong Kong)	arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:44
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:45
Asia Pacifico (Osaka)	arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:42
Asia Pacifico (Seoul)	arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:54
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:45
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:54

Region	ARN
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:13
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:55
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:00010852771:layer:AWS-AppConfig-Extension:61
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:44
Medio Oriente (Bahrein)	arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:44
AWS GovCloud (Stati Uniti orientali)	arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:20
AWS GovCloud (Stati Uniti occidentali)	arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:20

## Versioni di estensione precedenti (ARM64 piattaforma)

Le tabelle seguenti elencano ARNs Regioni AWS le versioni precedenti dell'estensione AWS AppConfig Agent Lambda sviluppata per la ARM64 piattaforma.

Data sostituita dalla nuova estensione: 20/11/2025

## Versione 2.0.2037

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:140
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:114
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension-Arm64:135
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:164
Canada (Centrale)	arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension-Arm64:72
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:436199621743:layer:AWS-AppConfig-Extension-Arm64:47
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:132
Europa (Zurigo)	arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension-Arm64:64
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:127

Region	ARN
Europe (Londra)	<code>arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:85</code>
Europe (Parigi)	<code>arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension-Arm64:81</code>
Europe (Stoccolma)	<code>arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension-Arm64:118</code>
Europa (Milano)	<code>arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension-Arm64:68</code>
Europa (Spagna)	<code>arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension-Arm64:63</code>
Asia Pacifico (Hong Kong)	<code>arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension-Arm64:70</code>
Asia Pacifico (Tokyo)	<code>arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:108</code>
Asia Pacifico (Seoul)	<code>arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension-Arm64:73</code>
Asia Pacifico (Osaka)	<code>arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension-Arm64:74</code>

Region	ARN
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:108
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:142
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension-Arm64:87
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension-Arm64:63
Asia Pacifico (Malesia)	arn:aws:lambda:ap-southeast-5:631746059939:layer:AWS-AppConfig-Extension-Arm64:30
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension-Arm64:117
Asia Pacifico (Hyderabad)	arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension-Arm64:62
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension-Arm64:103
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension-Arm64:80

Region	ARN
Medio Oriente (Emirati Arabi Uniti)	<code>arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension-Arm64:76</code>
Medio Oriente (Bahrein)	<code>arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension-Arm64:82</code>
Israele (Tel Aviv)	<code>arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension-Arm64:64</code>
Cina (Pechino)	<code>arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension-Arm64:55</code>
Cina (Ningxia)	<code>arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension-Arm64:53</code>
AWS GovCloud (Stati Uniti orientali)	<code>arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension-Arm64:56</code>
AWS GovCloud (Stati Uniti occidentali)	<code>arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension-Arm64:55</code>

Data sostituita dalla nuova estensione: 20/05/2025

## Versione 2.0.1079

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:107
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:85
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension-Arm64:100
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:132
Canada (Centrale)	arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension-Arm64:43
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:436199621743:layer:AWS-AppConfig-Extension-Arm64:18
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:102
Europa (Zurigo)	arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension-Arm64:35
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:98

Region	ARN
Europe (Londra)	<code>arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:73</code>
Europe (Parigi)	<code>arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension-Arm64:52</code>
Europe (Stoccolma)	<code>arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension-Arm64:84</code>
Europa (Milano)	<code>arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension-Arm64:39</code>
Europa (Spagna)	<code>arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension-Arm64:35</code>
Asia Pacifico (Hong Kong)	<code>arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension-Arm64:41</code>
Asia Pacifico (Tokyo)	<code>arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:79</code>
Asia Pacifico (Seoul)	<code>arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension-Arm64:44</code>
Asia Pacifico (Osaka)	<code>arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension-Arm64:45</code>

Region	ARN
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:86
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:108
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension-Arm64:58
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension-Arm64:34
Asia Pacifico (Malesia)	arn:aws:lambda:ap-southeast-5:631746059939:layer:AWS-AppConfig-Extension-Arm64:1
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension-Arm64:88
Asia Pacifico (Hyderabad)	arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension-Arm64:33
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension-Arm64:67
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension-Arm64:51

Region	ARN
Medio Oriente (Emirati Arabi Uniti)	<code>arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension-Arm64:47</code>
Medio Oriente (Bahrein)	<code>arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension-Arm64:53</code>
Israele (Tel Aviv)	<code>arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension-Arm64:35</code>
Cina (Pechino)	<code>arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension-Arm64:28</code>
Cina (Ningxia)	<code>arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension-Arm64:26</code>
AWS GovCloud (Stati Uniti orientali)	<code>arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension-Arm64:26</code>
AWS GovCloud (Stati Uniti occidentali)	<code>arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension-Arm64:26</code>

Data sostituita dalla nuova estensione: 12/12/2024

## Versione 2.0.678

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:106
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:84
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension-Arm64:98
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:131
Canada (Centrale)	arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension-Arm64:41
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:436199621743:layer:AWS-AppConfig-Extension-Arm64:17
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:101
Europa (Zurigo)	arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension-Arm64:33
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:97

Region	ARN
Europe (Londra)	<code>arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:72</code>
Europe (Parigi)	<code>arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension-Arm64:51</code>
Europe (Stoccolma)	<code>arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension-Arm64:83</code>
Europa (Milano)	<code>arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension-Arm64:38</code>
Europa (Spagna)	<code>arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension-Arm64:33</code>
Asia Pacifico (Hong Kong)	<code>arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension-Arm64:40</code>
Asia Pacifico (Tokyo)	<code>arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:78</code>
Asia Pacifico (Seoul)	<code>arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension-Arm64:43</code>
Asia Pacifico (Osaka)	<code>arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension-Arm64:44</code>

Region	ARN
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:84
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:107
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension-Arm64:57
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension-Arm64:33
Asia Pacifico (Malesia)	arn:aws:lambda:ap-southeast-5:631746059939:layer:AWS-AppConfig-Extension-Arm64:1
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension-Arm64:87
Asia Pacifico (Hyderabad)	arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension-Arm64:32
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension-Arm64:66
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension-Arm64:50

Region	ARN
Medio Oriente (Emirati Arabi Uniti)	<code>arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension-Arm64:46</code>
Medio Oriente (Bahrein)	<code>arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension-Arm64:52</code>
Israele (Tel Aviv)	<code>arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension-Arm64:33</code>
Cina (Pechino)	<code>arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension-Arm64:26</code>
Cina (Ningxia)	<code>arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension-Arm64:24</code>
AWS GovCloud (Stati Uniti orientali)	<code>arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension-Arm64:25</code>
AWS GovCloud (Stati Uniti occidentali)	<code>arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension-Arm64:25</code>

Data sostituita dalla nuova estensione: 08/08/2024

## Versione 2.0.678

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:100
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:78
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension-Arm64:90
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:125
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:436199621743:layer:AWS-AppConfig-Extension:11
Canada (Centrale)	arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension-Arm64:36
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:95
Europa (Zurigo)	arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension-Arm64:28
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:91

Region	ARN
Europe (Londra)	<code>arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:66</code>
Europe (Paris)	<code>arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension-Arm64:45</code>
Europe (Stockholm)	<code>arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension-Arm64:77</code>
Europa (Milano)	<code>arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension-Arm64:32</code>
Europa (Spagna)	<code>arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension-Arm64:28</code>
Asia Pacifico (Hong Kong)	<code>arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension-Arm64:34</code>
Asia Pacifico (Tokyo)	<code>arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:72</code>
Asia Pacifico (Seoul)	<code>arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension-Arm64:37</code>
Asia Pacifico (Osaka)	<code>arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension-Arm64:38</code>

Region	ARN
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:79
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:101
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension-Arm64:51
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension-Arm64:27
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension-Arm64:81
Asia Pacifico (Hyderabad)	arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension-Arm64:26
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:00010852771:layer:AWS-AppConfig-Extension-Arm64:60
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension-Arm64:44
Medio Oriente (Emirati Arabi Uniti)	arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension-Arm64:40

Region	ARN
Medio Oriente (Bahrein)	arn:aws:lambda:me-south-1:559955524753:layer:AWS- AppConfig-Extension-Arm64:46
Israele (Tel Aviv)	arn:aws:lambda:il-central-1:895787185223:layer:AWS- AppConfig-Extension-Arm64:28
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS- AppConfig-Extension-Arm64:21
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS- AppConfig-Extension-Arm64:19
AWS GovCloud (Stati Uniti orientali)	arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS- AppConfig-Extension-Arm64:19
AWS GovCloud (Stati Uniti occidentali)	arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS- AppConfig-Extension-Arm64:19

Data sostituita dalla nuova estensione: 23/07/2024

Versione 2.0.501

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS- AppConfig-Extension-Arm64:86

Region	ARN
Stati Uniti orientali (Ohio)	<code>arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:64</code>
Stati Uniti occidentali (California settentrionale)	<code>arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension-Arm64:72</code>
Stati Uniti occidentali (Oregon)	<code>arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:112</code>
Canada occidentale (Calgary)	<code>arn:aws:lambda:ca-west-1:436199621743:layer:AWS-AppConfig-Extension:1</code>
Canada (Centrale)	<code>arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension-Arm64:21</code>
Europa (Francoforte)	<code>arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:79</code>
Europa (Zurigo)	<code>arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension-Arm64:11</code>
Europa (Irlanda)	<code>arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:82</code>
Europe (Londra)	<code>arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:51</code>

Region	ARN
Europe (Paris)	arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension-Arm64:30
Europe (Stockholm)	arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension-Arm64:60
Europa (Milano)	arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension-Arm64:17
Europa (Spagna)	arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension-Arm64:11
Asia Pacifico (Hong Kong)	arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension-Arm64:19
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:57
Asia Pacifico (Seoul)	arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension-Arm64:22
Asia Pacifico (Osaka)	arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension-Arm64:22
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:64

Region	ARN
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:85
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension-Arm64:35
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension-Arm64:11
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension-Arm64:67
Asia Pacifico (Hyderabad)	arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension-Arm64:11
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:0010852771:layer:AWS-AppConfig-Extension-Arm64:43
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension-Arm64:30
Medio Oriente (Emirati Arabi Uniti)	arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension-Arm64:24
Medio Oriente (Bahrein)	arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension-Arm64:31

Region	ARN
Israele (Tel Aviv)	arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension-Arm64:11
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension-Arm64:7
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension-Arm64:5
AWS GovCloud (Stati Uniti orientali)	arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension-Arm64:5
AWS GovCloud (Stati Uniti occidentali)	arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension-Arm64:5

Data sostituita dalla nuova estensione: 07/01/2024

Versione 2.0.358

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:61
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:45

Region	ARN
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension-Arm64:18
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:63
Canada (Centrale)	arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension-Arm64:13
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:49
Europa (Zurigo)	arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension-Arm64:5
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:63
Europe (Londra)	arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:45
Europe (Parigi)	arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension-Arm64:17
Europe (Stoccolma)	arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension-Arm64:18

Region	ARN
Europa (Milano)	arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension-Arm64:11
Europa (Spagna)	arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension-Arm64:5
Asia Pacifico (Hong Kong)	arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension-Arm64:11
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:51
Asia Pacifico (Seoul)	arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension-Arm64:16
Asia Pacifico (Osaka)	arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension-Arm64:16
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:58
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:49
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension-Arm64:16

Region	ARN
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension-Arm64:5
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension-Arm64:49
Asia Pacifico (Hyderabad)	arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension-Arm64:5
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:00010852771:layer:AWS-AppConfig-Extension-Arm64:16
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension-Arm64:11
Medio Oriente (Emirati Arabi Uniti)	arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension-Arm64:5
Medio Oriente (Bahrein)	arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension-Arm64:13
Israele (Tel Aviv)	arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension-Arm64:5

Data sostituita dalla nuova estensione: 12/01/2023

## Versione 2.0.181

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:46
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:33
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension-Arm64:1
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:48
Canada (Centrale)	arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension-Arm64:1
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:36
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:48
Europa (Londra)	arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:33
Europa (Parigi)	arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension-Arm64:1

Region	ARN
Europe (Stockholm)	arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension-Arm64:1
Europa (Milano)	arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension-Arm64:1
Asia Pacifico (Hong Kong)	arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension-Arm64:1
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:37
Asia Pacifico (Seoul)	arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension-Arm64:1
Asia Pacifico (Osaka)	arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension-Arm64:1
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:43
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:36
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension-Arm64:1

Region	ARN
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension-Arm64:36
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension-Arm64:1
Africa (Città del Capo)	arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension-Arm64:1
Medio Oriente (Bahrein)	arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension-Arm64:1

Data sostituita dalla nuova estensione: 30/03/2023

Versione 2.0.165

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:43
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:31
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:45

Region	ARN
Europa (Francoforte)	<code>arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:34</code>
Europa (Irlanda)	<code>arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:46</code>
Europe (Londra)	<code>arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:31</code>
Asia Pacifico (Tokyo)	<code>arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:35</code>
Asia Pacifico (Singapore)	<code>arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:41</code>
Asia Pacifico (Sydney)	<code>arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:34</code>
Asia Pacifico (Mumbai)	<code>arn:aws:lambda:ap-south-1:55480029851:layer:AWS-AppConfig-Extension-Arm64:34</code>

Data sostituita dalla nuova estensione: 21/02/2023

## Versione 2.0.122

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:15
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:11
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:16
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:13
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:20
Europe (Londra)	arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:11
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:15
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:16
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:13

Region	ARN
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension-Arm64:13

Data sostituita dalla nuova estensione: 23/08/2022

Versione 2.0.58

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:2
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:2
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:3
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:2
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:7
Europa (Londra)	arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:2

Region	ARN
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:2
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:3
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:2
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension-Arm64:2

Data sostituita dalla nuova estensione: 21/04/2022

Versione 2.0.45

Region	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:1
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:1
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:2

Region	ARN
Europa (Francoforte)	arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:1
Europa (Irlanda)	arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:6
Europe (Londra)	arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:1
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:1
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:2
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:1
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:55480029851:layer:AWS-AppConfig-Extension-Arm64:1

## Utilizzo di AWS AppConfig Agent con Amazon EC2 e macchine locali

Puoi integrarti AWS AppConfig con le applicazioni in esecuzione sulle tue istanze Amazon Elastic Compute Cloud (Amazon EC2) Linux utilizzando AWS AppConfig Agent. L'agente migliora l'elaborazione e la gestione delle applicazioni nei seguenti modi:

- L'agente chiama AWS AppConfig per conto dell'utente utilizzando un ruolo AWS Identity and Access Management (IAM) e gestendo una cache locale dei dati di configurazione. Estraendo i

dati di configurazione dalla cache locale, l'applicazione richiede meno aggiornamenti del codice per gestire i dati di configurazione, recupera i dati di configurazione in millisecondi e non è interessata da problemi di rete che possono interrompere le chiamate per tali dati. \*

- L'agente offre un'esperienza nativa per il recupero e la risoluzione dei AWS AppConfig flag di funzionalità.
- Immediatamente, l'agente fornisce le migliori pratiche per le strategie di memorizzazione nella cache, gli intervalli di polling e la disponibilità dei dati di configurazione locali, tenendo traccia dei token di configurazione necessari per le successive chiamate di servizio.
- Durante l'esecuzione in background, l'agente analizza periodicamente il piano dati per verificare la presenza di aggiornamenti dei AWS AppConfig dati di configurazione. L'applicazione può recuperare i dati connettendosi a localhost sulla porta 2772 (un valore di porta predefinito personalizzabile) e chiamando HTTP GET per recuperare i dati.

\*AWS AppConfig L'agente memorizza i dati nella cache la prima volta che il servizio recupera i dati di configurazione. Per questo motivo, la prima chiamata per recuperare i dati è più lenta delle chiamate successive.

## Argomenti

- [Passaggio 1: \(Obbligatorio\) Creazione di risorse e configurazione delle autorizzazioni](#)
- [Fase 2: \(Obbligatorio\) Installazione e avvio AWS AppConfig dell'agente sulle EC2 istanze Amazon](#)
- [Passaggio 3: \(Facoltativo, ma consigliato\) Invio dei file di registro ai CloudWatch registri](#)
- [Fase 4: \(Facoltativo\) Utilizzo delle variabili di ambiente per configurare AWS AppConfig Agent for Amazon EC2](#)
- [Fase 5: \(Obbligatorio\) Recupero dei dati di configurazione](#)
- [Passaggio 6 \(facoltativo, ma consigliato\): Automatizzazione degli aggiornamenti all'agente AWS AppConfig](#)

## Passaggio 1: (Obbligatorio) Creazione di risorse e configurazione delle autorizzazioni

Per l'integrazione AWS AppConfig con le applicazioni in esecuzione sulle tue EC2 istanze Amazon, devi creare AWS AppConfig artefatti e dati di configurazione, inclusi flag di funzionalità o dati di configurazione in formato libero. Per ulteriori informazioni, consulta [Creazione di flag di funzionalità e dati di configurazione in formato libero in AWS AppConfig](#).

Per recuperare i dati di configurazione ospitati da AWS AppConfig, le applicazioni devono essere configurate con accesso al piano dati. AWS AppConfig Per consentire l'accesso alle tue applicazioni, aggiorna la policy di autorizzazione IAM assegnata al ruolo dell' EC2 istanza Amazon. In particolare, devi aggiungere le `appconfig:GetLatestConfiguration` azioni `appconfig:StartConfigurationSession` e alla policy. Ecco un esempio:

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "appconfig:StartConfigurationSession",  
        "appconfig:GetLatestConfiguration"  
      ],  
      "Resource": "*"  
    }  
  ]  
}
```

Per ulteriori informazioni sull'aggiunta di autorizzazioni a una policy, consulta [Aggiungere e rimuovere le autorizzazioni di identità IAM](#) nella IAM User Guide.

**Fase 2: (Obbligatorio) Installazione e avvio AWS AppConfig dell'agente sulle EC2 istanze Amazon**

AWS AppConfig L'agente è ospitato in un bucket Amazon Simple Storage Service (Amazon S3) gestito da AWS Usa la seguente procedura per installare la versione più recente dell'agente sulla tua istanza Linux. Se l'applicazione è distribuita su più istanze, è necessario eseguire questa procedura su ogni istanza che ospita l'applicazione.



Note

Prendi nota delle seguenti informazioni:

- AWS AppConfig L'agente è disponibile per i sistemi operativi Linux che eseguono la versione del kernel 4.15 o successiva. I sistemi basati su Debian, come Ubuntu, non sono supportati.
- L'agente supporta x86\_64 e architetture ARM64
- Per le applicazioni distribuite, ti consigliamo di aggiungere i comandi di installazione e avvio ai dati EC2 utente Amazon del tuo gruppo Auto Scaling. Se lo fai, ogni istanza esegue i comandi automaticamente. Per ulteriori informazioni, consulta [Esegui comandi sulla tua istanza Linux all'avvio](#) nella Amazon EC2 User Guide. Inoltre, consulta [Tutorial: Configura i dati utente per recuperare lo stato del ciclo di vita di destinazione tramite i metadati dell'istanza nella Amazon Auto Scaling EC2 User Guide](#).
- Le procedure illustrate in questo argomento descrivono come eseguire azioni come l'installazione dell'agente accedendo all'istanza per eseguire il comando. È possibile eseguire i comandi da un computer client locale e indirizzare una o più istanze utilizzando Run Command, uno strumento in AWS Systems Manager Per ulteriori informazioni, consulta [Run Command AWS Systems Manager](#) nella Guida per l'utente AWS Systems Manager .
- AWS AppConfig Agent on Amazon EC2 Linux Instances è un `systemd` servizio.

## Per installare e avviare AWS AppConfig Agent su un'istanza

1. Accedi alla tua istanza Linux.
2. Apri un terminale ed esegui uno dei seguenti comandi con autorizzazioni di amministratore:

x86\_64

```
sudo yum install https://s3.amazonaws.com/aws-appconfig-downloads/aws-appconfig-agent/linux/x86_64/latest/aws-appconfig-agent.rpm
```

ARM64

```
sudo yum install https://s3.amazonaws.com/aws-appconfig-downloads/aws-appconfig-agent/linux/arm64/latest/aws-appconfig-agent.rpm
```

Se desideri installare una versione specifica di AWS AppConfig Agent, sostituiscila `latest` nell'URL con un numero di versione specifico. Ecco un esempio per x86\_64:

```
sudo yum install https://s3.amazonaws.com/aws-appconfig-downloads/aws-appconfig-agent/linux/x86_64/2.0.2/aws-appconfig-agent.rpm
```

3. Esegui il comando seguente per avviare l'agente:

```
sudo systemctl start aws-appconfig-agent
```

4. Esegui il comando seguente per verificare che l'agente sia in esecuzione:

```
sudo systemctl status aws-appconfig-agent
```

In caso di successo, il comando restituisce informazioni come le seguenti:

```
aws-appconfig-agent.service - aws-appconfig-agent
...
Active: active (running) since Mon 2023-07-26 00:00:00 UTC; 0s ago
...
...
```

#### Note

Per arrestare l'agente, esegui questo comando:

```
sudo systemctl stop aws-appconfig-agent
```

## Passaggio 3: (Facoltativo, ma consigliato) Invio dei file di registro ai CloudWatch registri

Per impostazione predefinita, AWS AppConfig Agent pubblica i log su STDERR. Systemd reindirizza STDOUT e STDERR per tutti i servizi in esecuzione sull'istanza Linux al journal systemd. È possibile visualizzare e gestire i dati di registro nel journal systemd se si esegue Agent solo su una o due istanze. AWS AppConfig Una soluzione migliore, una soluzione che consigliamo vivamente per le applicazioni distribuite, è scrivere file di registro su disco e quindi utilizzare Amazon CloudWatch Agent per caricare i dati di registro sul AWS cloud. Inoltre, puoi configurare l' CloudWatch agente per eliminare i vecchi file di registro dall'istanza, in modo da evitare che l'istanza esaurisca lo spazio su disco.

Per abilitare la registrazione su disco, è necessario impostare la variabile di LOG\_PATH ambiente, come descritto in [Fase 4: \(Facoltativo\) Utilizzo delle variabili di ambiente per configurare AWS AppConfig Agent for Amazon EC2](#).

Per iniziare a usare l' CloudWatch agente, consulta [Raccogli metriche e log EC2 dalle istanze Amazon e dai server locali con l'agente CloudWatch nella](#) Amazon User Guide. CloudWatch È possibile utilizzare Quick Setup, uno strumento di Systems Manager per installare rapidamente l' CloudWatch agente. Per ulteriori informazioni, vedere [Quick Setup Host Management](#) nella Guida AWS Systems Manager per l'utente.

### Warning

Se si sceglie di scrivere i file di registro su disco senza utilizzare l' CloudWatch agente, è necessario eliminare i vecchi file di registro. AWS AppConfig L'agente ruota automaticamente i file di registro ogni ora. Se non elimini i vecchi file di registro, l'istanza può esaurire lo spazio su disco.

Dopo aver installato l' CloudWatch agente sull'istanza, crea un file di configurazione CloudWatch dell'agente. Il file di configurazione spiega all' CloudWatch agente come lavorare con i file di registro AWS AppConfig dell'agente. Per ulteriori informazioni sulla creazione di un file di configurazione CloudWatch dell'agente, vedere [Creare il file di configurazione dell' CloudWatch agente](#).

Aggiungi la logs sezione seguente al file di configurazione dell' CloudWatch agente sull'istanza e salva le modifiche:

```
"logs": {  
  "logs_collected": {  
    "files": {  
      "collect_list": [  
        {  
          "file_path": "/path_you_specified_for_logging",  
          "log_group_name": "${YOUR_LOG_GROUP_NAME}/aws-appconfig-agent.log",  
          "auto_removal": true  
        },  
        ...  
      ]  
    },  
    ...  
  },  
  ...  
},
```

```
...  
}
```

Se il valore di `auto_removal` è `true`, l' CloudWatch agente elimina automaticamente i file di registro AWS AppConfig dell'agente ruotati.

## Fase 4: (Facoltativo) Utilizzo delle variabili di ambiente per configurare AWS AppConfig Agent for Amazon EC2

Puoi configurare AWS AppConfig Agent for Amazon EC2 utilizzando variabili di ambiente. Per impostare le variabili di ambiente per un `systemd` servizio, crei un file di unità drop-in. L'esempio seguente mostra come creare un file di unità drop-in su cui impostare il livello di registrazione dell' AWS AppConfig agente. `DEBUG`

Esempio di come creare un file di unità drop-in per le variabili di ambiente

1. Accedi alla tua istanza Linux.
2. Apri un terminale ed esegui il seguente comando con i permessi di amministratore. Il comando crea una directory di configurazione:

```
sudo mkdir /etc/systemd/system/aws-appconfig-agent.service.d
```

3. Esegui il comando seguente per creare il file dell'unità drop-in. Sostituisci `file_name` con un nome per il file. L'estensione deve essere `.conf`:

```
sudo touch /etc/systemd/system/aws-appconfig-agent.service.d/file_name.conf
```

4. Inserisci le informazioni nel file dell'unità drop-in. L'esempio seguente aggiunge una `Service` sezione che definisce una variabile di ambiente. L'esempio imposta il livello di registro AWS AppConfig dell'agente su`DEBUG`.

```
[Service]  
Environment=LOG_LEVEL=DEBUG
```

5. Eseguite il comando seguente per ricaricare la configurazione `systemd`:

```
sudo systemctl daemon-reload
```

6. Esegui il comando seguente per riavviare AWS AppConfig l'agente:

```
sudo systemctl restart aws-appconfig-agent
```

Puoi configurare AWS AppConfig Agent for Amazon EC2 specificando le seguenti variabili di ambiente in un file di unità drop-in.

 Note

La tabella seguente include una colonna di valori di esempio. A seconda della risoluzione del monitor, potrebbe essere necessario scorrere fino alla fine della tabella e quindi scorrere verso destra per visualizzare la colonna.

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
ACCESS_TOKEN	<p>Questa variabile di ambiente definisce un token che deve essere fornito quando si richiedono i dati di configurazione dal server HTTP dell'agente. Il valore del token deve essere impostato nell'interazione di autorizzazione della richiesta HTTP con un tipo di autorizzazione di Bearer. Ecco un esempio.</p> <pre>GET /applications/my_app/... Host: localhost:2772</pre>	Nessuno	MyAccessToken

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
	<pre>Authorization: Bearer &lt;token value&gt;</pre>		

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
BACKUP_DIRECTORY	<p>Questa variabile di ambiente consente all' AWS AppConfig agente di salvare un backup di ogni configurazione recuperata nella directory specificata.</p> <div style="border: 1px solid #f08080; padding: 10px; border-radius: 10px; margin-top: 10px;"><p><b>⚠️ Important</b></p><p>Le configurazioni di cui è stato eseguito il backup su disco non sono crittografate. Se la configurazione contiene dati sensibili, si AWS AppConfig consiglia di applicare il principio del privilegio minimo con le autorizzazioni del file system. Per ulteriori informazioni, consulta <a href="#">Sicurezza</a></p></div>	Nessuno	/path/to/backups

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
	<p><a href="#">in AWS AppConfig.</a></p>		
HTTP_PORT	<p>Questa variabile di ambiente specifica la porta su cui viene eseguito il server HTTP per l'agente.</p>	2772	2772
LOG_LEVEL	<p>Questa variabile di ambiente specifica il livello di dettaglio registrato dall'agente. Ogni livello include il livello corrente e tutti i livelli superiori. Il valore non fa distinzione tra maiuscole e minuscole. Dal più dettagliato al meno dettagliato, i livelli di registro sono: <code>tracedebug,,info,warning,error,fatal,none</code>. Il <code>trace</code> registro include informazioni dettagliate, incluse informazioni sulla tempistica, sull'agente.</p>	Info	<p>trace debug Info avviso error fatali nessuno</p>

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
LOG_PATH	La posizione su disco in cui vengono scritti i log. Se non specificato, i log vengono scritti su stderr.	Nessuno	./log path/to/logs/agent
MANIFEST	Questa variabile di ambiente configura AWS AppConfig Agent per sfruttare funzionalità aggiuntive relative alla configurazione, come il recupero di più account e il salvataggio della configurazione su disco. Per ulteriori informazioni su queste caratteristiche, consultare <a href="#">Utilizzo di un manifesto per abilitare funzionalità di recupero aggiuntive</a> .	Nessuno	Quando si utilizza la configurazione come manifesto: AWS AppConfig MyApp : MyEnv : MyManifestConfig Quando si carica il manifesto dal disco: file:/path/to/manifest.json
MAX_CONNECTIONS	Questa variabile di ambiente configura il numero massimo di connessioni da cui l'agente utilizza per recuperare le configurazioni. AWS AppConfig	3	3

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
POLL_INTERVAL	<p>Questa variabile di ambiente controlla la frequenza con cui l'agente richiede dati di configurazione aggiornati. AWS AppConfig È possibile specificare un numero di secondi per l'intervallo. È inoltre possibile specificare un numero con un'unità di tempo: s per secondi, m per minuti e h per ore. Se non viene specificata un'unità, l'agente utilizza come impostazione predefinita i secondi. Ad esempio, 60, 60 e 1 m generano lo stesso intervallo di sondaggio.</p>	45 secondi	45 45 secondi 5 min 1 ora

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
PREFETCH_LIST	<p>Questa variabile di ambiente specifica i dati di configurazione richiesti dall'agente non AWS AppConfig appena viene avviato. È possibile fornire più identificatori di configurazione in un elenco separato da virgole.</p>	Nessuno	<p>MyApp:MyEnv:MyConfig</p> <p>abcd123:efgh456:ijkl789</p> <p>MyApp::Config1, ::Config2</p> <p>MyEnv MyApp MyEnv</p>

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
PRELOAD_BACKUPS	Se impostato <code>tru</code> e, AWS AppConfig l'agente carica i backup di configurazione trovati <code>BACKUP_DIRECTORY</code> nella memoria e verifica immediatamente se esiste una versione più recente del servizio. Se impostato <code>fa</code> lse, l' AWS AppConfig agente carica i contenuti da un backup di configurazione solo se non è in grado di recuperare i dati di configurazione dal servizio, ad esempio se c'è un problema con la rete.	true	true false
PROXY_HEADERS	Questa variabile di ambiente specifica le intestazioni richieste dal proxy a cui fa riferimento la variabile di ambiente. <code>PROXY_URL</code> Il valore è un elenco di intestazioni separate da virgole.	Nessuno	intestazione: valore h1: v1, h2: v2

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
PROXY_URL	Questa variabile di ambiente specifica l'URL del proxy da utilizzare per le connessioni dall'agente a, incluso. Servizi AWS AWS AppConfig HTTPSe HTTP URLs sono supportati.	Nessuno	<code>http://localhost:7474</code> <code>https://my-proxy.example.com</code>

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
REQUEST_TIMEOUT	<p>Questa variabile di ambiente controlla la quantità di tempo da AWS AppConfig cui l'agente attende una risposta. Se il servizio non risponde, la richiesta ha esito negativo.</p> <p>Se la richiesta riguarda il recupero iniziale dei dati, l'agente restituisce un errore all'applicazione.</p> <p>Se il timeout si verifica durante un controllo in background per verificare la presenza di dati aggiornati, l'agente registra l'errore e riprova dopo un breve ritardo.</p> <p>È possibile specificare il numero di millisecondi per il timeout. È inoltre possibile specificare un numero con un'unità di tempo: ms per millisecondi e s per secondi. Se non viene</p>	<p>3000 ms</p> <p>3000 ms</p> <p>5 secondi</p>	<p>3000</p> <p>3000 ms</p> <p>5 secondi</p>

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
	specificata un'unità, l'agente utilizza come impostazione predefinita i millisecondi. Ad esempio, 5000, 5000 ms e 5 secondi generano lo stesso valore di timeout della richiesta.		
ROLE_ARN	Questa variabile di ambiente specifica l'Amazon Resource Name (ARN) di un ruolo IAM. AWS AppConfig L'agente assume questo ruolo per recuperare i dati di configurazione.	Nessuno	arn:aws:iam:123456789012:role/ MyRole
ROLE_EXTERNAL_ID	Questa variabile di ambiente specifica l'ID esterno da utilizzare con il ruolo ARN assunto.	Nessuno	MyExternalId
ROLE_SESSION_NAME	Questa variabile di ambiente specifica il nome della sessione da associare alle credenziali per il ruolo IAM assunto.	Nessuno	AWSAppConfigAgentSession

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
SERVICE_REGION	<p>Questa variabile di ambiente specifica un'alternativa Regione AWS utilizzata da AWS AppConfig Agent per chiamare il servizio AWS AppConfig. Se non viene definita, l'agente tenta di determinare la regione corrente. In caso contrario, l'agente non si avvia.</p>	Nessuno	us-east-1 eu-west-1
WAIT_ON_MANIFEST	Questa variabile di ambiente configura l' AWS AppConfig agente in modo che attenda l'elaborazione del manifesto prima di completare l'avvio.	true	true false

## Fase 5: (Obbligatorio) Recupero dei dati di configurazione

È possibile recuperare i dati di configurazione dall' AWS AppConfig agente utilizzando una chiamata HTTP localhost. Gli esempi seguenti vengono utilizzati `curl` con un client HTTP. È possibile chiamare l'agente utilizzando qualsiasi client HTTP disponibile supportato dal linguaggio dell'applicazione o dalle librerie disponibili, incluso un AWS SDK.

Per recuperare il contenuto completo di qualsiasi configurazione distribuita

```
$ curl "http://localhost:2772/applications/application_name/environments/environment_name/configurations/configuration_name"
```

Per recuperare un singolo flag e i relativi attributi da una AWS AppConfig configurazione di tipo

## Feature Flag

```
$ curl "http://localhost:2772/applications/application_name/environments/environment_name/configurations/configuration_name?flag=flag_name"
```

Per accedere a più flag e ai relativi attributi da una AWS AppConfig configurazione di tipo **Feature Flag**

```
$ curl "http://localhost:2772/applications/application_name/environments/environment_name/configurations/configuration_name?flag=flag_name_one&flag=flag_name_two"
```

Passaggio 6 (facoltativo, ma consigliato): Automatizzazione degli aggiornamenti all'agente AWS AppConfig

AWS AppConfig L'agente viene aggiornato periodicamente. Per assicurarti di eseguire la versione più recente di AWS AppConfig Agent sulle tue istanze, ti consigliamo di aggiungere i seguenti comandi ai tuoi dati EC2 utente Amazon. È possibile aggiungere i comandi ai dati utente sull'istanza o sul gruppo EC2 Auto Scaling. Lo script installa e avvia la versione più recente dell'agente ogni volta che un'istanza viene avviata o riavviata.

```
#!/bin/bash
# install the latest version of the agent
yum install -y https://s3.amazonaws.com/aws-appconfig-downloads/aws-appconfig-agent/
linux/x86_64/latest/aws-appconfig-agent.rpm
# optional: configure the agent
mkdir /etc/systemd/system/aws-appconfig-agent.service.d
echo "${MY_AGENT_CONFIG}" > /etc/systemd/system/aws-appconfig-agent.service.d/
overrides.conf
systemctl daemon-reload
# start the agent
systemctl start aws-appconfig-agent
```

## Utilizzo di AWS AppConfig Agent con Amazon ECS e Amazon EKS

Puoi effettuare l'integrazione AWS AppConfig con Amazon Elastic Container Service (Amazon ECS) e Amazon Elastic Kubernetes Service (Amazon EKS) utilizzando Agent. AWS AppConfig L'agente funziona come un contenitore secondario che funziona insieme alle applicazioni container Amazon

ECS e Amazon EKS. L'agente migliora l'elaborazione e la gestione delle applicazioni containerizzate nei seguenti modi:

- L'agente chiama AWS AppConfig per conto dell'utente utilizzando un ruolo AWS Identity and Access Management (IAM) e gestendo una cache locale di dati di configurazione. Estraendo i dati di configurazione dalla cache locale, l'applicazione richiede meno aggiornamenti del codice per gestire i dati di configurazione, recupera i dati di configurazione in millisecondi e non è interessata da problemi di rete che possono interrompere le chiamate per tali dati. \*
- L'agente offre un'esperienza nativa per il recupero e la risoluzione dei AWS AppConfig flag di funzionalità.
- Immediatamente, l'agente fornisce le migliori pratiche per le strategie di memorizzazione nella cache, gli intervalli di polling e la disponibilità dei dati di configurazione locali, tenendo traccia dei token di configurazione necessari per le successive chiamate di servizio.
- Durante l'esecuzione in background, l'agente analizza periodicamente il piano dati per verificare la presenza di aggiornamenti dei AWS AppConfig dati di configurazione. L'applicazione containerizzata può recuperare i dati connettendosi a localhost sulla porta 2772 (un valore di porta predefinito personalizzabile) e chiamando HTTP GET per recuperare i dati.
- AWS AppConfig L'agente aggiorna i dati di configurazione nei contenitori senza dover riavviare o riciclare tali contenitori.

\*AWS AppConfig L'agente memorizza i dati nella cache la prima volta che il servizio recupera i dati di configurazione. Per questo motivo, la prima chiamata per recuperare i dati è più lenta delle chiamate successive.

## Prima di iniziare

Per l'integrazione AWS AppConfig con le applicazioni container, è necessario creare AWS AppConfig artefatti e dati di configurazione, inclusi flag di funzionalità o dati di configurazione in formato libero. Per ulteriori informazioni, consulta [Creazione di flag di funzionalità e dati di configurazione in formato libero in AWS AppConfig](#).

Per recuperare i dati di configurazione ospitati da AWS AppConfig, le applicazioni container devono essere configurate con accesso al piano dati. AWS AppConfig Per consentire l'accesso alle applicazioni, aggiorna la policy di autorizzazione IAM utilizzata dal ruolo IAM del servizio container. In particolare, è necessario aggiungere le `appconfig:GetLatestConfiguration` azioni `appconfig:StartConfigurationSession` e alla policy. I ruoli IAM del servizio container includono i seguenti:

- Il ruolo dell'attività di Amazon ECS
- Il ruolo del nodo Amazon EKS
- Il ruolo di esecuzione del AWS Fargate pod (se i contenitori Amazon EKS utilizzano Fargate per l'elaborazione del calcolo)

Per ulteriori informazioni sull'aggiunta di autorizzazioni a una policy, consulta [Aggiungere e rimuovere le autorizzazioni di identità IAM](#) nella Guida per l'utente IAM.

## Argomenti

- [Avvio dell' AWS AppConfig agente per l'integrazione con Amazon ECS](#)
- [Avvio dell' AWS AppConfig agente per l'integrazione con Amazon EKS](#)
- [\(Facoltativo\) Esecuzione AWS AppConfig come DaemonSet in Amazon EKS](#)
- [\(Facoltativo\) Utilizzo di variabili di ambiente per configurare AWS AppConfig Agent for Amazon ECS e Amazon EKS](#)
- [Recupero dei dati di configurazione per le applicazioni in esecuzione in Amazon ECS e Amazon EKS](#)

## Avvio dell' AWS AppConfig agente per l'integrazione con Amazon ECS

Il contenitore sidecar AWS AppConfig Agent è automaticamente disponibile nel tuo ambiente Amazon ECS. Per utilizzarlo, è necessario aviarlo, come descritto nella procedura seguente.

### Per avviare Amazon ECS (console)

1. [Apri la console nella versione 2](#)`https://console.aws.amazon.com/ecs/`.
2. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
3. Scegliete la definizione dell'attività per l'applicazione, quindi selezionate la revisione più recente.
4. Scegli Crea nuova revisione, Crea nuova revisione.
5. Scegli Aggiungi altri contenitori.
6. In Nome, inserisci un nome univoco per il contenitore AWS AppConfig dell'agente.
7. Per l'URI dell'immagine, inserisci: **public.ecr.aws/aws-appconfig/aws-appconfig-agent:2.x**
8. Per Essential container, scegli Sì.

9. Nella sezione Mappature delle porte, scegli Aggiungi mappatura delle porte.
10. Per Container port, inserisci. **2772**

 Note

AWS AppConfig Per impostazione predefinita, l'agente viene eseguito sulla porta 2772. È possibile specificare una porta diversa.

11. Scegli Create (Crea). Amazon ECS crea una nuova revisione del contenitore e ne visualizza i dettagli.
12. Nel pannello di navigazione, scegli Clusters, quindi scegli il tuo cluster di applicazioni nell'elenco.
13. Nella scheda Servizi, seleziona il servizio per la tua applicazione.
14. Scegliere Aggiorna.
15. In Configurazione di distribuzione, per Revisione, scegli la revisione più recente.
16. Scegliere Aggiorna. Amazon ECS implementa la definizione di attività più recente.
17. Al termine della distribuzione, puoi verificare che l' AWS AppConfig agente sia in esecuzione nella scheda Configurazione e attività. Nella scheda Attività, scegli l'attività in esecuzione.
18. Nella sezione Contenitori, verifica che il contenitore AWS AppConfig dell'agente sia elencato.
19. Per verificare che AWS AppConfig l'agente sia stato avviato, scegli la scheda Registri. Individua un'istruzione come la seguente per il contenitore dell' AWS AppConfig agente: [appconfig agent] 1970/01/01 00:00:00 INFO serving on localhost:2772

 Note

Osservare le seguenti informazioni.

- AWS AppConfig L'agente è un processo a esecuzione prolungata. Come best practice per i contenitori Amazon ECS, configura i controlli dello stato dei contenitori, in particolare impostando la dipendenza dal contenitore sulla condizione **HEALTHY**. Per ulteriori informazioni, consulta il riferimento [ContainerDependency](#) all'API di Amazon Elastic Container Service.
- Puoi modificare il comportamento predefinito di AWS AppConfig Agent inserendo o modificando le variabili di ambiente. Per informazioni sulle variabili di ambiente disponibili, vedere [\(Facoltativo\) Utilizzo di variabili di ambiente per configurare AWS AppConfig Agent for Amazon ECS e Amazon EKS](#). Per informazioni su come modificare le variabili di

ambiente in Amazon ECS, consulta [Passing environment variables to a container](#) nella Amazon Elastic Container Service Developer Guide.

## Avvio dell' AWS AppConfig agente per l'integrazione con Amazon EKS

Il contenitore sidecar AWS AppConfig Agent è automaticamente disponibile nel tuo ambiente Amazon EKS. Per utilizzarlo, devi avviarlo. La procedura seguente descrive come utilizzare lo strumento da riga di `kubectl` comando di Amazon EKS per avviare l'agente.

### Note

Prima di continuare, assicurati che il `kubeconfig` file sia aggiornato. Per ulteriori informazioni sulla creazione o la modifica di un `kubeconfig` file, consulta [Creazione o aggiornamento di un file kubeconfig per un cluster Amazon EKS](#) nella Guida per l'utente di Amazon EKS.

Per avviare AWS AppConfig Agent (strumento da riga di comando `kubectl`)

1. Apri il manifesto della tua applicazione e verifica che l'applicazione Amazon EKS sia in esecuzione come distribuzione a contenitore singolo. Il contenuto del file dovrebbe essere simile al seguente.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
  namespace: my-namespace
  labels:
    app: my-application-label
spec:
  replicas: 1
  selector:
    matchLabels:
      app: my-application-label
  template:
    metadata:
      labels:
        app: my-application-label
```

```
spec:  
  containers:  
    - name: my-app  
      image: my-repo/my-image  
      imagePullPolicy: IfNotPresent
```

- Aggiungi i dettagli della definizione del contenitore AWS AppConfig Agent al manifesto di distribuzione.

```
- name: appconfig-agent  
  image: public.ecr.aws/aws-appconfig/aws-appconfig-agent:2.x  
  ports:  
    - name: http  
      containerPort: 2772  
      protocol: TCP  
  env:  
    - name: SERVICE_REGION  
      value: Regione AWS  
  imagePullPolicy: IfNotPresent
```

### Note

Osservare le seguenti informazioni.

- AWS AppConfig Per impostazione predefinita, l'agente viene eseguito sulla porta 2772. È possibile specificare una porta diversa.
- È possibile modificare il comportamento predefinito di AWS AppConfig Agent inserendo le variabili di ambiente. Per ulteriori informazioni, consulta [\(Facoltativo\) Utilizzo di variabili di ambiente per configurare AWS AppConfig Agent for Amazon ECS e Amazon EKS](#).
- Per *Regione AWS*, specificare il Regione AWS codice (ad esempio us-west-1) in cui AWS AppConfig Agent recupera i dati di configurazione.

- Esegui il `kubectl` comando seguente per applicare le modifiche al cluster. Sostituiscilo *my-deployment* con il nome del manifesto di distribuzione.

```
kubectl apply -f my-deployment.yml
```

- Al termine della distribuzione, verifica che AWS AppConfig l'agente sia in esecuzione. Utilizzate il comando seguente per visualizzare il file di registro del pod dell'applicazione.

```
kubectl logs -n my-namespace -c appconfig-agent my-pod
```

Individuate un'istruzione come la seguente per il contenitore AWS AppConfig Agent:

```
[appconfig agent] 1970/01/01 00:00:00 INFO serving on localhost:2772
```

#### Note

È possibile regolare il comportamento predefinito di AWS AppConfig Agent inserendo o modificando le variabili di ambiente. Per informazioni sulle variabili di ambiente disponibili, vedere [\(Facoltativo\) Utilizzo di variabili di ambiente per configurare AWS AppConfig Agent for Amazon ECS e Amazon EKS](#).

### (Facoltativo) Esecuzione AWS AppConfig come DaemonSet in Amazon EKS

Con Amazon EKS, puoi eseguire AWS AppConfig Agent come sidecar, il che si traduce in un contenitore di agenti per pod di applicazioni. Oppure, se preferisci, puoi eseguire AWS AppConfig Agent as a [DaemonSet](#), il che si traduce in un contenitore di agenti per nodo nel cluster.

#### Note

Se esegui AWS AppConfig Agent as a DaemonSet, l'agent viene eseguito in un contenitore separato, il che significa che non puoi accedervi con chiamate a localhost. È necessario inserire o altrimenti scoprire l'indirizzo IP del pod dell'agente per chiamarlo.

Per eseguire AWS AppConfig Agent come file DaemonSet, create un file manifest con i seguenti contenuti. Sostituisci il **highlighted** testo con i dettagli dell'applicazione e dell'ambiente.

Per **Regione AWS**, specifica un Regione AWS codice (ad esempio, `us-west-1`).

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: aws-appconfig-agent
  namespace: my_namespace
  labels:
    app: my_application_label
```

```
spec:
  selector:
    matchLabels:
      app: my_application_label
  template:
    metadata:
      labels:
        app: my_application_label
  spec:
    containers:
      - name: aws-appconfig-agent
        image: public.ecr.aws/aws-appconfig/aws-appconfig-agent:2.x
        ports:
          - name: http
            containerPort: 2772
            protocol: TCP
        env:
          - name: SERVICE_REGION
            value: Regione AWS
        imagePullPolicy: IfNotPresent
      # set a high priority class to ensure the agent is running on every node
      priorityClassName: system-node-critical
```

Esegui il comando seguente per applicare l' AWS AppConfig agente DaemonSet al cluster.

Sostituiscilo *aws\_appconfig\_agent\_daemonset* con il nome del tuo DaemonSet manifesto.

```
kubectl apply -f aws_appconfig_agent_daemonset.yml
```

(Facoltativo) Utilizzo di variabili di ambiente per configurare AWS AppConfig Agent for Amazon ECS e Amazon EKS

È possibile configurare AWS AppConfig l'agente modificando le seguenti variabili di ambiente per il contenitore dell'agente.

 Note

La tabella seguente include una colonna di valori di esempio. A seconda della risoluzione del monitor, potrebbe essere necessario scorrere fino alla fine della tabella e quindi scorrere verso destra per visualizzare la colonna.

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
ACCESS_TOKEN	<p>Questa variabile di ambiente definisce un token che deve essere fornito quando si richiedono i dati di configurazione dal server HTTP dell'agente. Il valore del token deve essere impostato nell'installazione di autorizzazione della richiesta HTTP con un tipo di autorizzazione di <code>Bearer</code>. Ecco un esempio.</p> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <pre>GET /applications/my_app/... Host: localhost:2772 Authorization: Bearer &lt;token value&gt;</pre> </div>	Nessuno	MyAccessToken
BACKUP_DIRECTORY	<p>Questa variabile di ambiente consente all'AWS AppConfig agente di salvare un backup di ogni configurazione recuperata nella directory specificata.</p>	Nessuno	/path/to/backups

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
	<p><b>⚠️ Important</b></p> <p>Le configurazioni di cui è stato eseguito il backup su disco non sono crittografate. Se la configurazione contiene dati sensibili, si AWS AppConfig consiglia di applicare il principio del privilegio minimo con le autorizzazioni del file system. Per ulteriori informazioni, consulta <a href="#">Sicurezza in AWS AppConfig</a>.</p>		
HTTP_PORT	Questa variabile di ambiente specifica la porta su cui viene eseguito il server HTTP per l'agente.	2772	2772

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
LOG_LEVEL	<p>Questa variabile di ambiente specifica il livello di dettaglio registrato dall'agente. Ogni livello include il livello corrente e tutti i livelli superiori. Il valore non fa distinzione tra maiuscole e minuscole. Dal più dettagliato al meno dettagliato, i livelli di registro sono:</p> <p><code>tracedebug,,info,warning,error,fatal,none</code>.</p> <p>Il <code>trace</code> registro include informazioni dettagliate, incluse informazioni sulla tempistica, sull'agente.</p>	Info	<p>trace debug Info avviso error fatali nessuno</p>
LOG_PATH	<p>La posizione su disco in cui vengono scritti i log. Se non specificato, i log vengono scritti su <code>stderr</code>.</p>	Nessuno	<code>./log path/to/logs/agent</code>

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
MANIFEST	<p>Questa variabile di ambiente configura AWS AppConfig Agent per sfruttare funzionalità aggiuntive relative alla configurazione, come il recupero di più account e il salvataggio della configurazione su disco. Per ulteriori informazioni su queste caratteristiche, consultare <a href="#">Utilizzo di un manifesto per abilitare funzionalità di recupero aggiuntive</a>.</p>	Nessuno	<p>Quando si utilizza la configurazione come manifesto: AWS AppConfig MyApp:MyEnv:MyManifestConfig</p> <p>Quando si carica il manifesto dal disco: file:/path/to/manifest.json</p>
MAX_CONNECTIONS	Questa variabile di ambiente configura il numero massimo di connessioni da cui l'agente utilizza per recuperare le configurazioni. AWS AppConfig	3	3

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
POLL_INTERVAL	<p>Questa variabile di ambiente controlla la frequenza con cui l'agente richiede dati di configurazione aggiornati. AWS AppConfig È possibile specificare un numero di secondi per l'intervallo. È inoltre possibile specificare un numero con un'unità di tempo: s per secondi, m per minuti e h per ore. Se non viene specificata un'unità, l'agente utilizza come impostazione predefinita i secondi. Ad esempio, 60, 60 e 1 m generano lo stesso intervallo di sondaggio.</p>	45 secondi	45 45 secondi 5 min 1 ora

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
PREFETCH_LIST	<p>Questa variabile di ambiente specifica i dati di configurazione richiesti dall'agente non AWS AppConfig appena viene avviato. È possibile fornire più identificatori di configurazione in un elenco separato da virgole.</p>	Nessuno	<p>MyApp:MyEnv:MyConfig</p> <p>abcd123:efgh456:ijkl789</p> <p>MyApp::Config1, ::Config2</p> <p>MyEnv MyApp MyEnv</p>

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
PRELOAD_BACKUPS	Se impostato <code>tru</code> e, AWS AppConfig l'agente carica i backup di configurazione trovati <code>BACKUP_DIRECTORY</code> nella memoria e verifica immediatamente se esiste una versione più recente del servizio. Se impostato <code>fa</code> lse, l' AWS AppConfig agente carica i contenuti da un backup di configurazione solo se non è in grado di recuperare i dati di configurazione dal servizio, ad esempio se c'è un problema con la rete.	true	true false
PROXY_HEADERS	Questa variabile di ambiente specifica le intestazioni richieste dal proxy a cui fa riferimento la variabile di ambiente. <code>PROXY_URL</code> Il valore è un elenco di intestazioni separate da virgole.	Nessuno	intestazione: valore h1: v1, h2: v2

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
PROXY_URL	Questa variabile di ambiente specifica l'URL del proxy da utilizzare per le connessioni dall'agente a, incluso. Servizi AWS AWS AppConfig HTTPSe HTTP URLs sono supportati.	Nessuno	<code>http://localhost:7474</code> <code>https://my-proxy.example.com</code>

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
REQUEST_TIMEOUT	<p>Questa variabile di ambiente controlla la quantità di tempo da AWS AppConfig cui l'agente attende una risposta. Se il servizio non risponde, la richiesta ha esito negativo.</p> <p>Se la richiesta riguarda il recupero iniziale dei dati, l'agente restituisce un errore all'applicazione.</p> <p>Se il timeout si verifica durante un controllo in background per verificare la presenza di dati aggiornati, l'agente registra l'errore e riprova dopo un breve ritardo.</p> <p>È possibile specificare il numero di millisecondi per il timeout. È inoltre possibile specificare un numero con un'unità di tempo: ms per millisecondi e s per secondi. Se non viene</p>	3000 ms	3000 3000 ms 5 secondi

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
	specificata un'unità, l'agente utilizza come impostazione predefinita i millisecondi. Ad esempio, 5000, 5000 ms e 5 secondi generano lo stesso valore di timeout della richiesta.		
ROLE_ARN	Questa variabile di ambiente specifica l'Amazon Resource Name (ARN) di un ruolo IAM. AWS AppConfig L'agente assume questo ruolo per recuperare i dati di configurazione.	Nessuno	arn:aws:iam:123456789012:role/ MyRole
ROLE_EXTERNAL_ID	Questa variabile di ambiente specifica l'ID esterno da utilizzare con il ruolo ARN assunto.	Nessuno	MyExternalId
ROLE_SESSION_NAME	Questa variabile di ambiente specifica il nome della sessione da associare alle credenziali per il ruolo IAM assunto.	Nessuno	AWSAppConfigAgentSession

Variabile di ambiente	Informazioni	Valore predefinito	Valore (i) di esempio
SERVICE_REGION	<p>Questa variabile di ambiente specifica un'alternativa Regione AWS utilizzata da AWS AppConfig Agent per chiamare il servizio AWS AppConfig. Se non viene definita, l'agente tenta di determinare la regione corrente. In caso contrario, l'agente non si avvia.</p>	Nessuno	us-east-1 eu-west-1
WAIT_ON_MANIFEST	Questa variabile di ambiente configura l' AWS AppConfig agente in modo che attenda l'elaborazione del manifesto prima di completare l'avvio.	true	true false

## Recupero dei dati di configurazione per le applicazioni in esecuzione in Amazon ECS e Amazon EKS

Puoi recuperare i dati di configurazione da AWS AppConfig Agent per le applicazioni in esecuzione in Amazon ECS e Amazon EKS utilizzando una chiamata HTTP localhost. Gli esempi seguenti vengono utilizzati `curl` con un client HTTP. È possibile chiamare l'agente utilizzando qualsiasi client HTTP disponibile supportato dal linguaggio dell'applicazione o dalle librerie disponibili.

### Note

Per recuperare i dati di configurazione se l'applicazione utilizza una barra, ad esempio «`test-backend/test-service`», è necessario utilizzare la codifica URL.

Per recuperare il contenuto completo di qualsiasi configurazione distribuita

```
$ curl "http://localhost:2772/applications/application_name/environments/environment_name/configurations/configuration_name"
```

Per recuperare un singolo flag e i relativi attributi da una AWS AppConfig configurazione di tipo

### Feature Flag

```
$ curl "http://localhost:2772/applications/application_name/environments/environment_name/configurations/configuration_name?flag=flag_name"
```

Per accedere a più flag e ai relativi attributi da una AWS AppConfig configurazione di tipo **Feature Flag**

```
$ curl "http://localhost:2772/applications/application_name/environments/environment_name/configurations/configuration_name?flag=flag_name_one&flag=flag_name_two"
```

La chiamata restituisce i metadati di configurazione nelle intestazioni HTTP, tra cui la versione di configurazione, il tipo di contenuto e l'etichetta della versione di configurazione (se applicabile). Il corpo della risposta dell'agente contiene il contenuto della configurazione. Ecco un esempio:

```
HTTP/1.1 200 OK
Configuration-Version: 1
Content-Type: application/json
Date: Tue, 18 Feb 2025 20:20:16 GMT
Content-Length: 31
```

```
My test config
```

## Recupero dei flag delle funzionalità di base e multivarianti

Per le configurazioni dei feature flag (configurazioni di tipo `AWS.AppConfig.FeatureFlags`), l'AWS AppConfig Agent consente di recuperare un singolo flag o un sottoinsieme di flag in una configurazione. Il recupero di uno o due flag è utile se il caso d'uso richiede solo l'utilizzo di alcuni flag dal profilo di configurazione. I seguenti esempi utilizzano cURL.

### Note

La possibilità di richiamare un singolo flag di funzionalità o un sottoinsieme di flag in una configurazione è disponibile solo nella versione AWS AppConfig Agent 2.0.45 e successive.

È possibile recuperare i dati di AWS AppConfig configurazione da un endpoint HTTP locale. Per accedere a un flag specifico o a un elenco di flag, utilizzate il parametro di `?flag=FLAG_KEY` query per un AWS AppConfig profilo di configurazione.

Per recuperare un singolo flag e i relativi attributi

```
curl "http://localhost:2772/applications/APPLICATION_NAME/environments/ENVIRONMENT_NAME/configurations/CONFIGURATION_NAME?flag=FLAG_KEY"
```

Per recuperare più bandiere e i relativi attributi

```
curl "http://localhost:2772/applications/APPLICATION_NAME/environments/ENVIRONMENT_NAME/configurations/CONFIGURATION_NAME?flag=FLAG_KEY_ONE&flag=FLAG_KEY_TWO"
```

Per recuperare le varianti dei feature flag in base al contesto del chiamante

I seguenti esempi di cURL mostrano come recuperare le varianti dei feature flag in base al contesto del chiamante. Per illustrare al meglio come effettuare queste chiamate, questa sezione utilizza chiamate di esempio basate su uno scenario in cui un cliente ha creato varianti simili alle seguenti:

Feature flag variants <small>Info</small>			
Name	Enabled value	Attribute values	Rule
beta testers	 ON	-	(or (eq \$userId "Alice") (eq \$userId "123456789012"))
EU demographic	 ON	-	(and (ends_with \$email "@example.com") (eq \$continent "EU"))
QA testing	 ON	-	(and (matches pattern: ".@example.com" in:\$email) (contains \$roles "Engineer") (gt \$tenure 5))
default	 ON	-	-

 **Variant order is used for evaluation logic**  
Variants are evaluated as an ordered list based on the order shown and any specified rules. The variant at the top of the list is evaluated first. If no rules match the supplied context, AWS AppConfig returns the default variant.

## Note

Per recuperare le varianti del flag, è necessario utilizzare la versione più recente di AWS AppConfig Agent nel proprio ambiente di calcolo. Per ulteriori informazioni, consulta i seguenti argomenti che descrivono come aggiornare, installare o aggiungere l'agente per ciascuno dei seguenti ambienti di calcolo:

- Per gli ambienti di calcolo Lambda: [Aggiungere l'estensione AWS AppConfig Agent Lambda](#)
- Per gli ambienti di EC2 elaborazione Amazon: [Fase 2: \(Obbligatorio\) Installazione e avvio AWS AppConfig dell'agente sulle EC2 istanze Amazon](#)
- Per gli ambienti di calcolo Amazon ECS: [Avvio dell' AWS AppConfig agente per l'integrazione con Amazon ECS](#)
- Per gli ambienti di calcolo Amazon EKS: [Avvio dell' AWS AppConfig agente per l'integrazione con Amazon EKS](#)

Per recuperare i dati relativi ai flag utilizzando il contesto del chiamante di `jane_doe@example.org` (che non ha aderito al programma beta):

```
curl http://localhost:2772/applications/UIRefresh/environments/Production/
configurations/Features \
-H "Context: email=jane_doe@example.org" \
-H "Context: opted_in_to_beta=false"
{
  "ui_refresh": {"_variant": "QA", "dark_mode_support": true, "enabled": true}
}
```

Per recuperare i dati relativi ai flag utilizzando il contesto del chiamante di `jane_doe@example.org` (chi ha aderito al programma beta):

```
curl http://localhost:2772/applications/UIRefresh/environments/Production/
configurations/Features \
-H "Context: email=jane_doe@example.org" \
-H "Context: opted_in_to_beta=true"
{
  "ui_refresh": {"_variant": "QA", "dark_mode_support": true, "enabled": true}
}
```

Per recuperare i dati di segnalazione utilizzando il contesto del chiamante di `jane_doe@qa-testers.example.org` (che è un tester di garanzia della qualità presso Example Organization):

```
curl http://localhost:2772/applications/UIRefresh/environments/Production/
configurations/Features \
-H "Context: email=jane_doe@qa-testers.example.org"
{
  "ui_refresh": {"_variant": "QA", "dark_mode_support": true, "enabled": true}
}
```

Per recuperare i dati del flag senza il contesto del chiamante (che restituisce la variante predefinita)

```
curl http://localhost:2772/applications/UIRefresh/environments/Production/
configurations/Features
{
  "ui_refresh": {"_variant": "Default Variant", "enabled": false}
}
```

Recuperare i dati di segnalazione per uno scenario di suddivisione del traffico, per determinare se 1 chiamante casuale su 10 riceve la variante «popolazione campione»

```
for i in {0..9} do ; \
curl http://localhost:2772/applications/UIRefresh/environments/Production/
configurations/Features \
-H "Context: email=$i@example.org"
{
  "ui_refresh": {"_variant": "Default Variant", "enabled": false}
}
{
  "ui_refresh": {"_variant": "Sample
Population", "dark_mode_support": false, "enabled": true}
}
{
```

Utilizzo di un manifesto per abilitare funzionalità di recupero aggiuntive

AWS AppConfig Agent offre le seguenti funzionalità aggiuntive per aiutarvi a recuperare le configurazioni per le vostre applicazioni.

- [Configurazione AWS AppConfig dell'agente per recuperare le configurazioni da più account](#): Utilizzate AWS AppConfig Agent da un sistema primario o di recupero per recuperare i dati di configurazione Account AWS da account di più fornitori.
  - [Configurazione AWS AppConfig dell'agente per scrivere copie di configurazione su disco](#): utilizza AWS AppConfig Agent per scrivere i dati di configurazione su disco. Questa funzionalità consente l'integrazione con i clienti con applicazioni che leggono i dati di configurazione dal disco AWS AppConfig.

## Comprensione dei manifesti degli agenti

Per abilitare queste funzionalità AWS AppConfig dell'agente, è necessario creare un manifesto. Un manifesto è un insieme di dati di configurazione forniti per controllare le azioni che l'agente può eseguire. Un manifesto è scritto in JSON. Contiene un set di chiavi di primo livello che corrispondono a diverse configurazioni che hai distribuito utilizzando AWS AppConfig.

Un manifesto può includere più configurazioni. Inoltre, ogni configurazione nel manifesto può identificare una o più funzionalità dell'agente da utilizzare per la configurazione specificata. Il contenuto del manifesto utilizza il seguente formato:

```
{  
  "application_name:environment_name:configuration_name": {  
    "agent_feature_to_enable_1": {  
      "feature-setting-key": "feature-setting-value"  
    },  
    "agent_feature_to_enable_2": {  
      "feature-setting-key": "feature-setting-value"  
    }  
  }  
}
```

Ecco un esempio di JSON per un manifesto con due configurazioni. La prima configurazione (*MyApp*) non utilizza alcuna funzionalità AWS AppConfig dell'agente. La seconda configurazione (*My2ndApp*) utilizza la copia di scrittura della configurazione su disco e le funzionalità di recupero per più account:

```
{  
  "MyApp:Test:MyAllowListConfiguration": {},  
  
  "My2ndApp:Beta:MyEnableMobilePaymentsFeatureFlagConfiguration": {  
    "credentials": {  
      "roleArn": "arn:aws:us-west-1:iam::123456789012:role/MyTestRole",  
      "roleExternalId": "00b148e2-4ea4-46a1-ab0f-c422b54d0aac",  
      "roleSessionName": "Aws AppConfig Agent",  
      "credentialsDuration": "2h"  
    },  
    "writeTo": {  
      "path": "/tmp/aws-appconfig/my-2nd-app/beta/my-enable-payments-feature-  
flag-configuration.json"  
    }  
  }  
}
```

## Come fornire un manifesto dell'agente

È possibile archiviare il manifesto come file in una posizione in cui AWS AppConfig Agent può leggerlo. In alternativa, è possibile archiviare il manifesto come AWS AppConfig configurazione e

indirizzare l'agente verso di esso. Per fornire un manifesto dell'agente, è necessario impostare una variabile di MANIFEST ambiente con uno dei seguenti valori:

Posizione del manifesto	Valore della variabile d'ambiente	Caso d'uso
File	file:/path/to/agent-manifest.json	Usa questo metodo se il tuo manifesto non cambia spesso.
AWS AppConfig configurazione	<i>application-name:environment-name:configuration-name</i>	Utilizzate questo metodo per gli aggiornamenti dinamici. È possibile aggiornare e distribuire un manifesto archiviato AWS AppConfig come configurazione nello stesso modo in cui si archiviano altre AWS AppConfig configurazioni.
Variabile di ambiente	Contenuto del manifesto (JSON)	Usa questo metodo se il tuo manifesto non cambia spesso. Questo metodo è utile in ambienti container in cui è più facile impostare una variabile di ambiente piuttosto che esporre un file.

Per ulteriori informazioni sull'impostazione delle variabili per AWS AppConfig Agent, consultate l'argomento pertinente al vostro caso d'uso:

- [Configurazione dell'estensione AWS AppConfig Agent Lambda](#)
- [Utilizzo di AWS AppConfig Agent con Amazon EC2](#)
- [Utilizzo di AWS AppConfig Agent con Amazon ECS e Amazon EKS](#)

## Configurazione AWS AppConfig dell'agente per recuperare le configurazioni da più account

È possibile configurare AWS AppConfig l'agente per recuperare le configurazioni da più configurazioni Account AWS inserendo le sostituzioni delle credenziali nel manifesto dell'agente. AWS AppConfig Le sostituzioni delle credenziali includono l'Amazon Resource Name (ARN) di un ruolo AWS Identity and Access Management (IAM), un ID di ruolo, un nome di sessione e la durata per cui l'agente può assumere il ruolo.

Inserisci questi dettagli in una sezione «credenziali» del manifesto. La sezione «credenziali» utilizza il seguente formato:

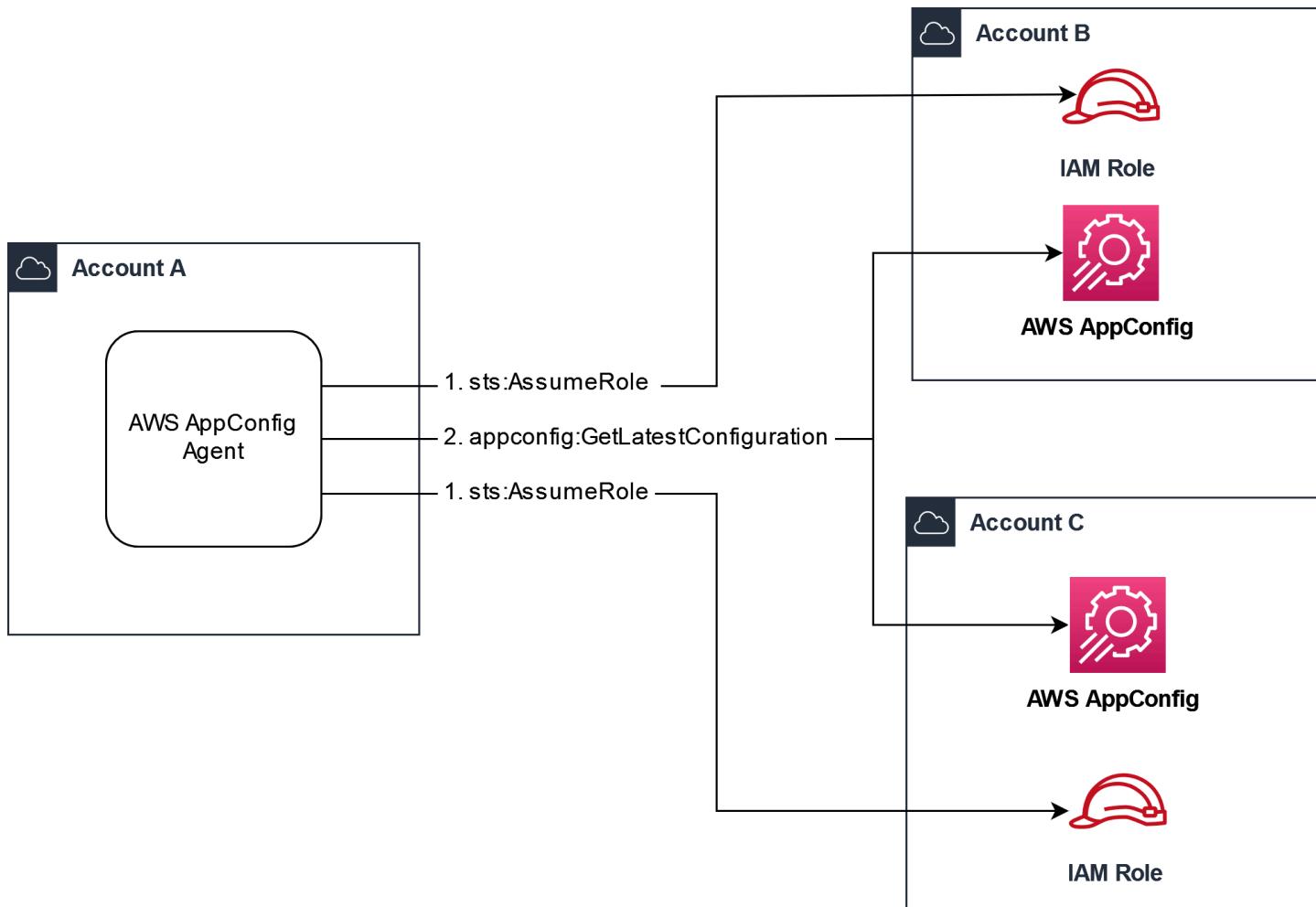
```
{  
  "application_name:environment_name:configuration_name": {  
    "credentials": {  
      "roleArn": "arn:partition:iam::account_ID:role/roleName",  
      "roleExternalId": "string",  
      "roleSessionName": "string",  
      "credentialsDuration": "time_in_hours"  
    }  
  }  
}
```

Ecco un esempio:

```
{  
  "My2ndApp:Beta:MyEnableMobilePaymentsFeatureFlagConfiguration": {  
    "credentials": {  
      "roleArn": "arn:aws:us-west-1:iam::123456789012:role/MyTestRole",  
      "roleExternalId": "00b148e2-4ea4-46a1-ab0f-c422b54d0aac",  
      "roleSessionName": "AWSAppConfigAgent",  
      "credentialsDuration": "2h"  
    }  
  }  
}
```

Prima di recuperare una configurazione, l'agente legge i dettagli delle credenziali per la configurazione dal manifest e quindi assume il ruolo IAM specificato per quella configurazione. È possibile specificare un set diverso di sostituzioni di credenziali per diverse configurazioni in un unico manifesto. Il diagramma seguente mostra come l' AWS AppConfig agente, durante l'esecuzione

nell'account A (l'account di recupero), assuma ruoli separati specificati per gli account B e C (gli account fornitori) e quindi richiami l'operazione [GetLatestConfiguration](#) API per recuperare i dati di configurazione dall'esecuzione in tali account: AWS AppConfig



Configura le autorizzazioni per recuperare i dati di configurazione dagli account dei fornitori

**AWS AppConfig** L'agente in esecuzione nell'account di recupero necessita dell'autorizzazione per recuperare i dati di configurazione dagli account del fornitore. Si concede l'autorizzazione all'agente creando un ruolo AWS Identity and Access Management (IAM) in ciascuno degli account del fornitore. **AWS AppConfig** L'agente nell'account di recupero assume questo ruolo per ottenere dati dagli account dei fornitori. Completa le procedure in questa sezione per creare una policy di autorizzazioni IAM, un ruolo IAM e aggiungere le sostituzioni degli agenti al manifesto.

Prima di iniziare

Raccogli le seguenti informazioni prima di creare una politica di autorizzazione e un ruolo in IAM.

- Il IDs per ciascuno Account AWS. L'account di recupero è l'account che chiamerà altri account per i dati di configurazione. Gli account fornitori sono gli account che invieranno i dati di configurazione all'account di recupero.
  - Il nome del ruolo IAM utilizzato da AWS AppConfig nell'account di recupero. Ecco un elenco dei ruoli utilizzati da AWS AppConfig, per impostazione predefinita:
    - Per Amazon Elastic Compute Cloud (Amazon EC2), AWS AppConfig utilizza il ruolo di istanza.
    - Per AWS Lambda, AWS AppConfig utilizza il ruolo di esecuzione Lambda.
    - Per Amazon Elastic Container Service (Amazon ECS) e Amazon Elastic Kubernetes Service (Amazon EKS), utilizza il ruolo contenitore. AWS AppConfig

Se hai configurato l' AWS AppConfig agente per utilizzare un ruolo IAM diverso specificando la variabile di `ROLE_ARN` ambiente, prendi nota di quel nome.

## Crea la politica delle autorizzazioni

Utilizza la seguente procedura per creare una politica di autorizzazioni utilizzando la console IAM. Completa la procedura in ciascuna unità Account AWS che fornirà i dati di configurazione per l'account di recupero.

## Per creare una policy IAM

1. Accedi all'account di un Console di gestione AWS fornitore.
  2. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
  3. Nel pannello di navigazione, scegli Policy e Crea policy.
  4. Scegli l'opzione JSON.
  5. Nell'editor delle politiche, sostituisci il codice JSON predefinito con la seguente dichiarazione di policy. Aggiorna ciascuno *example resource placeholder* con i dettagli dell'account del fornitore.

JSON

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

```

```
        "appconfig:StartConfigurationSession",
        "appconfig:GetLatestConfiguration"
    ],
    "Resource": "arn:aws:appconfig:us-
east-1:111122223333:application/vendor_application_ID/
environment/vendor_environment_ID/configuration/vendor_configuration_ID"
}
]
}
```

Ecco un esempio:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "appconfig:StartConfigurationSession",
                "appconfig:GetLatestConfiguration"
            ],
            "Resource": "arn:aws:appconfig:us-east-2:111122223333:application/
abc123/environment/def456/configuration/hij789"
        }
    ]
}
```

6. Scegli Next (Successivo).
7. Nel campo Nome della politica, inserisci un nome.
8. (Facoltativo) Per Aggiungi tag, aggiungi una o più coppie tag-chiave-valore per organizzare, tracciare o controllare l'accesso a questa politica.
9. Scegli Crea policy. Il sistema visualizza di nuovo la pagina Policies (Policy).
10. Ripetete questa procedura in ciascuna unità Account AWS che fornirà i dati di configurazione per l'account di recupero.

Crea il ruolo IAM

Utilizza la seguente procedura per creare un ruolo IAM utilizzando la console IAM. Completa la procedura in ciascuna Account AWS unità che fornirà i dati di configurazione per l'account di recupero.

Per creare un ruolo IAM

1. Accedi all'account di un Console di gestione AWS fornitore.
2. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
3. Nel riquadro di navigazione, scegli Ruoli, quindi scegli Crea politica.
4. Per Tipo di entità attendibile, seleziona Account AWS.
5. Nella Account AWSsezione, scegli Altro Account AWS.
6. Nel campo ID account, inserisci l'ID dell'account di recupero.
7. (Facoltativo) Come procedura consigliata di sicurezza per questo ruolo, scegli Richiedi ID esterno e inserisci una stringa.
8. Scegli Next (Successivo).
9. Nella pagina Aggiungi autorizzazioni, utilizza il campo Cerca per individuare la politica creata nella procedura precedente. Seleziona la casella accanto al suo nome.
10. Scegli Next (Successivo).
11. In Role name (Nome ruolo), immettere un nome.
12. (Facoltativo) In Descrizione, immetti una descrizione.
13. Per il passaggio 1: seleziona le entità attendibili, scegli Modifica. Sostituisci la politica di fiducia JSON predefinita con la seguente politica. Aggiorna ciascuno *example resource placeholder* con le informazioni del tuo account di recupero.

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "AWS":  
          "arn:aws:iam::111122223333:role/appconfig_role_in_retrieval_account"  
      },  
      "Action": "sts:AssumeRole"  
    }  
  ]  
}
```

```
        }
    ]
}
```

14. (Facoltativo) In Tag, aggiungi una o più coppie tag chiave-valore per organizzare, monitorare o controllare l'accesso per questo ruolo.
15. Scegliere Crea ruolo. Il sistema visualizza di nuovo la pagina Ruoli.
16. Cerca il ruolo che hai appena creato. Sceglilo. Nella sezione ARN, copia l'ARN. Queste informazioni verranno specificate nella procedura successiva.

Aggiungi sostituzioni di credenziali al manifesto

Dopo aver creato il ruolo IAM nel tuo account fornitore, aggiorna il manifesto nell'account di recupero. In particolare, aggiungi il blocco delle credenziali e l'ARN del ruolo IAM per recuperare i dati di configurazione dall'account del fornitore. Ecco il formato JSON:

```
{
  "vendor_application_name": "vendor_environment_name": "vendor_configuration_name": {
    "credentials": {
      "roleArn": "arn:partition:iam::vendor_account_ID:role/name_of_role_created_in_vendor_account",
      "roleExternalId": "string",
      "roleSessionName": "string",
      "credentialsDuration": "time_in_hours"
    }
  }
}
```

Ecco un esempio:

```
{
  "My2ndApp:Beta:MyEnableMobilePaymentsFeatureFlagConfiguration": {
    "credentials": {
      "roleArn": "arn:aws:us-west-1:iam::123456789012:role/MyTestRole",
      "roleExternalId": "00b148e2-4ea4-46a1-ab0f-c422b54d0aac",
      "roleSessionName": "Aws AppConfig Agent",
      "credentialsDuration": "2h"
    }
  }
}
```

## Verifica che il recupero da più account funzioni

È possibile verificare che l'agente sia in grado di recuperare i dati di configurazione da più account esaminando i registri dell'agente. AWS AppConfig Il registro INFO di livello per i dati iniziali recuperati per 'YourApplicationName:YourEnvironmentName:YourConfigurationName' è l'indicatore migliore per il successo dei recuperi. Se i recuperi non riescono, dovrebbe apparire un registro dei ERROR livelli che indica il motivo dell'errore. Di seguito è riportato un esempio di recupero riuscito da un account fornitore:

```
[appconfig agent] 2023/11/13 11:33:27 INFO AppConfig Agent 2.0.x
[appconfig agent] 2023/11/13 11:33:28 INFO serving on localhost:2772
[appconfig agent] 2023/11/13 11:33:28 INFO retrieved initial data for
'MyTestApplication:MyTestEnvironment:MyDenyListConfiguration' in XX.Xms
```

## Configurazione AWS AppConfig dell'agente per scrivere copie di configurazione su disco

È possibile configurare AWS AppConfig Agent per archiviare automaticamente una copia di una configurazione su disco in formato testo semplice. Questa funzionalità consente l'integrazione con i clienti con applicazioni che leggono i dati di configurazione dal disco AWS AppConfig.

Questa funzionalità non è progettata per essere utilizzata come funzionalità di backup della configurazione. AWS AppConfig L'agente non legge i file di configurazione copiati su disco. Se desideri eseguire il backup delle configurazioni su disco, consulta le variabili di PRELOAD\_BACKUP ambiente per [Using AWS AppConfig Agent with Amazon EC2](#) o [Using AWS AppConfig Agent with Amazon ECS BACKUP\\_DIRECTORY e Amazon EKS](#).

### Warning

Tieni presente le seguenti informazioni importanti su questa funzionalità:

- Le configurazioni salvate su disco vengono archiviate in testo semplice e sono leggibili dall'uomo. Non abilitare questa funzionalità per le configurazioni che includono dati sensibili.
- Questa funzionalità scrive sul disco locale. Utilizza il principio del privilegio minimo per le autorizzazioni del filesystem. Per ulteriori informazioni, consulta [Implementazione dell'accesso con privilegi minimi](#).

Per abilitare la scrittura (copia della configurazione su disco)

1. Modifica il manifesto.
2. Scegliete la configurazione che desiderate AWS AppConfig scrivere su disco e aggiungete un `writeTo` elemento. Ecco un esempio:

```
{  
  "application_name:environment_name:configuration_name": {  
    "writeTo": {  
      "path": "path_to_configuration_file"  
    }  
  }  
}
```

Ecco un esempio:

```
{  
  "MyTestApp:MyTestEnvironment:MyNewConfiguration": {  
    "writeTo": {  
      "path": "/tmp/aws-appconfig/mobile-app/beta/enable-mobile-payments"  
    }  
  }  
}
```

3. Salvare le modifiche. Il file configuration.json verrà aggiornato ogni volta che vengono distribuiti nuovi dati di configurazione.

Verifica che la copia di scrittura della configurazione su disco funzioni

È possibile verificare che le copie di una configurazione vengano scritte su disco esaminando i log dell' AWS AppConfig agente. La voce di INFO registro con la frase «INFO ha scritto la configurazione '*application:environment:configuration*' su *file\_path*' indica che l' AWS AppConfig agente scrive copie di configurazione su disco.

Ecco un esempio:

```
[appconfig agent] 2023/11/13 11:33:27 INFO AppConfig Agent 2.0.x  
[appconfig agent] 2023/11/13 11:33:28 INFO serving on localhost:2772  
[appconfig agent] 2023/11/13 11:33:28 INFO retrieved initial data for  
'MobileApp:Beta:EnableMobilePayments' in XX.Xms
```

```
[appconfig agent] 2023/11/13 17:05:49 INFO wrote configuration  
'MobileApp:Beta:EnableMobilePayments' to /tmp/configs/your-app/your-env/your-  
config.json
```

## Generazione di un client utilizzando la specifica OpenAPI

[È possibile utilizzare la seguente specifica YAML per OpenAPI per creare un SDK utilizzando uno strumento come OpenAPI Generator.](#) È possibile aggiornare questa specifica per includere valori codificati per l'applicazione, l'ambiente o la configurazione. Puoi anche aggiungere percorsi aggiuntivi (se disponi di più tipi di configurazione) e includere schemi di configurazione per generare modelli tipizzati specifici della configurazione per i tuoi client SDK. [Per ulteriori informazioni su OpenAPI \(noto anche come Swagger\), consulta la specifica OpenAPI.](#)

```
openapi: 3.0.0
info:
  version: 1.0.0
  title: AWS AppConfig Agent API
  description: An API model for AWS AppConfig Agent.
servers:
  - url: http://localhost:{port}/
    variables:
      port:
        default:
          '2772'
paths:
  /applications/{Application}/environments/{Environment}/configurations/{Configuration}:
    get:
      operationId: getConfiguration
      tags:
        - configuration
      parameters:
        - in: path
          name: Application
          description: The application for the configuration to get. Specify either the application name or the application ID.
          required: true
          schema:
            type: string
        - in: path
          name: Environment
```

```
        description: The environment for the configuration to get. Specify either the
environment name or the environment ID.
        required: true
        schema:
          type: string
- in: path
  name: Configuration
  description: The configuration to get. Specify either the configuration name
or the configuration ID.
  required: true
  schema:
    type: string
- in: query
  name: flag
  description: The key(s) of the feature flag(s) to retrieve. If not provided,
all flags are returned.
  required: false
  schema:
    type: array
    items:
      type: string
- in: header
  name: context
  description: Request context used to evaluate multi-variant feature flags.
  required: false
  schema:
    type: array
    items:
      type: string
      pattern: '^\\w+=\\w+$'
responses:
  200:
    headers:
      ConfigurationVersion:
        schema:
          type: string
    content:
      application/octet-stream:
        schema:
          type: string
          format: binary
        description: successful config retrieval
  400:
    description: BadRequestException
```

```
content:
  application/text:
    schema:
      $ref: '#/components/schemas/Error'
404:
  description: ResourceNotFoundException
  content:
    application/text:
      schema:
        $ref: '#/components/schemas/Error'
500:
  description: InternalServerError
  content:
    application/text:
      schema:
        $ref: '#/components/schemas/Error'
502:
  description: BadGatewayException
  content:
    application/text:
      schema:
        $ref: '#/components/schemas/Error'
504:
  description: GatewayTimeoutException
  content:
    application/text:
      schema:
        $ref: '#/components/schemas/Error'

components:
  schemas:
    Error:
      type: string
      description: The response error
```

## Utilizzo della modalità di sviluppo locale di Agent AWS AppConfig

AWS AppConfig Agent supporta una modalità di sviluppo locale. Se si abilita la modalità di sviluppo locale, l'agente legge i dati di configurazione da una directory specificata su disco. Non recupera i dati di configurazione da AWS AppConfig. È possibile simulare le distribuzioni di configurazione aggiornando i file nella directory specificata. Consigliamo la modalità di sviluppo locale per i seguenti casi d'uso:

- Prova diverse versioni di configurazione prima di distribuirle utilizzando AWS AppConfig.
- Prova diverse opzioni di configurazione per una nuova funzionalità prima di apportare modifiche al tuo repository di codice.
- Prova diversi scenari di configurazione per verificare che funzionino come previsto.

 **Warning**

Non utilizzate la modalità di sviluppo locale negli ambienti di produzione. Questa modalità non supporta importanti funzionalità di AWS AppConfig sicurezza come la convalida dell'implementazione e i rollback automatici.

Utilizzare la procedura seguente per configurare AWS AppConfig Agent per la modalità di sviluppo locale.

Per configurare AWS AppConfig Agent per la modalità di sviluppo locale

1. Installa l'agente utilizzando il metodo descritto per il tuo ambiente di calcolo. AWS AppConfig L'agente funziona con quanto segue Servizi AWS:
  - [AWS Lambda](#)
  - [Amazon EC2](#)
  - [Amazon ECS e Amazon EKS](#)
2. Se l'agente è in esecuzione, interrompilo.
3. Aggiungi LOCAL\_DEVELOPMENT\_DIRECTORY all'elenco delle variabili di ambiente. Specificate una directory sul filesystem che fornisca all'agente i permessi di lettura. Ad esempio, /tmp/local\_configs.
4. Crea un file nella directory. Il nome del file deve utilizzare il seguente formato:

*application\_name:environment\_name:configuration\_profile\_name*

Ecco un esempio:

`Mobile:Development:EnableMobilePaymentsFeatureFlagConfiguration`

### Note

- Per visualizzare esempi di feature flag che potete aggiungere a un file nella vostra LOCAL\_DEVELOPMENT\_DIRECTORY directory, vedete [Esempi di feature flag per la modalità di sviluppo locale di AWS AppConfig Agent](#).
- (Facoltativo) È possibile controllare il tipo di contenuto che l'agente restituisce per i dati di configurazione in base all'estensione assegnata al file. Ad esempio, se denominate il file con un'estensione.json, l'agente restituisce un tipo di contenuto application/json quando l'applicazione lo richiede. Se omettete l'estensione, l'agente la utilizza application/octet-stream per il tipo di contenuto. Se hai bisogno di un controllo preciso, puoi fornire un'estensione nel formato *type%subtype*. L'agente restituirà un tipo di contenuto di *.type/subtype*.

5. Esegui il comando seguente per riavviare l'agente e richiedere i dati di configurazione.

```
curl http://localhost:2772/applications/application_name/environments/environment_name/configurations/configuration_name
```

L'agente verifica le modifiche al file locale all'intervalllo di polling specificato per l'agente. Se l'intervalllo di sondaggio non è specificato, l'agente utilizza l'intervalllo predefinito di 45 secondi. Questo controllo a intervalli di sondaggio assicura che l'agente si comporti in un ambiente di sviluppo locale allo stesso modo in cui si comporta quando è configurato per interagire con il servizio. AWS AppConfig

### Note

Per distribuire una nuova versione di un file di configurazione di sviluppo locale, aggiorna il file con nuovi dati.

## Esempi di feature flag per la modalità di sviluppo locale di AWS AppConfig Agent

Questa sezione include esempi di feature flag che è possibile utilizzare con AWS AppConfig Agent in modalità di sviluppo locale. La modalità di sviluppo locale prevede che i dati dei feature flag vengano visualizzati nel formato del tempo di recupero dei dati. Il formato del tempo di recupero è il formato restituito quando il flag viene recuperato dall'[GetLatestConfiguration](#) API, che contiene solo il valore del flag. Il formato del tempo di recupero non include la definizione completa del flag

(passata all'API). [CreateHostedConfigurationVersion](#) La definizione completa di un flag contiene anche informazioni come i nomi e i valori degli attributi, i vincoli e lo stato di attivazione del flag.

## Argomenti

- [Esempi di bandiere con funzionalità di base](#)
- [Esempi di feature flag multivarianti](#)

### Esempi di bandiere con funzionalità di base

Usa i seguenti esempi di feature flag di base con AWS AppConfig Agent in modalità di sviluppo locale.

#### Note

Se desideri che l'agente riporti il tipo di contenuto dei dati dei feature flag locali `application/json` (come quando recupera i dati dei flag AWS AppConfig in un ambiente che non è in modalità di sviluppo locale), i file dei feature flag locali devono utilizzare l'estensione.json. Ad esempio, `Local:MyFeatureFlags:SampleB1.json`.

Esempio 1: un singolo flag che rappresenta un aggiornamento dell'interfaccia utente.

```
{  
  "ui_refresh": {  
    "enabled": true,  
    "new_styleguide_colors": true  
  }  
}
```

Esempio 2: più flag che rappresentano i flag delle funzionalità operative.

```
{  
  "background_worker": {  
    "enabled": true,  
    "num_threads": 4,  
    "queue_name": "MyWorkQueue"  
  },  
  "emergency_shutoff_switch": {  
    "enabled": false  
  },  
  "ui_refresh": {  
    "enabled": true,  
    "new_styleguide_colors": true  
  }  
}
```

```
"logger_settings": {  
    "enabled": true,  
    "level": "INFO"  
}  
}
```

## Esempi di feature flag multivarianti

Il formato del tempo di recupero di una configurazione di feature flag che contiene almeno un feature flag multi-variante è rappresentato come dati [Amazon Ion anziché come dati JSON](#). In questo formato, i flag multivarianti sono rappresentati come un elenco annotato e i flag di base sono rappresentati come una stringa annotata. Gli elementi dell'elenco di un flag multivariante sono una tupla (una lista con una lunghezza di due), che rappresenta una singola variante, o una stringa, che rappresenta la variante predefinita. All'interno di una tupla variante, il primo elemento è un'espressione s che rappresenta la regola della variante e il secondo elemento è una stringa che rappresenta il contenuto della variante.

Affinché l'agente interpreti correttamente questi file, i file dei feature flag locali devono utilizzare la seguente estensione: `application%ion%type=AWS.AppConfig.FeatureFlags`.

Ad esempio, `Local:MyFeatureFlags:SampleMV1.application%ion`

`%type=AWS.AppConfig.FeatureFlags`.

Esempio 1: un flag multivariante che rappresenta una versione a più livelli di una nuova funzionalità.

```
'tiered_release'::[  
    [  
        (or (and (eq $group "Tier1") (split by::$userId pct::1 seed::"2025.01.01")) (and  
            (eq $group "Tier2") (split by::$userId pct::7 seed::"2025.01.01"))),  
        '""{_variant": "ShowFeature", "enabled": true}'''  
    ],  
    '""{_variant": "HideFeature", "enabled": false}'''  
]
```

Esempio 2: più flag che rappresentano diversi display UX in base all'ID dell'utente. Le prime due bandiere sono multivarianti e la bandiera finale è di base.

```
'colorway'::[  
    [  
        (contains $userId "beta"),  
        '""{_variant": "BetaTesters", "enabled": true, "background": "blue", "foreground":  
            "red}'''  
    ]
```

```
],
[
  (split by::$userId pct::10),
  '''{"_variant": "SplitRollOutRedAndBlue", "enabled": true, "background": "blue",
"foreground": "red"}''',
],
  '''{"_variant": "default", "enabled": true, "background": "green", "foreground": "green"}''',
]
]

'simple_feature'::[
  [
    (contains $userId "beta"),
    '''{"_variant": "BetaTesters", "enabled": true}'''
  ],
  '''{"_variant": "default", "enabled": false}'''
]

'button_color':'''{"enabled": true, "color": "orange"}'''
```

## AWS AppConfig considerazioni sull'uso del browser e dei dispositivi mobili

I flag di funzionalità consentono di aggiornare istantaneamente l'esperienza delle pagine Web e delle applicazioni mobili, senza i costi, i rischi o la rigidità di una versione dell'App Store. Utilizzando i flag di funzionalità, puoi apportare gradualmente una modifica alla tua base di utenti in un momento a tua scelta. Se si verifica un errore, è possibile ripristinare immediatamente la modifica senza richiedere agli utenti di eseguire l'aggiornamento a una nuova versione del software. In breve, i flag di funzionalità offrono maggiore controllo e flessibilità durante l'implementazione delle modifiche all'applicazione.

Le sezioni seguenti descrivono importanti considerazioni sull'utilizzo dei flag di AWS AppConfig funzionalità con pagine Web e dispositivi mobili.

### Argomenti

- [Dati di configurazione e recupero delle bandiere](#)
- [Autenticazione e Amazon Cognito](#)
- [Caching](#)
- [Segmentazione](#)

- [Larghezza di banda \(casi d'uso mobili\)](#)
- [Casi d'uso aggiuntivi delle bandiere](#)

## Dati di configurazione e recupero delle bandiere

Per i casi d'uso su browser e dispositivi mobili, molti clienti scelgono di utilizzare un livello proxy tra il Web o l'applicazione mobile e AWS AppConfig. In questo modo, il volume delle AWS AppConfig chiamate viene separato dalla dimensione della base di utenti, riducendo così i costi. [Consente inoltre di sfruttare l'AWS AppConfig Agent, che ottimizza le prestazioni di recupero delle bandiere e supporta funzionalità come i flag multivarianti](#). AWS AppConfig AWS Lambda consiglia di utilizzare per creare il proxy. Invece di recuperare i flag direttamente da AWS AppConfig, configura l'[estensione AWS AppConfig Lambda per recuperare i flag delle funzionalità all'interno](#) di una funzione Lambda. Scrivi la funzione per accettare i parametri di AWS AppConfig recupero dalla richiesta di evento e restituire i dati di configurazione corrispondenti nella risposta Lambda. Esponi il tuo proxy a Internet utilizzando la funzione [URLsLambda](#).

Dopo aver configurato il proxy, considera la frequenza con cui recuperi i dati. I casi d'uso per dispositivi mobili in genere non richiedono intervalli di polling ad alta frequenza. Configura AWS AppConfig Agent in modo che aggiorni i dati con AWS AppConfig maggiore frequenza rispetto agli aggiornamenti dell'applicazione dal proxy.

## Autenticazione e Amazon Cognito

La funzione Lambda URLs supporta [due forme di controllo degli accessi](#) e AWS\_IAM. NONE Utilizzalo se preferisci implementare la tua autenticazione e autorizzazione nella tua funzione Lambda. NONE è anche l'opzione consigliata se il tuo caso d'uso consente di esporre l'endpoint al pubblico e i dati di configurazione non contengono dati sensibili. Per tutti gli altri casi d'uso, usa AWS\_IAM

### Important

Se esponi il tuo endpoint a Internet senza autenticazione, assicurati che dai dati di configurazione non vengano divulgati dati sensibili, tra cui informazioni di identificazione personale (PII) IDs, utenti o nomi di funzionalità non rilasciate.

Se scegli di utilizzare AWS\_IAM, dovrà gestire le credenziali con [Amazon Cognito](#). Per iniziare a usare Amazon Cognito, devi creare un pool di identità. Un pool di identità ti consente di fornire credenziali

a breve termine alla tua applicazione per utenti autenticati o ospiti. Dovrai aggiungere ruoli nel pool di identità che consentano agli utenti di utilizzare la funzione Lambda InvokeFunctionUrl per la tua. In questo modo, le istanze dell'applicazione possono accedere alle credenziali necessarie per recuperare i dati di configurazione.

Quando utilizzi Amazon Cognito nella tua applicazione, valuta la possibilità di utilizzare [AWS Amplify](#). Amplify mobile/web semplifica le interazioni tra le applicazioni e fornisce supporto AWS integrato per Amazon Cognito.

## Caching

Durante l'utilizzo AWS AppConfig, devi sempre memorizzare nella cache i dati di configurazione localmente sul dispositivo o nel browser. La memorizzazione nella cache offre i seguenti vantaggi:

- Migliora le prestazioni riducendo la latenza e il consumo della batteria
- Offre stabilità eliminando le dipendenze dall'accesso alla rete
- Riduce i costi riducendo la frequenza di recupero dei dati

Per i casi d'uso su dispositivi mobili, consigliamo di implementare cache in memoria e cache persistenti sul dispositivo. Configura l'applicazione in modo da tentare di recuperare la configurazione desiderata dalla cache in memoria e, se necessario, riprendila dal proxy. Una volta recuperata con successo dal proxy, aggiorna la cache in memoria e quindi invia la configurazione al dispositivo.

Utilizza un processo in background per scorrere la cache e aggiornare ogni configurazione. Quando recuperate la configurazione per la prima volta dopo l'avvio dell'applicazione, se il recupero non riesce, rimandate alla configurazione persistente (e usatela per effettuare il seeding della cache in memoria).

## Segmentazione

Quando utilizzate i flag di funzionalità, potrete voler segmentare l'esperienza di segnalazione delle funzionalità all'interno della vostra base clienti. A tal fine, fornisci un contesto alle chiamate di recupero dei flag e configura le regole per restituire diverse [varianti dei feature flag in base al contesto](#). Ad esempio, potresti avere una variante feature flag per gli utenti iOS 18.X, una variante per gli utenti iOS 17.X e un flag predefinito per tutte le altre versioni di iOS. Con le varianti, puoi configurare ogni versione iOS dell'applicazione in modo che utilizzi la stessa configurazione nello stesso ambiente, ma in base al contesto fornito nella chiamata di recupero (ad esempio, «version»: «i OS18 .1»), i dispositivi riceveranno la variante appropriata della configurazione.

### Note

Se si utilizzano varianti di AWS AppConfig feature flag per un caso d'uso mobile, è necessario utilizzare l' AWS AppConfig agente e un proxy per recuperare i flag di funzionalità.

Se scegli di non utilizzare AWS AppConfig Agent per recuperare i feature flag, puoi sfruttare gli AWS AppConfig [ambienti](#) per una segmentazione semplice e a bassa cardinalità. Un ambiente è un gruppo di distribuzione logico per i tuoi obiettivi. Oltre a partizionare le configurazioni in ambienti di sviluppo, test e produzione, è possibile suddividere la base clienti creando ambienti specifici per dispositivi mobili, come il tipo di dispositivo (tablet o telefono) o le versioni principali del sistema operativo. Con ambienti separati, è possibile implementare gli stessi o diversi set di dati di configurazione per soddisfare i requisiti particolari della propria base clienti.

## Larghezza di banda (casi d'uso mobili)

In generale, cerca di mantenere piccole le dimensioni di ogni set di bandiere. I casi d'uso mobili tendono a comportare vincoli di larghezza di banda ridotta. Ridurre al minimo le dimensioni dei dati ti aiuterà a mantenere un'esperienza coerente per tutta la tua base di utenti. Inoltre, considerate che, poiché i dispositivi mobili spesso operano tra ambienti con larghezza di banda ridotta e senza larghezza di banda, la memorizzazione nella cache sul dispositivo è fondamentale. Anche il codice applicativo che fallisce correttamente se non è possibile recuperare i dati di configurazione è fondamentale.

## Casi d'uso aggiuntivi delle bandiere

La potenza dei flag di funzionalità va oltre la praticità del rilascio delle funzionalità. È possibile utilizzare flag operativi di lunga data per migliorare la postura operativa dell'applicazione. Ad esempio, è possibile creare un interruttore di monitoraggio delle prestazioni che emetta metriche e dati di debug aggiuntivi durante un evento. In alternativa, potresti voler mantenere e modificare le frequenze di aggiornamento delle applicazioni per un segmento della tua base clienti.

## Recupero dei dati di configurazione senza Agent AWS AppConfig

Il modo consigliato per recuperare i dati di configurazione AWS AppConfig è utilizzare l'agente sviluppato e gestito da Amazon. AWS AppConfig Con l'agente, puoi memorizzare nella cache i dati di configurazione localmente e interrogare in modo asincrono il servizio Data Plane per verificare la

presenza di aggiornamenti. AWS AppConfig Questo caching/polling processo garantisce che i dati di configurazione siano sempre disponibili per l'applicazione, riducendo al minimo latenza e costi. Se preferisci non utilizzare l'agente, puoi effettuare chiamate pubbliche APIs direttamente dal servizio AWS AppConfig Data Plane.

Il servizio data plane utilizza due azioni API [StartConfigurationSession](#) [GetLatestConfiguration](#). Il servizio del piano dati utilizza anche [endpoint separati](#) dal piano AWS AppConfig di controllo.

#### Note

Il servizio del piano dati sostituisce il precedente processo di recupero dei dati di configurazione utilizzando l'azione API `GetConfiguration`. L'`GetConfiguration` API è obsoleta.

### Come funziona

Ecco come funziona il processo di chiamata diretta AWS AppConfig APIs tramite il servizio data plane.

L'applicazione recupera i dati di configurazione stabilendo prima una sessione di configurazione utilizzando l'operazione [StartConfigurationSession](#) API. Il client della sessione effettua quindi chiamate periodiche per [GetLatestConfiguration](#) per verificare e recuperare i dati più recenti disponibili.

Durante la chiamata `StartConfigurationSession`, il codice invia le seguenti informazioni:

- Identificatori (ID o nome) di un' AWS AppConfig applicazione, di un ambiente e di un profilo di configurazione monitorati dalla sessione.
- (Facoltativo) La quantità minima di tempo che il client della sessione deve attendere tra una chiamata e l'altra. `GetLatestConfiguration`

In risposta, AWS AppConfig fornisce un `InitialConfigurationToken` messaggio da dare al client della sessione e da utilizzare la prima volta che effettua una chiamata `GetLatestConfiguration` per quella sessione.

#### Important

Questo token deve essere usato solo una volta nella prima chiamata `aGetLatestConfiguration`. È necessario utilizzare il nuovo token nella

`GetLatestConfiguration` risposta (`NextPollConfigurationToken`) in ogni chiamata successiva a `GetLatestConfiguration`. Per supportare casi d'uso prolungati con sondaggi, i token sono validi per un massimo di 24 ore. Se una `GetLatestConfiguration` chiamata utilizza un token scaduto, il sistema ritorna `BadRequestException`

Durante la chiamata `GetLatestConfiguration`, il codice cliente invia il `ConfigurationToken` valore più recente a sua disposizione e riceve in risposta:

- `NextPollConfigurationToken`: il `ConfigurationToken` valore da utilizzare nella chiamata successiva a `GetLatestConfiguration`.
- `NextPollIntervalInSeconds`: la durata che il client deve attendere prima di effettuare la chiamata successiva `GetLatestConfiguration`.
- La configurazione: i dati più recenti destinati alla sessione. Questo campo può essere vuoto se il client dispone già dell'ultima versione della configurazione.

#### Important

Prendi nota delle seguenti informazioni importanti.

- L'[StartConfigurationSession](#) API deve essere chiamata solo una volta per applicazione, ambiente, profilo di configurazione e client per stabilire una sessione con il servizio. Questa operazione viene in genere eseguita all'avvio dell'applicazione o immediatamente prima del primo recupero di una configurazione.
- Se la configurazione viene distribuita utilizzando `aKmsKeyId`, la richiesta di ricezione della configurazione deve includere l'autorizzazione alla chiamata. `kms:Decrypt` Per ulteriori informazioni, [consulta Decrypt](#) nel riferimento API AWS Key Management Service
- L'operazione API precedentemente utilizzata per recuperare i dati di `GetConfiguration`, è obsoleta. L'operazione `GetConfiguration` API non supporta configurazioni crittografate.

## (Esempio) Recupero di una configurazione chiamando AWS AppConfig APIs

L'AWS CLI esempio seguente mostra come recuperare i dati di configurazione utilizzando le AWS AppConfig operazioni Data e API. `StartConfigurationSession` `GetLatestConfiguration` Il primo comando avvia una sessione di configurazione. Questa chiamata include IDs (o i nomi) dell'AWS AppConfig applicazione, dell'ambiente e del profilo di configurazione. L'API restituisce un file `InitialConfigurationToken` usato per recuperare i dati di configurazione.

```
aws appconfigdata start-configuration-session \
  --application-identifier application_name_or_ID \
  --environment-identifier environment_name_or_ID \
  --configuration-profile-identifier configuration_profile_name_or_ID
```

Il sistema risponde con informazioni nel formato seguente.

```
{  
  "InitialConfigurationToken": initial configuration token  
}
```

Dopo aver avviato una sessione, usa [InitialConfigurationToken](#) to call [GetLatestConfiguration](#) per recuperare i dati di configurazione. I dati di configurazione vengono salvati nel `mydata.json` file.

```
aws appconfigdata get-latest-configuration \
  --configuration-token initial configuration token mydata.json
```

La prima chiamata `GetLatestConfiguration` utilizza il codice `ConfigurationToken` ottenuto da `StartConfigurationSession`. Vengono restituite le seguenti informazioni.

```
{  
  "NextPollConfigurationToken" : next configuration token,  
  "ContentType" : content type of configuration,  
  "NextPollIntervalInSeconds" : 60  
}
```

Le chiamate successive `GetLatestConfiguration` devono `NextPollConfigurationToken` provenire dalla risposta precedente.

```
aws appconfigdata get-latest-configuration \
```

```
--configuration-token next configuration token mydata.json
```

### Important

Tieni presente i seguenti dettagli importanti sul funzionamento dell'GetLatestConfigurationAPI:

- La GetLatestConfiguration risposta include una Configuration sezione che mostra i dati di configurazione. La Configuration sezione viene visualizzata solo se il sistema trova dati di configurazione nuovi o aggiornati. Se il sistema non trova dati di configurazione nuovi o aggiornati, i Configuration dati sono vuoti.
- Ne ricevi uno nuovo ConfigurationToken in ogni risposta daGetLatestConfiguration.
- Si consiglia di ottimizzare la frequenza di polling delle chiamate API GetLatestConfiguration in base al budget, alla frequenza prevista delle distribuzioni di configurazione e al numero di destinazioni per una configurazione.

# Estendere i AWS AppConfig flussi di lavoro utilizzando le estensioni

Un'estensione aumenta la capacità di inserire logica o comportamento in punti diversi durante il AWS AppConfig flusso di lavoro di creazione o distribuzione di una configurazione. Ad esempio, puoi utilizzare le estensioni per eseguire i seguenti tipi di attività (solo per citarne alcune):

- Invia una notifica a un argomento di Amazon Simple Notification Service (Amazon SNS) quando viene distribuito un profilo di configurazione.
- Scorri il contenuto di un profilo di configurazione alla ricerca di dati sensibili prima dell'inizio della distribuzione.
- Crea o aggiorna un problema di Atlassian Jira ogni volta che viene apportata una modifica a un flag di funzionalità.
- Unisci i contenuti di un servizio o di una fonte di dati ai dati di configurazione quando avvii una distribuzione.
- Esegui il backup di una configurazione in un bucket Amazon Simple Storage Service (Amazon S3) ogni volta che viene distribuita una configurazione.

Puoi associare questi tipi di attività ad AWS AppConfig applicazioni, ambienti e profili di configurazione.

## Indice

- [Comprendere AWS AppConfig le estensioni](#)
- [Lavorare con le AWS estensioni create](#)
- [Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig](#)

## Comprendere AWS AppConfig le estensioni

Questo argomento introduce i concetti e la terminologia relativi alle AWS AppConfig estensioni. Le informazioni vengono discusse nel contesto di ogni passaggio necessario per configurare e utilizzare AWS AppConfig le estensioni.

### Argomenti

- [Passaggio 1: Stabilisci cosa vuoi fare con le estensioni](#)

- [Passo 2: Determina quando vuoi che l'estensione venga eseguita](#)
- [Passaggio 3: Creare un'associazione di estensioni](#)
- [Passaggio 4: Implementa una configurazione e verifica che le azioni di estensione vengano eseguite](#)

## Passaggio 1: Stabilisci cosa vuoi fare con le estensioni

Vuoi ricevere una notifica su un webhook che invia messaggi a Slack ogni volta che viene completata una AWS AppConfig distribuzione? Vuoi eseguire il backup di un profilo di configurazione in un bucket Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) prima di implementare una configurazione? Vuoi cancellare i dati di configurazione alla ricerca di informazioni sensibili prima che la configurazione venga distribuita? È possibile utilizzare le estensioni per eseguire questo tipo di attività e altro ancora. È possibile creare estensioni personalizzate o utilizzare le estensioni AWS create incluse in AWS AppConfig.

### Note

Nella maggior parte dei casi d'uso, per creare un'estensione personalizzata, è necessario creare una AWS Lambda funzione per eseguire qualsiasi calcolo ed elaborazione definiti nell'estensione. Per ulteriori informazioni, consulta [Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig](#).

Le seguenti estensioni AWS create possono aiutarti a integrare rapidamente le distribuzioni di configurazione con altri servizi. Puoi utilizzare queste estensioni nella AWS AppConfig console o richiamando [le azioni dell'API delle estensioni](#) direttamente da AWS CLI AWS Strumenti per PowerShell, o dall'SDK.

Estensione	Description
<a href="#">Amazon sta CloudWatch evidentemente testando A/B</a>	Questa estensione consente all'applicazione di assegnare variazioni alle sessioni utente localmente anziché richiamare l' <a href="#">EvaluateFeature</a> operazione. Per ulteriori informazioni, consulta <a href="#">Utilizzo dell'estensione Amazon CloudWatch Evidently</a> .

Estensione	Description
<a href="#">AWS AppConfig eventi di distribuzione a EventBridge</a>	Questa estensione invia gli eventi al bus degli eventi EventBridge predefinito quando viene distribuita una configurazione.
<a href="#">AWS AppConfig eventi di distribuzione su Amazon Simple Notification Service (Amazon SNS)</a>	Questa estensione invia messaggi a un argomento di Amazon SNS specificato quando viene distribuita una configurazione.
<a href="#">AWS AppConfig eventi di distribuzione su Amazon Simple Queue Service (Amazon SQS)</a>	Questa estensione inserisce i messaggi nella coda di Amazon SQS quando viene distribuita una configurazione.
<a href="#">Estensione di integrazione: Atlassian Jira</a>	<a href="#">Queste estensioni consentono di AWS AppConfig creare e aggiornare problemi ogni volta che si apportano modifiche a un feature flag.</a>

## Passo 2: Determina quando vuoi che l'estensione venga eseguita

Un'estensione definisce una o più azioni da eseguire durante un AWS AppConfig flusso di lavoro. Ad esempio, l'AWS AppConfig deployment events to Amazon SNS estensione AWS creata include un'azione per inviare una notifica a un argomento di Amazon SNS. Ogni azione viene richiamata quando interagisci con AWS AppConfig o quando AWS AppConfig esegui un processo per tuo conto. Questi sono chiamati punti d'azione. AWS AppConfig le estensioni supportano i seguenti punti di azione:

punti d'azione PRE\_\*: le azioni di estensione configurate sui punti di PRE\_\* azione vengono applicate dopo la convalida della richiesta, ma prima dell' AWS AppConfig esecuzione dell'attività corrispondente al nome del punto di azione. Queste invocazioni di azioni vengono elaborate contemporaneamente a una richiesta. Se viene effettuata più di una richiesta, le chiamate alle azioni vengono eseguite in sequenza. Si noti inoltre che i punti di PRE\_\* azione ricevono e possono modificare il contenuto di una configurazione. PRE\_\* i punti di azione possono anche rispondere a un errore e impedire che si verifichi un'azione.

- PRE\_CREATE\_HOSTED\_CONFIGURATION\_VERSION

- PRE\_START\_DEPLOYMENT

Punti di azione ON\_\*: un'estensione può anche essere eseguita in parallelo a un AWS AppConfig flusso di lavoro utilizzando un punto ON\_\* azione. ON\_\*i punti di azione vengono richiamati in modo asincrono. ON\_\*i punti di azione non ricevono il contenuto di una configurazione. Se un'estensione riscontra un errore durante un punto di ON\_\* azione, il servizio ignora l'errore e continua il flusso di lavoro.

- ON\_DEPLOYMENT\_START
- ON\_DEPLOYMENT\_STEP
- ON\_DEPLOYMENT\_BAKING
- ON\_DEPLOYMENT\_COMPLETE
- ON\_DEPLOYMENT\_ROLLED\_BACK

Punti di azione AT\_\*: le azioni di estensione configurate sui punti di AT\_\* azione vengono richiamate in modo sincrono e parallelo a un flusso di lavoro. AWS AppConfig Se un'estensione riscontra un errore durante un punto di AT\_\* azione, il servizio interrompe il flusso di lavoro e ripristina la distribuzione.

- AT\_DEPLOYMENT\_TICK

## Passaggio 3: Creare un'associazione di estensioni

Per creare un'estensione o configurare un' AWS estensione creata, si definiscono i punti di azione che richiamano un'estensione quando viene utilizzata una AWS AppConfig risorsa specifica. Ad esempio, puoi scegliere di eseguire l'AWS AppConfig deployment events to Amazon SNSestensione e ricevere notifiche su un argomento di Amazon SNS ogni volta che viene avviata una distribuzione di configurazione per un'applicazione specifica. La definizione dei punti di azione che richiamano un'estensione per una AWS AppConfig risorsa specifica viene chiamata associazione di estensioni. Un'associazione di estensione è una relazione specifica tra un'estensione e una AWS AppConfig risorsa, ad esempio un'applicazione o un profilo di configurazione.

Una singola AWS AppConfig applicazione può includere più ambienti e profili di configurazione. Se si associa un'estensione a un'applicazione o a un ambiente, AWS AppConfig richiama l'estensione per tutti i flussi di lavoro relativi all'applicazione o alle risorse dell'ambiente, se applicabile.

Ad esempio, supponiamo di avere un' AWS AppConfig applicazione chiamata MobileApps che include un profilo di configurazione chiamato AccessList. Supponiamo che l' MobileApps applicazione includa ambienti beta, di integrazione e di produzione. Crei un'associazione di estensione per l' AWS estensione di notifica Amazon SNS creata e associ l'estensione all' MobileApps applicazione. L'estensione di notifica di Amazon SNS viene richiamata ogni volta che la configurazione viene distribuita per l'applicazione in uno dei tre ambienti.

 Note

Non è necessario creare un'estensione per utilizzare le estensioni AWS create, ma è necessario creare un'associazione di estensioni.

## Passaggio 4: Implementa una configurazione e verifica che le azioni di estensione vengano eseguite

Dopo aver creato un'associazione, quando viene creata una configurazione ospitata o viene distribuita una configurazione, AWS AppConfig richiama l'estensione ed esegue le azioni specificate. Quando viene richiamata un'estensione, se il sistema riscontra un errore durante un punto di PRE - \* azione, AWS AppConfig restituisce informazioni su tale errore.

## Lavorare con le AWS estensioni create

AWS AppConfig include le seguenti estensioni AWS create. Queste estensioni possono aiutarti a integrare il AWS AppConfig flusso di lavoro con altri servizi. Puoi utilizzare queste estensioni in Console di gestione AWS o chiamando [le azioni dell'API delle estensioni](#) direttamente da AWS CLI, AWS Strumenti per PowerShell, o dall'SDK.

Estensione	Description
<a href="#">Amazon sta CloudWatch evidentemente testando A/B</a>	Questa estensione consente all'applicazione di assegnare variazioni alle sessioni utente localmente anziché richiamare l' <a href="#">EvaluateFeature</a> operazione. Per ulteriori informazioni, consulta <a href="#">Utilizzo dell'estensione Amazon CloudWatch Evidently</a> .

Estensione	Description
<a href="#">AWS AppConfig eventi di distribuzione a EventBridge</a>	Questa estensione invia gli eventi al bus degli eventi EventBridge predefinito quando viene distribuita una configurazione.
<a href="#">AWS AppConfig eventi di distribuzione su Amazon Simple Notification Service (Amazon SNS)</a>	Questa estensione invia messaggi a un argomento di Amazon SNS specificato quando viene distribuita una configurazione.
<a href="#">AWS AppConfig eventi di distribuzione su Amazon Simple Queue Service (Amazon SQS)</a>	Questa estensione inserisce i messaggi nella coda di Amazon SQS quando viene distribuita una configurazione.
<a href="#">Estensione di integrazione: Atlassian Jira</a>	<a href="#">Queste estensioni consentono di AWS AppConfig creare e aggiornare problemi ogni volta che si apportano modifiche a un feature flag.</a>

## Utilizzo dell'estensione Amazon CloudWatch Evidently

Puoi usare Amazon CloudWatch Evidently per convalidare nuove funzionalità in modo sicuro offrendole disponibili a una percentuale specifica di utenti durante il rollout della funzionalità. È possibile monitorare le prestazioni della nuova funzionalità per aiutarti a decidere quando aumentare il traffico verso gli utenti. Ciò consente di ridurre i rischi e identificare le conseguenze non intenzionali prima di avviare completamente la funzionalità. Puoi anche condurre A/B esperimenti per prendere decisioni sulla progettazione delle funzionalità sulla base di prove e dati.

L' AWS AppConfig estensione per CloudWatch Evidently consente all'applicazione di assegnare variazioni alle sessioni utente localmente anziché richiamare l'[EvaluateFeature](#) operazione. Una sessione locale mitiga i rischi di latenza e disponibilità associati a una chiamata API. Per informazioni su come configurare e utilizzare l'estensione, consulta [Esegui lanci ed A/B esperimenti con CloudWatch Evidently](#) nella Amazon CloudWatch User Guide.

## Utilizzo degli eventi AWS AppConfig di distribuzione nell' EventBridge estensione Amazon

L'AWS AppConfig deployment events to Amazon EventBridge estensione è un' AWS estensione creata che ti aiuta a monitorare e ad agire sul flusso di lavoro di AWS AppConfig configurazione e distribuzione. L'estensione invia notifiche di eventi al bus degli eventi EventBridge predefinito ogni volta che viene distribuita una configurazione. Dopo aver associato l'estensione a una delle AWS AppConfig applicazioni, degli ambienti o dei profili di configurazione, AWS AppConfig invia notifiche di eventi al bus degli eventi dopo l'inizio, la fine e il rollback di ogni implementazione della configurazione.

Se desideri un maggiore controllo sui punti di azione che inviano EventBridge le notifiche, puoi creare un'estensione personalizzata e inserire il bus degli eventi EventBridge predefinito Amazon Resource Name (ARN) per il campo URI. Per informazioni sulla creazione di un'estensione, consulta [Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig](#).

### Important

Questa estensione supporta solo il bus degli eventi EventBridge predefinito.

## Utilizzo dell'estensione

Per utilizzare l'AWS AppConfig deployment events to Amazon EventBridge estensione, devi innanzitutto collegarla a una delle tue AWS AppConfig risorse creando un'associazione di estensioni. L'associazione viene creata utilizzando la AWS AppConfig console o l'azione [CreateExtensionAssociation](#) API. Quando si crea l'associazione, si specifica l'ARN di un' AWS AppConfig applicazione, di un ambiente o di un profilo di configurazione. Se si associa l'estensione a un'applicazione o a un ambiente, viene inviata una notifica di evento per qualsiasi profilo di configurazione contenuto nell'applicazione o nell'ambiente specificato.

Dopo aver creato l'associazione, quando viene distribuita una configurazione per la AWS AppConfig risorsa specificata, AWS AppConfig richiama l'estensione e invia notifiche in base ai punti di azione specificati nell'estensione.

### Note

Questa estensione viene richiamata dai seguenti punti di azione:

- ON\_DEPLOYMENT\_START
- ON\_DEPLOYMENT\_COMPLETE
- ON\_DEPLOYMENT\_ROLLED\_BACK

Non puoi personalizzare i punti di azione per questa estensione. Per richiamare diversi punti di azione, puoi creare la tua estensione. Per ulteriori informazioni, consulta [Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig](#).

Utilizzate le seguenti procedure per creare un'associazione di AWS AppConfig estensioni utilizzando la AWS Systems Manager console o il AWS CLI.

Per creare un'associazione di estensioni (console)

1. Apri la AWS Systems Manager console su <https://console.aws.amazon.com/systems-manager/appconfig/>.
2. Nel pannello di navigazione, scegli AWS AppConfig.
3. Nella scheda Estensioni, scegli Aggiungi alla risorsa.
4. Nella sezione Dettagli della risorsa di estensione, per Tipo di risorsa, scegli un tipo di AWS AppConfig risorsa. A seconda della risorsa scelta, AWS AppConfig ti chiede di scegliere altre risorse.
5. Scegli Crea associazione alla risorsa.

Ecco un esempio di evento inviato EventBridge quando viene richiamata l'estensione.

```
{  
  "version": "0",  
  "id": "c53dbd72-c1a0-2302-9ed6-c076e9128277",  
  "detail-type": "On Deployment Complete",  
  "source": "aws.appconfig",  
  "account": "111122223333",  
  "time": "2022-07-09T01:44:15Z",  
  "region": "us-east-1",  
  "resources": [  
    "arn:aws:appconfig:us-east-1:111122223333:extensionassociation/z763ff5"  
  ],  
  "detail": {
```

```
"InvocationId":"5tfjcg",
"Parameters":{

},
"Type":"OnDeploymentComplete",
"Application":{

  "Id":"ba8toh7",
  "Name":"MyApp"
},
"Environment":{

  "Id":"pgil2o7",
  "Name":"MyEnv"
},
"ConfigurationProfile":{

  "Id":"ga3tqep",
  "Name":"MyConfigProfile"
},
"DeploymentNumber":1,
"ConfigurationVersion":"1"
}
}
```

## Utilizzo degli eventi AWS AppConfig di distribuzione nell'estensione Amazon SNS

L'AWS AppConfig deployment events to Amazon SNS estensione è un' AWS estensione creata che ti aiuta a monitorare il flusso di lavoro di configurazione e distribuzione e ad agire di conseguenza. AWS AppConfig L'estensione pubblica messaggi su un argomento di Amazon SNS ogni volta che viene distribuita una configurazione. Dopo aver associato l'estensione a una delle tue AWS AppConfig applicazioni, ambienti o profili di configurazione, AWS AppConfig pubblica un messaggio sull'argomento dopo ogni inizio, fine e ripristino della configurazione.

Se desideri un maggiore controllo su quali punti di azione inviano notifiche Amazon SNS, puoi creare un'estensione personalizzata e inserire un argomento Amazon SNS Amazon Resource Name (ARN) per il campo URI. Per informazioni sulla creazione di un'estensione, consulta. [Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig](#)

### Utilizzo dell'estensione

Questa sezione descrive come utilizzare l'AWS AppConfig deployment events to Amazon SNS estensione.

## Fase 1: AWS AppConfig Configurare la pubblicazione dei messaggi su un argomento

Aggiungi una policy di controllo degli accessi al tuo argomento Amazon SNS: concessione delle autorizzazioni di pubblicazione AWS AppConfig (appconfig.amazonaws.com). sns:Publish Per ulteriori informazioni, consulta [Casi di esempio per il controllo degli accessi di Amazon SNS](#).

## Passaggio 2: creare un'associazione di estensioni

Collega l'estensione a una delle tue AWS AppConfig risorse creando un'associazione di estensioni. L'associazione viene creata utilizzando la AWS AppConfig console o l'azione [CreateExtensionAssociation](#) API. Quando si crea l'associazione, si specifica l'ARN di un' AWS AppConfig applicazione, di un ambiente o di un profilo di configurazione. Se si associa l'estensione a un'applicazione o a un ambiente, viene inviata una notifica per qualsiasi profilo di configurazione contenuto nell'applicazione o nell'ambiente specificato. Quando crei l'associazione, devi inserire un valore per il `topicArn` parametro che contiene l'ARN dell'argomento Amazon SNS che desideri utilizzare.

Dopo aver creato l'associazione, quando viene distribuita una configurazione per la AWS AppConfig risorsa specificata, AWS AppConfig richiama l'estensione e invia notifiche in base ai punti di azione specificati nell'estensione.

### Note

Questa estensione viene richiamata dai seguenti punti di azione:

- ON\_DEPLOYMENT\_START
- ON\_DEPLOYMENT\_COMPLETE
- ON\_DEPLOYMENT\_ROLLED\_BACK

Non puoi personalizzare i punti di azione per questa estensione. Per richiamare diversi punti di azione, puoi creare la tua estensione. Per ulteriori informazioni, consulta [Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig](#).

Utilizzate le seguenti procedure per creare un'associazione di AWS AppConfig estensioni utilizzando la AWS Systems Manager console o il AWS CLI.

## Per creare un'associazione di estensioni (console)

1. Apri la AWS Systems Manager console su <https://console.aws.amazon.com/systems-manager/appconfig/>.
2. Nel pannello di navigazione, scegli AWS AppConfig.
3. Nella scheda Estensioni, scegli Aggiungi alla risorsa.
4. Nella sezione Dettagli della risorsa di estensione, per Tipo di risorsa, scegli un tipo di AWS AppConfig risorsa. A seconda della risorsa scelta, AWS AppConfig ti chiede di scegliere altre risorse.
5. Scegli Crea associazione alla risorsa.

Ecco un esempio del messaggio inviato all'argomento Amazon SNS quando viene richiamata l'estensione.

```
{  
  "Type": "Notification",  
  "MessageId": "ae9d702f-9a66-51b3-8586-2b17932a9f28",  
  "TopicArn": "arn:aws:sns:us-east-1:111122223333:MySNSTopic",  
  "Message": {  
    "InvocationId": "7itcaxp",  
    "Parameters": {  
      "topicArn": "arn:aws:sns:us-east-1:111122223333:MySNSTopic"  
    },  
    "Application": {  
      "Id": "1a2b3c4d",  
      "Name": MyApp  
    },  
    "Environment": {  
      "Id": "1a2b3c4d",  
      "Name": MyEnv  
    },  
    "ConfigurationProfile": {  
      "Id": "1a2b3c4d",  
      "Name": "MyConfigProfile"  
    },  
    "Description": null,  
    "DeploymentNumber": "3",  
    "ConfigurationVersion": "1",  
    "Type": "OnDeploymentComplete"  
  },  
}
```

```
"Timestamp": "2022-06-30T20:26:52.067Z",
"SignatureVersion": "1",
"Signature": "<...>",
"SigningCertURL": "<...>",
"UnsubscribeURL": "<...>",
"MessageAttributes": {
    "MessageType": {
        "Type": "String",
        "Value": "OnDeploymentStart"
    }
}
```

## Utilizzo degli eventi AWS AppConfig di distribuzione nell'estensione Amazon SQS

L'AWS AppConfig deployment events to Amazon SQS estensione è un' AWS estensione creata che ti aiuta a monitorare e ad agire sul flusso di lavoro di configurazione e distribuzione. AWS AppConfig L'estensione inserisce i messaggi nella coda di Amazon Simple Queue Service (Amazon SQS) ogni volta che viene distribuita una configurazione. Dopo aver associato l'estensione a una delle tue AWS AppConfig applicazioni, ambienti o profili di configurazione, AWS AppConfig inserisce un messaggio nella coda dopo ogni inizio, fine e rollback della configurazione.

Se desideri un maggiore controllo su quali punti di azione inviano notifiche Amazon SQS, puoi creare un'estensione personalizzata e inserire una coda Amazon SQS Amazon Resource Name (ARN) per il campo URI. Per informazioni sulla creazione di un'estensione, consulta. [Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig](#)

### Utilizzo dell'estensione

Questa sezione descrive come utilizzare l'AWS AppConfig deployment events to Amazon SQS estensione.

#### Fase 1: Configurazione AWS AppConfig per accodare i messaggi

Aggiungi una policy Amazon SQS alla tua coda Amazon SQS AWS AppConfig granting `appconfig.amazonaws.com()` send message permissions (`sq:SendMessage`). Per ulteriori informazioni, consulta [Esempi di base delle politiche di Amazon SQS](#).

#### Passaggio 2: creare un'associazione di estensioni

Collega l'estensione a una delle tue AWS AppConfig risorse creando un'associazione di estensioni. L'associazione viene creata utilizzando la AWS AppConfig console o l'azione [CreateExtensionAssociation](#) API. Quando si crea l'associazione, si specifica l'ARN di un'AWS AppConfig applicazione, di un ambiente o di un profilo di configurazione. Se si associa l'estensione a un'applicazione o a un ambiente, viene inviata una notifica per qualsiasi profilo di configurazione contenuto nell'applicazione o nell'ambiente specificato. Quando crei l'associazione, devi inserire un `Here` parametro che contenga l'ARN della coda Amazon SQS che desideri utilizzare.

Dopo aver creato l'associazione, quando viene creata o distribuita una configurazione per la AWS AppConfig risorsa specificata, AWS AppConfig richiama l'estensione e invia notifiche in base ai punti di azione specificati nell'estensione.

#### Note

Questa estensione viene richiamata dai seguenti punti di azione:

- ON\_DEPLOYMENT\_START
- ON\_DEPLOYMENT\_COMPLETE
- ON\_DEPLOYMENT\_ROLLED\_BACK

Non puoi personalizzare i punti di azione per questa estensione. Per richiamare diversi punti di azione, puoi creare la tua estensione. Per ulteriori informazioni, consulta [Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig](#).

Utilizzate le seguenti procedure per creare un'associazione di AWS AppConfig estensioni utilizzando la AWS Systems Manager console o il AWS CLI.

Per creare un'associazione di estensioni (console)

1. Apri la AWS Systems Manager console su <https://console.aws.amazon.com/systems-manager/appconfig/>.
2. Nel pannello di navigazione, scegli AWS AppConfig.
3. Nella scheda Estensioni, scegli Aggiungi alla risorsa.
4. Nella sezione Dettagli della risorsa di estensione, per Tipo di risorsa, scegli un tipo di AWS AppConfig risorsa. A seconda della risorsa scelta, AWS AppConfig ti chiede di scegliere altre risorse.

## 5. Scegli Crea associazione alla risorsa.

Ecco un esempio del messaggio inviato alla coda di Amazon SQS quando viene richiamata l'estensione.

```
{  
  "InvocationId": "7itcaxp",  
  "Parameters": {  
    "queueArn": "arn:aws:sqs:us-east-1:111122223333:MySQSQueue"  
  },  
  "Application": {  
    "Id": "1a2b3c4d",  
    "Name": "MyApp"  
  },  
  "Environment": {  
    "Id": "1a2b3c4d",  
    "Name": "MyEnv"  
  },  
  "ConfigurationProfile": {  
    "Id": "1a2b3c4d",  
    "Name": "MyConfigProfile"  
  },  
  "Description": null,  
  "DeploymentNumber": "3",  
  "ConfigurationVersion": "1",  
  "Type": "OnDeploymentComplete"  
}
```

## Utilizzo dell'estensione Atlassian Jira per AWS AppConfig

Grazie all'integrazione con Atlassian Jira, AWS AppConfig puoi creare e aggiornare problemi nella console Atlassian ogni volta che apporti modifiche a un flag di funzionalità nel tuo caso. Account AWS Regione AWS Ogni numero di Jira include il nome del flag, l'ID dell'applicazione, l'ID del profilo di configurazione e i valori dei flag. Dopo aver aggiornato, salvato e distribuito le modifiche ai flag, Jira aggiorna i problemi esistenti con i dettagli della modifica.

### Note

Jira aggiorna i problemi ogni volta che crei o aggiorni un feature flag. Jira aggiorna anche i problemi quando elimini un attributo di flag a livello di figlio da un flag di livello principale. Jira non registra informazioni quando elimini un flag di livello principale.

Per configurare l'integrazione, devi fare quanto segue:

- [Configurazione delle autorizzazioni per AWS AppConfig l'integrazione con Jira](#)
- [Configurazione dell'applicazione di integrazione AWS AppConfig Jira](#)

## Configurazione delle autorizzazioni per AWS AppConfig l'integrazione con Jira

Quando configuri AWS AppConfig l'integrazione con Jira, specifichi le credenziali per un utente. In particolare, inserisci l'ID della chiave di accesso e la chiave segreta dell'utente nell'applicazione AWS AppConfig per Jira. Questo utente concede a Jira il permesso di comunicare con AWS AppConfig. AWS AppConfig utilizza queste credenziali una sola volta per stabilire un'associazione tra AWS AppConfig e Jira. Le credenziali non vengono archiviate. Puoi rimuovere l'associazione disinstallando l'applicazione AWS AppConfig per Jira.

L'account utente richiede una politica di autorizzazione che includa le seguenti azioni:

- `appconfig:CreateExtensionAssociation`
- `appconfig:GetConfigurationProfile`
- `appconfig>ListApplications`
- `appconfig>ListConfigurationProfiles`
- `appconfig>ListExtensionAssociations`
- `sts:GetCallerIdentity`

Completa le seguenti attività per creare una policy di autorizzazione IAM e un utente per AWS AppConfig l'integrazione con Jira:

### Attività

- [Attività 1: creare una politica di autorizzazione IAM per l'integrazione AWS AppConfig con Jira](#)

- [Attività 2: creare un utente per l'integrazione AWS AppConfig con Jira](#)

Attività 1: creare una politica di autorizzazione IAM per l'integrazione AWS AppConfig con Jira

Utilizza la seguente procedura per creare una policy di autorizzazione IAM che consenta ad Atlassian Jira di comunicare con AWS AppConfig. Ti consigliamo di creare una nuova policy e di collegarla a un nuovo ruolo IAM. L'aggiunta dell'autorizzazione richiesta a una policy e a un ruolo IAM esistenti è contraria al principio del privilegio minimo e non è consigliata.

Per creare una policy IAM per l'integrazione AWS AppConfig con Jira

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione, scegli Policy e Crea policy.
3. Nella pagina Crea policy, scegli la scheda JSON e sostituisci il contenuto predefinito con la seguente policy. Nella seguente politica, sostituisci *Region account\_IDapplication\_ID*, e *configuration\_profile\_ID* con le informazioni del tuo ambiente di flag delle AWS AppConfig funzionalità.

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "appconfig:CreateExtensionAssociation",  
        "appconfig>ListExtensionAssociations",  
        "appconfig:GetConfigurationProfile"  
      ],  
      "Resource": [  
        "arn:aws:appconfig:us-  
east-1:111122223333:application/application_ID",  
        "arn:aws:appconfig:us-  
east-1:111122223333:application/application_ID/  
        configurationprofile/configuration_profile_ID"  
      ]  
    },  
    {  
      "Effect": "Allow",  
    }  
  ]  
}
```

```

        "Action": [
            "appconfig>ListApplications"
        ],
        "Resource": [
            "arn:aws:appconfig:us-east-1:111122223333:/*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "appconfig>ListConfigurationProfiles"
        ],
        "Resource": [
            "arn:aws:appconfig:us-
east-1:111122223333:application/application_ID"
        ]
    },
    {
        "Effect": "Allow",
        "Action": "sts:GetCallerIdentity",
        "Resource": "*"
    }
]
}

```

4. Scegliere Next: Tags (Successivo: Tag).
5. (Facoltativo) Aggiungere una o più coppie tag-valore chiave per organizzare, monitorare o controllare l'accesso per questa policy, quindi scegliere Next: Review (Successivo: Rivedi).
6. Nella pagina Review policy (Rivedi policy), immetti un nome nella casella Name (Nome), ad esempio  **AppConfigJiraPolicy**, quindi immetti una descrizione facoltativa.
7. Scegli Crea policy.

#### Attività 2: creare un utente per l'integrazione AWS AppConfig con Jira

Usa la seguente procedura per creare un utente per AWS AppConfig l'integrazione con Atlassian Jira. Dopo aver creato l'utente, puoi copiare l'ID della chiave di accesso e la chiave segreta, che specificherai al termine dell'integrazione.

Per creare un utente AWS AppConfig e un'integrazione con Jira

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.

2. Nel riquadro di navigazione, scegli Users (Utenti), quindi scegli Add users (Aggiungi utenti).
3. Nel campo Nome utente, inserisci un nome, ad esempio **AppConfigJiraUser**.
4. Per Selezione il tipo di AWS credenziale, scegli Chiave di accesso - Accesso programmatico.
5. Scegli Successivo: autorizzazioni.
6. Nella pagina Imposta autorizzazioni, scegli Allega direttamente le politiche esistenti. Cerca e seleziona la casella di controllo relativa alla politica in cui hai creato [Attività 1: creare una politica di autorizzazione IAM per l'integrazione AWS AppConfig con Jira](#), quindi scegli Avanti: tag.
7. Nella pagina Aggiungi tag (opzionale), aggiungi una o più coppie tag-chiave-valore per organizzare, tracciare o controllare l'accesso per questo utente. Scegli Prossimo: Rivedi.
8. Nella pagina Revisione, verifica i dettagli dell'utente.
9. Selezionare Create user (Crea utente). Il sistema visualizza l'ID della chiave di accesso e la chiave segreta dell'utente. Scarica il file.csv o copia queste credenziali in una posizione separata. Queste credenziali verranno specificate durante la configurazione dell'integrazione.

## Configurazione dell'applicazione di integrazione AWS AppConfig Jira

Usa la seguente procedura per configurare le opzioni richieste nell'applicazione AWS AppConfig per Jira. Dopo aver completato questa procedura, Jira crea un nuovo problema per ogni feature flag presente nel tuo account Account AWS per quanto specificato. Regione AWS Se apporti modifiche a un feature flag in AWS AppConfig, Jira registra i dettagli nei problemi esistenti.

### Note

Un AWS AppConfig feature flag può includere più attributi di flag a livello di bambino. Jira crea un problema per ogni feature flag di livello principale. Se modifichi un attributo di flag a livello di figlio, puoi visualizzare i dettagli di tale modifica nel numero di Jira per il flag di livello genitore.

## Per configurare l'integrazione

1. Accedi all'[Atlassian Marketplace](#).
2. Digita **AWS AppConfig** nel campo di ricerca e premi Invio.
3. Installa l'applicazione sulla tua istanza Jira.
4. Nella console Atlassian, scegli Gestisci app, quindi scegli AWS AppConfig Jira.

5. Scegli Configura.
6. In Dettagli di configurazione, scegli Jira project, quindi scegli il progetto che desideri associare al tuo feature flag. AWS AppConfig
7. Scegli Regione AWS, quindi scegli la regione in cui si trova il tuo AWS AppConfig feature flag.
8. Nel campo ID applicazione, inserisci il nome dell' AWS AppConfig applicazione che contiene il tuo flag di funzionalità.
9. Nel campo ID del profilo di configurazione, inserite il nome del profilo di AWS AppConfig configurazione per il vostro feature flag.
10. Nei campi ID chiave di accesso e Chiave segreta, inserisci le credenziali che hai copiato. [Attività 2: creare un utente per l'integrazione AWS AppConfig con Jira](#) Facoltativamente, puoi anche specificare un token di sessione.
11. Seleziona Invia.
12. Nella console Atlassian, scegli Progetti, quindi scegli il progetto selezionato per l'integrazione. AWS AppConfig La pagina Problemi mostra un problema per ogni indicatore di funzionalità nel campo specificato Account AWS e. Regione AWS

## Eliminazione dell'applicazione e dei dati AWS AppConfig per Jira

Se non desideri più utilizzare l'integrazione di Jira con i flag di AWS AppConfig funzionalità, puoi eliminare l'applicazione AWS AppConfig per Jira nella console Atlassian. L'eliminazione dell'applicazione di integrazione esegue le seguenti operazioni:

- Elimina l'associazione tra l'istanza di Jira e AWS AppConfig
- Elimina i dettagli dell'istanza Jira da AWS AppConfig

Per eliminare l'applicazione AWS AppConfig for Jira

1. Nella console Atlassian, scegli Gestisci app.
2. Scegli AWS AppConfig per Jira.
3. Scegliere Uninstall (Disinstalla).

# Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig

Per creare un' AWS AppConfig estensione personalizzata, completa le seguenti attività. Ogni attività è descritta più dettagliatamente negli argomenti successivi.

## Note

È possibile visualizzare esempi di AWS AppConfig estensioni personalizzate su GitHub:

- [Estensione di esempio che impedisce le implementazioni con un calendario di blocked day moratoria utilizzando Systems Manager Change Calendar](#)
- [Estensione di esempio che impedisce la divulgazione di segreti nei dati di configurazione utilizzando git-secrets](#)
- [Estensione di esempio che impedisce la fuoriuscita di informazioni di identificazione personale \(PII\) nei dati di configurazione utilizzando Amazon Comprehend](#)

## 1. [Crea una funzione AWS Lambda](#)

Nella maggior parte dei casi d'uso, per creare un'estensione personalizzata, è necessario creare una AWS Lambda funzione per eseguire qualsiasi calcolo ed elaborazione definiti nell'estensione. Un'eccezione a questa regola è [AWS rappresentata dalla creazione di versioni personalizzate delle estensioni di notifica create](#) per aggiungere o rimuovere punti di azione. Per ulteriori dettagli su questa eccezione, consulta [Fase 3: Creare un'estensione personalizzata AWS AppConfig](#).

## 2. [Configura le autorizzazioni per la tua estensione personalizzata](#)

Per configurare le autorizzazioni per l'estensione personalizzata, puoi effettuare una delle seguenti operazioni:

- Crea un ruolo di servizio AWS Identity and Access Management (IAM) che includa le InvokeFunction autorizzazioni.
- Crea una politica delle risorse utilizzando l'azione [AddPermission](#) API Lambda.

Questa procedura dettagliata descrive come creare il ruolo di servizio IAM.

### 3. [Crea un'estensione](#)

Puoi creare un'estensione utilizzando la AWS AppConfig console o richiamando l'azione [CreateExtension](#)API da AWS CLI AWS Strumenti per PowerShell, o dall'SDK. La procedura dettagliata utilizza la console.

### 4. [Crea un'associazione di estensioni](#)

Puoi creare un'associazione di estensioni utilizzando la AWS AppConfig console o richiamando l'azione [CreateExtensionAssociation](#)API da AWS CLI AWS Strumenti per PowerShell, o dall'SDK. La procedura dettagliata utilizza la console.

### 5. Esegui un'azione che richiami l'estensione

Dopo aver creato l'associazione, AWS AppConfig richiama l'estensione quando si verificano i punti d'azione definiti dall'estensione per quella risorsa. Ad esempio, se si associa un'estensione che contiene un'PRE\_CREATE\_HOSTED\_CONFIGURATION\_VERSIONazione, l'estensione viene richiamata ogni volta che si crea una nuova versione di configurazione ospitata.

Gli argomenti di questa sezione descrivono ogni attività coinvolta nella creazione di un'AWS AppConfig estensione personalizzata. Ogni attività viene descritta nel contesto di un caso d'uso in cui un cliente desidera creare un'estensione che esegua automaticamente il backup di una configurazione in un bucket Amazon Simple Storage Service (Amazon S3). L'estensione viene eseguita ogni volta che una configurazione ospitata viene creata (PRE\_CREATE\_HOSTED\_CONFIGURATION\_VERSION) o distribuita (PRE\_START\_DEPLOYMENT).

#### Argomenti

- [Fase 1: Creare una funzione Lambda per un'estensione personalizzata AWS AppConfig](#)
- [Passaggio 2: configura le autorizzazioni per un'estensione personalizzata AWS AppConfig](#)
- [Fase 3: Creare un'estensione personalizzata AWS AppConfig](#)
- [Fase 4: Creare un'associazione di estensioni per un'estensione personalizzata AWS AppConfig](#)

## Fase 1: Creare una funzione Lambda per un'estensione personalizzata AWS AppConfig

Nella maggior parte dei casi d'uso, per creare un'estensione personalizzata, è necessario creare una AWS Lambda funzione per eseguire qualsiasi calcolo ed elaborazione definiti nell'estensione. Questa sezione include il codice di esempio della funzione Lambda per un'estensione personalizzata.

AWS AppConfig. Questa sezione include anche i dettagli di riferimento della richiesta e della risposta del payload. Per informazioni sulla creazione di una funzione Lambda, consulta [Getting started with Lambda](#) nella Developer Guide.AWS Lambda

## Codice di esempio

Il seguente codice di esempio per una funzione Lambda, quando richiamato, esegue automaticamente il backup di una AWS AppConfig configurazione in un bucket Amazon S3. Viene eseguito il backup della configurazione ogni volta che viene creata o distribuita una nuova configurazione. L'esempio utilizza i parametri di estensione in modo che il nome del bucket non debba essere codificato nella funzione Lambda. Utilizzando i parametri di estensione, l'utente può collegare l'estensione a più applicazioni ed eseguire il backup delle configurazioni in diversi bucket. L'esempio di codice include commenti per spiegare ulteriormente la funzione.

### Esempio di funzione Lambda per un'estensione AWS AppConfig

```
from datetime import datetime
import base64
import json

import boto3

def lambda_handler(event, context):
    print(event)

    # Extensions that use the PRE_CREATE_HOSTED_CONFIGURATION_VERSION and
    # PRE_START_DEPLOYMENT
    # action points receive the contents of AWS AppConfig configurations in Lambda
    # event parameters.
    # Configuration contents are received as a base64-encoded string, which the lambda
    # needs to decode
    # in order to get the configuration data as bytes. For other action points, the
    # content
    # of the configuration isn't present, so the code below will fail.
    config_data_bytes = base64.b64decode(event["Content"])

    # You can specify parameters for extensions. The CreateExtension API action lets
    # you define
    # which parameters an extension supports. You supply the values for those
    # parameters when you
```

```
# create an extension association by calling the CreateExtensionAssociation API
action.

# The following code uses a parameter called S3_BUCKET to obtain the value
specified in the

# extension association. You can specify this parameter when you create the
extension

# later in this walkthrough.

extension_association_params = event.get('Parameters', {})
bucket_name = extension_association_params['S3_BUCKET']
write_backup_to_s3(bucket_name, config_data_bytes)

# The PRE_CREATE_HOSTED_CONFIGURATION_VERSION and PRE_START_DEPLOYMENT action
points can

# modify the contents of a configuration. The following code makes a minor change
# for the purposes of a demonstration.

old_config_data_string = config_data_bytes.decode('utf-8')
new_config_data_string = old_config_data_string.replace('hello', 'hello!')
new_config_data_bytes = new_config_data_string.encode('utf-8')

# The lambda initially received the configuration data as a base64-encoded string
# and must return it in the same format.

new_config_data_base64string =
base64.b64encode(new_config_data_bytes).decode('ascii')

return {

    'statusCode': 200,
    # If you want to modify the contents of the configuration, you must include the
    new_contents in the

    # Lambda response. If you don't want to modify the contents, you can omit the
    'Content' field shown here.

    'Content': new_config_data_base64string
}

def write_backup_to_s3(bucket_name, config_data_bytes):
    s3 = boto3.resource('s3')
    new_object = s3.Object(bucket_name,
f"config_backup_{datetime.now().isoformat()}.txt")
    new_object.put(Body=config_data_bytes)
```

Se desideri utilizzare questo esempio durante questa procedura dettagliata, salvalo con il nome **MyS3ConfigurationBackUpExtension** e copia l'Amazon Resource Name (ARN) per la funzione. L'ARN viene specificato quando si crea il ruolo di assunzione AWS Identity and Access Management

(IAM) nella sezione successiva. L'ARN e il nome vengono specificati al momento della creazione dell'estensione.

## Riferimento del payload

Questa sezione include i dettagli di riferimento della richiesta di payload e della risposta per l'utilizzo delle estensioni personalizzate AWS AppConfig .

### Struttura della richiesta

#### AtDeploymentTick

```
{  
  'InvocationId': 'o2xbtm7',  
  'Parameters': {  
    'ParameterOne': 'ValueOne',  
    'ParameterTwo': 'ValueTwo'  
  },  
  'Type': 'OnDeploymentStart',  
  'Application': {  
    'Id': 'abcd123'  
  },  
  'Environment': {  
    'Id': 'efgh456'  
  },  
  'ConfigurationProfile': {  
    'Id': 'ijkl789',  
    'Name': 'ConfigurationName'  
  },  
  'DeploymentNumber': 2,  
  'Description': 'Deployment description',  
  'ConfigurationVersion': '2',  
  'DeploymentState': 'DEPLOYING',  
  'PercentageComplete': '0.0'  
}
```

### Struttura della richiesta

#### PreCreateHostedConfigurationVersion

```
{  
  'InvocationId': 'vlns753', // id for specific invocation  
  'Parameters': {
```

```
    'ParameterOne': 'ValueOne',
    'ParameterTwo': 'ValueTwo'
},
'ContentType': 'text/plain',
'ContentVersion': '2',
'Content': 'SGVsbG8gZWYdGgh', // Base64 encoded content
'Application': {
    'Id': 'abcd123',
    'Name': 'ApplicationName'
},
'ConfigurationProfile': {
    'Id': 'ijkl789',
    'Name': 'ConfigurationName'
},
'Description': '',
'Type': 'PreCreateHostedConfigurationVersion',
'PreviousContent': {
    'ContentType': 'text/plain',
    'ContentVersion': '1',
    'Content': 'SGVsbG8gd29ybGQh'
}
}
```

## PreStartDeployment

```
{
    'InvocationId': '765ahdm',
    'Parameters': {
        'ParameterOne': 'ValueOne',
        'ParameterTwo': 'ValueTwo'
    },
    'ContentType': 'text/plain',
    'ContentVersion': '2',
    'Content': 'SGVsbG8gZWYdGgh',
    'Application': {
        'Id': 'abcd123',
        'Name': 'ApplicationName'
    },
    'Environment': {
        'Id': 'ibpnqlq',
        'Name': 'EnvironmentName'
    },
    'ConfigurationProfile': {
```

```
        'Id': 'ijkl1789',
        'Name': 'ConfigurationName'
    },
    'DeploymentNumber': 2,
    'Description': 'Deployment description',
    'Type': 'PreStartDeployment'
}
```

## Eventi asincroni

OnStartDeployment, OnDeploymentStep, OnDeployment

```
{
    'InvocationId': 'o2xbtm7',
    'Parameters': {
        'ParameterOne': 'ValueOne',
        'ParameterTwo': 'ValueTwo'
    },
    'Type': 'OnDeploymentStart',
    'Application': {
        'Id': 'abcd123'
    },
    'Environment': {
        'Id': 'efgh456'
    },
    'ConfigurationProfile': {
        'Id': 'ijkl1789',
        'Name': 'ConfigurationName'
    },
    'DeploymentNumber': 2,
    'Description': 'Deployment description',
    'ConfigurationVersion': '2'
}
```

## Struttura della risposta

Gli esempi seguenti mostrano cosa restituisce la funzione Lambda in risposta alla richiesta di un'estensione personalizzata AWS AppConfig .

**PRE\_\*** Eventi sincroni: risposta riuscita

Se desideri trasformare il contenuto, usa quanto segue:

```
"Content": "SomeBase64EncodedByteArray"
```

### AT\_\* Eventi sincroni: risposta riuscita

Se desideri controllare le fasi successive di una distribuzione (continuare una distribuzione o ripristinarla), imposta **Directive** e **Description** attributi nella risposta.

```
"Directive": "ROLL_BACK"  
"Description": "Deployment event log description"
```

**Directive** supporta due valori: **CONTINUE** o **ROLL\_BACK**. Usa queste enumerazioni nella tua risposta al payload per controllare i passaggi successivi di una distribuzione.

### Eventi sincroni: risposta riuscita

Se vuoi trasformare il contenuto, usa quanto segue:

```
"Content": "SomeBase64EncodedByteArray"
```

Se non vuoi trasformare il contenuto, non restituisci nulla.

### Eventi asincroni: risposta riuscita

Non restituisci nulla.

### Tutti gli eventi di errore

```
{  
  "Error": "BadRequestError",  
  "Message": "There was malformed stuff in here",  
  "Details": [  
    {"Type": "Malformed",  
     "Name": "S3 pointer",  
     "Reason": "S3 bucket did not exist"}]
```

## Passaggio 2: configura le autorizzazioni per un'estensione personalizzata AWS AppConfig

Utilizza la seguente procedura per creare e configurare un ruolo di servizio AWS Identity and Access Management (IAM) (o assumere un ruolo). AWS AppConfig utilizza questo ruolo per richiamare la funzione Lambda.

Per creare un ruolo di servizio IAM e consentire di AWS AppConfig assumerlo

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione, scegli Ruoli e quindi Crea ruolo.
3. In Seleziona il tipo di entità affidabile, scegli Politica di fiducia personalizzata.
4. Incolla la seguente politica JSON nel campo Politica di attendibilità personalizzata.

JSON

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

Scegli Next (Successivo).

5. Nella pagina Aggiungi autorizzazioni, scegli Crea politica. La pagina Create policy (Crea policy) viene aperta in una nuova scheda.
6. Scegli la scheda JSON, quindi incolla la seguente politica di autorizzazione nell'editor.  
L'lambda:InvokeFunctionazione viene utilizzata per i punti PRE\_\* d'azione.  
L'lambda:InvokeAsyncazione viene utilizzata per i punti ON\_\* azione. Sostituisci **Your Lambda ARN** con l'Amazon Resource Name (ARN) della tua Lambda.

## JSON

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "VisualEditor0",  
            "Effect": "Allow",  
            "Action": [  
                "lambda:InvokeFunction",  
                "lambda:InvokeAsync"  
            ],  
            "Resource": "arn:aws:lambda:us-east-1:111222333:function:Function-  
name"  
        }  
    ]  
}
```

7. Scegli Successivo: Tag.
8. Nella pagina Aggiungi tag (opzionale), aggiungi una o più coppie chiave-valore, quindi scegli Avanti: revisione.
9. Nella pagina Revisione della politica, inserisci un nome e una descrizione, quindi scegli Crea politica.
10. Nella scheda del browser relativa alla politica di attendibilità personalizzata, scegli l'icona Aggiorna, quindi cerca la politica di autorizzazione appena creata.
11. Seleziona la casella di controllo relativa alla politica di autorizzazione, quindi scegli Avanti.
12. Nella pagina Nome, rivedi e crea, inserisci un nome nella casella Nome del ruolo, quindi inserisci una descrizione.
13. Scegliere Create role (Crea ruolo). Il sistema visualizza di nuovo la pagina Ruoli. Scegli Visualizza ruolo nel banner.
14. Copia l'ARN. Specificate questo ARN quando create l'estensione.

## Fase 3: Creare un'estensione personalizzata AWS AppConfig

Un'estensione definisce una o più azioni che esegue durante un AWS AppConfig flusso di lavoro. Ad esempio, l'AWS AppConfig deployment events to Amazon SNS estensione AWS creata

include un'azione per inviare una notifica a un argomento di Amazon SNS. Ogni azione viene richiamata quando interagisci con AWS AppConfig o quando AWS AppConfig esegui un processo per tuo conto. Questi sono chiamati punti d'azione. AWS AppConfig le estensioni supportano i seguenti punti di azione:

punti d'azione PRE\_\*: le azioni di estensione configurate sui punti di PRE\_\* azione vengono applicate dopo la convalida della richiesta, ma prima dell' AWS AppConfig esecuzione dell'attività corrispondente al nome del punto di azione. Queste invocazioni di azioni vengono elaborate contemporaneamente a una richiesta. Se viene effettuata più di una richiesta, le chiamate alle azioni vengono eseguite in sequenza. Si noti inoltre che i punti di PRE\_\* azione ricevono e possono modificare il contenuto di una configurazione. PRE\_\*i punti di azione possono anche rispondere a un errore e impedire che si verifichi un'azione.

- PRE\_CREATE\_HOSTED\_CONFIGURATION\_VERSION
- PRE\_START\_DEPLOYMENT

Punti di azione ON\_\*: un'estensione può anche essere eseguita in parallelo a un AWS AppConfig flusso di lavoro utilizzando un punto ON\_\* azione. ON\_\*i punti di azione vengono richiamati in modo asincrono. ON\_\*i punti di azione non ricevono il contenuto di una configurazione. Se un'estensione riscontra un errore durante un punto di ON\_\* azione, il servizio ignora l'errore e continua il flusso di lavoro.

- ON\_DEPLOYMENT\_START
- ON\_DEPLOYMENT\_STEP
- ON\_DEPLOYMENT\_BAKING
- ON\_DEPLOYMENT\_COMPLETE
- ON\_DEPLOYMENT\_ROLLED\_BACK

Punti di azione AT\_\*: le azioni di estensione configurate sui punti di AT\_\* azione vengono richiamate in modo sincrono e parallelo a un flusso di lavoro. AWS AppConfig Se un'estensione riscontra un errore durante un punto di AT\_\* azione, il servizio interrompe il flusso di lavoro e ripristina la distribuzione.

- AT\_DEPLOYMENT\_TICK

### Note

Il punto di AT\_DEPLOYMENT\_TICK azione supporta l'integrazione del monitoraggio di terze parti. AT\_DEPLOYMENT\_TICK viene richiamato durante l'orchestrazione dell'elaborazione della configurazione, della distribuzione. Se utilizzi una soluzione di monitoraggio di terze parti (ad esempio Datadog), puoi creare un' AWS AppConfig estensione che verifica la presenza di allarmi nel punto di AT\_DEPLOYMENT\_TICK azione e, come barriera di sicurezza, ripristina l'implementazione se ha attivato un allarme. [Per visualizzare un esempio di codice di un' AWS AppConfig estensione che utilizza il punto di AT\\_DEPLOYMENT\\_TICK azione per l'integrazione con Datadog, consulta aws-samples/-for-datadog on. aws-appconfig-tick-extn GitHub](#)

## Estensione di esempio

L'estensione di esempio seguente definisce un'azione che richiama il punto di PRE\_CREATE\_HOSTED\_CONFIGURATION\_VERSION azione. Nel Uri campo, l'azione specifica l'Amazon Resource Name (ARN) della funzione MyS3ConfigurationBackUpExtension Lambda creata in precedenza in questa procedura dettagliata. L'azione specifica anche l'ARN di assunzione del ruolo AWS Identity and Access Management (IAM) creato in precedenza in questa procedura dettagliata.

## Estensione di esempio AWS AppConfig

```
{  
  "Name": "MySampleExtension",  
  "Description": "A sample extension that backs up configurations to an S3 bucket.",  
  "Actions": {  
    "PRE_CREATE_HOSTED_CONFIGURATION_VERSION": [  
      {  
        "Name": "PreCreateHostedConfigVersionActionForS3Backup",  
        "Uri": "arn:aws:lambda:aws-  
region:111122223333:function:MyS3ConfigurationBackUpExtension",  
        "RoleArn": "arn:aws:iam::111122223333:role/ExtensionsTestRole"  
      }  
    ]  
  },  
  "Parameters" : {  
    "S3_BUCKET": {  
      "Required": false  
    }  
  }  
}
```

```
    }  
}  
}
```

### Note

Per visualizzare la sintassi della richiesta e le descrizioni dei campi durante la creazione di un'estensione, consulta l'[CreateExtension](#) argomento nell'AWS AppConfig API Reference.

## Per creare un'estensione (console)

1. Apri la AWS Systems Manager console su <https://console.aws.amazon.com/systems-manager/appconfig/>.
2. Nel pannello di navigazione, scegli AWS AppConfig.
3. Nella scheda Estensioni, scegli Crea estensione.
4. Per Nome estensione, inserisci un nome univoco. Ai fini di questa procedura dettagliata, immettere. **MyS3ConfigurationBackUpExtension** Facoltativamente, inserire una descrizione.
5. Nella sezione Azioni, scegli Aggiungi nuova azione.
6. Per Nome operazione, inserire un nome univoco. Ai fini di questa procedura dettagliata, inserisci. **PreCreateHostedConfigVersionActionForS3Backup** Questo nome descrive il punto d'azione utilizzato dall'azione e lo scopo dell'estensione.
7. Nell'elenco dei punti d'azione, scegliete **PRE\_CREATE\_HOSTED\_CONFIGURATION\_VERSION**.
8. Per Uri, scegli la funzione Lambda, quindi scegli la funzione nell'elenco delle funzioni Lambda. Se non vedi la tua funzione, verifica di trovarla nella stessa posizione in Regione AWS cui l'hai creata.
9. Per IAM Role, scegli il ruolo che hai creato in precedenza in questa procedura dettagliata.
10. Nella sezione Parametri di estensione (opzionale), scegli Aggiungi nuovo parametro.
11. Per Nome del parametro, inserisci un nome. Ai fini di questa procedura dettagliata, immettere. **S3\_BUCKET**
12. Ripetete i passaggi da 5 a 11 per creare una seconda azione per il punto d'azione. **PRE\_START\_DEPLOYMENT**
13. Scegliete Crea estensione.

## Personalizzazione delle estensioni di AWS notifica create

Non è necessario creare una Lambda o un'estensione per utilizzare le estensioni di notifica [AWS create](#). Puoi semplicemente creare un'associazione di estensioni e quindi eseguire un'operazione che richiama uno dei punti di azione supportati. Per impostazione predefinita, le AWS estensioni di notifica create supportano i seguenti punti di azione:

- ON\_DEPLOYMENT\_START
- ON\_DEPLOYMENT\_COMPLETE
- ON\_DEPLOYMENT\_ROLLED\_BACK

Se crei versioni personalizzate dell'AWS AppConfig deployment events to Amazon SNS estensione e delle AWS AppConfig deployment events to Amazon SQS estensioni, puoi specificare i punti di azione per i quali desideri ricevere notifiche.

### Note

L'AWS AppConfig deployment events to EventBridge estensione non supporta i punti di PRE\_\* azione. Puoi creare una versione personalizzata se desideri rimuovere alcuni dei punti di azione predefiniti assegnati alla AWS versione creata.

Non è necessario creare una funzione Lambda se si creano versioni personalizzate delle estensioni di notifica AWS create. Devi solo specificare un Amazon Resource Name (ARN) nel Uri campo per la nuova versione dell'estensione.

- Per un'estensione di EventBridge notifica personalizzata, inserisci l'ARN degli eventi EventBridge predefiniti nel Uri campo.
- Per un'estensione di notifica Amazon SNS personalizzata, inserisci l'ARN di un argomento Amazon SNS nel campo. Uri
- Per un'estensione di notifica Amazon SQS personalizzata, inserisci l'ARN di una coda di messaggi Amazon SQS nel campo. Uri

## Fase 4: Creare un'associazione di estensioni per un'estensione personalizzata AWS AppConfig

Per creare un'estensione o configurare un'estensione AWS creata, definisci i punti di azione che richiamano un'estensione quando viene utilizzata una AWS AppConfig risorsa specifica. Ad esempio, puoi scegliere di eseguire l'AWS AppConfig deployment events to Amazon SNS estensione e ricevere notifiche su un argomento di Amazon SNS ogni volta che viene avviata una distribuzione di configurazione per un'applicazione specifica. La definizione dei punti di azione che richiamano un'estensione per una AWS AppConfig risorsa specifica viene chiamata associazione di estensioni. Un'associazione di estensione è una relazione specifica tra un'estensione e una AWS AppConfig risorsa, ad esempio un'applicazione o un profilo di configurazione.

Una singola AWS AppConfig applicazione può includere più ambienti e profili di configurazione. Se si associa un'estensione a un'applicazione o a un ambiente, AWS AppConfig richiama l'estensione per tutti i flussi di lavoro relativi all'applicazione o alle risorse dell'ambiente, se applicabile.

Ad esempio, supponiamo di avere un'AWS AppConfig applicazione chiamata MobileApps che include un profilo di configurazione chiamato. AccessList Supponiamo che l' MobileApps applicazione includa ambienti beta, di integrazione e di produzione. Crei un'associazione di estensione per l' AWS estensione di notifica Amazon SNS creata e associa l'estensione all' MobileApps applicazione. L'estensione di notifica di Amazon SNS viene richiamata ogni volta che la configurazione viene distribuita per l'applicazione in uno dei tre ambienti.

Utilizza le seguenti procedure per creare un'associazione di AWS AppConfig estensioni utilizzando la console. AWS AppConfig

Per creare un'associazione di estensioni (console)

1. Apri la AWS Systems Manager console su <https://console.aws.amazon.com/systems-manager/appconfig/>.
2. Nel pannello di navigazione, scegli AWS AppConfig.
3. Nella scheda Estensioni, scegli un pulsante di opzione per un'estensione, quindi scegli Aggiungi alla risorsa. Ai fini di questa procedura dettagliata, scegli MyS3. ConfigurationBackUpExtension
4. Nella sezione Dettagli della risorsa di estensione, per Tipo di risorsa, scegli un tipo di risorsa. AWS AppConfig A seconda della risorsa scelta, AWS AppConfig ti chiede di scegliere altre risorse. Ai fini di questa procedura dettagliata, scegli Applicazione.
5. Scegli un'applicazione nell'elenco.

6. Nella sezione Parametri, verifica che S3\_BUCKET sia elencato nel campo Chiave. Nel campo Valore, incolla l'ARN delle estensioni Lambda. Ad esempio: `arn:aws:lambda:aws-region:111122223333:function:MyS3ConfigurationBackUpExtension`.
7. Scegli Crea associazione alla risorsa.

Dopo aver creato l'associazione, puoi richiamare

`IMyS3ConfigurationBackUpExtension` estensione creando un nuovo profilo di configurazione che lo specifichi `hosted`. `SourceUri` Come parte del flusso di lavoro per creare la nuova configurazione, AWS AppConfig incontra il punto d'azione.

`PRE_CREATE_HOSTED_CONFIGURATION_VERSION` L'incontro di questo punto di azione richiama `IMyS3ConfigurationBackUpExtension` estensione, che esegue automaticamente il backup della configurazione appena creata nel bucket S3 specificato nella sezione dell'associazione di estensione.

Parameter

# Utilizzo di esempi di codice per eseguire AWS AppConfig attività comuni

Questa sezione include esempi di codice per eseguire azioni comuni AWS AppConfig a livello di codice. Ti consigliamo di utilizzare questi esempi con [Java](#), [Python](#) e di [JavaScript](#) SDKs eseguire le azioni in un ambiente di test. Questa sezione include un esempio di codice per ripulire l'ambiente di test al termine.

## Argomenti

- [Creazione o aggiornamento di una configurazione in formato libero archiviata nell'archivio di configurazione ospitato](#)
- [Creazione di un profilo di configurazione per un segreto archiviato in Secrets Manager](#)
- [Implementazione di un profilo di configurazione](#)
- [Utilizzo di AWS AppConfig Agent per leggere un profilo di configurazione in formato libero](#)
- [Utilizzo di AWS AppConfig Agent per leggere un indicatore di funzionalità specifico](#)
- [Utilizzo di AWS AppConfig Agent per recuperare un flag di funzionalità con varianti](#)
- [Utilizzo dell'azione GetLatestConfiguration API per leggere un profilo di configurazione in formato libero](#)
- [Pulizia dell'ambiente](#)

## Creazione o aggiornamento di una configurazione in formato libero archiviata nell'archivio di configurazione ospitato

Ciascuno dei seguenti esempi include commenti sulle azioni eseguite dal codice. Gli esempi di questa sezione richiamano quanto segue APIs:

- [CreateApplication](#)
- [CreateConfigurationProfile](#)
- [CreateHostedConfigurationVersion](#)

## Java

```
public CreateHostedConfigurationVersionResponse createHostedConfigVersion() {
```

```
AppConfigClient appconfig = AppConfigClient.create();

    // Create an application
    CreateApplicationResponse app = appconfig.createApplication(req ->
req.name("MyDemoApp"));

    // Create a hosted, freeform configuration profile
    CreateConfigurationProfileResponse configProfile =
appconfig.createConfigurationProfile(req -> req
        .applicationId(app.id())
        .name("MyConfigProfile")
        .locationUri("hosted")
        .type("AWS.Freeform"));

    // Create a hosted configuration version
    CreateHostedConfigurationVersionResponse hcv =
appconfig.createHostedConfigurationVersion(req -> req
        .applicationId(app.id())
        .configurationProfileId(configProfile.id())
        .contentType("text/plain; charset=utf-8")
        .content(SdkBytes.fromUtf8String("my config data")));

    return hcv;
}
```

## Python

```
import boto3

appconfig = boto3.client('appconfig')

# create an application
application = appconfig.create_application(Name='MyDemoApp')

# create a hosted, freeform configuration profile
config_profile = appconfig.create_configuration_profile(
    ApplicationId=application['Id'],
    Name='MyConfigProfile',
    LocationUri='hosted',
    Type='AWS.Freeform')

# create a hosted configuration version
hcv = appconfig.create_hosted_configuration_version(
```

```
ApplicationId=application['Id'],
ConfigurationProfileId=config_profile['Id'],
Content=b'my config data',
ContentType='text/plain')
```

## JavaScript

```
import {
  AppConfigClient,
  CreateApplicationCommand,
  CreateConfigurationProfileCommand,
  CreateHostedConfigurationVersionCommand,
} from "@aws-sdk/client-appconfig";

const appconfig = new AppConfigClient();

// create an application
const application = await appconfig.send(
  new CreateApplicationCommand({ Name: "MyDemoApp" })
);

// create a hosted, freeform configuration profile
const profile = await appconfig.send(
  new CreateConfigurationProfileCommand({
    ApplicationId: application.Id,
    Name: "MyConfigProfile",
    LocationUri: "hosted",
    Type: "AWS.Freeform",
  })
);

// create a hosted configuration version
await appconfig.send(
  new CreateHostedConfigurationVersionCommand({
    ApplicationId: application.Id,
    ConfigurationProfileId: profile.Id,
    ContentType: "text/plain",
    Content: "my config data",
  })
);
```

# Creazione di un profilo di configurazione per un segreto archiviato in Secrets Manager

Ciascuno dei seguenti esempi include commenti sulle azioni eseguite dal codice. Gli esempi di questa sezione richiamano quanto segue APIs:

- [CreateApplication](#)
- [CreateConfigurationProfile](#)

## Java

```
private void createSecretsManagerConfigProfile() {
    AppConfigClient appconfig = AppConfigClient.create();

    // Create an application
    CreateApplicationResponse app = appconfig.createApplication(req ->
        req.name("MyDemoApp"));

    // Create a configuration profile for Secrets Manager Secret
    CreateConfigurationProfileResponse configProfile =
        appconfig.createConfigurationProfile(req -> req
            .applicationId(app.id())
            .name("MyConfigProfile")
            .locationUri("secretsmanager://MySecret")
            .retrievalRoleArn("arn:aws:iam::000000000000:role/
RoleTrustedByAppConfigThatCanRetrieveSecret")
            .type("AWS.Freeform"));
}
```

## Python

```
import boto3

appconfig = boto3.client('appconfig')

# create an application
application = appconfig.create_application(Name='MyDemoApp')

# create a configuration profile for Secrets Manager Secret
config_profile = appconfig.create_configuration_profile(
```

```
ApplicationId=application['Id'],
Name='MyConfigProfile',
LocationUri='secretsmanager://MySecret',
RetrievalRoleArn='arn:aws:iam::000000000000:role/
RoleTrustedBy AppConfigThatCanRetrieveSecret',
Type='AWS.Freeform')
```

## JavaScript

```
import {
  AppConfigClient,
  CreateConfigurationProfileCommand,
} from "@aws-sdk/client-appconfig";

const appconfig = new AppConfigClient();

// create an application
const application = await appconfig.send(
  new CreateApplicationCommand({ Name: "MyDemoApp" })
);

// create a configuration profile for Secrets Manager Secret
await appconfig.send(
  new CreateConfigurationProfileCommand({
    ApplicationId: application.Id,
    Name: "MyConfigProfile",
    LocationUri: "secretsmanager://MySecret",
    RetrievalRoleArn: "arn:aws:iam::000000000000:role/
RoleTrustedBy AppConfigThatCanRetrieveSecret",
    Type: "AWS.Freeform",
  })
);
```

## Implementazione di un profilo di configurazione

Ciascuno dei seguenti esempi include commenti sulle azioni eseguite dal codice. Gli esempi di questa sezione richiamano quanto segue APIs:

- [CreateApplication](#)
- [CreateConfigurationProfile](#)
- [CreateHostedConfigurationVersion](#)

- [CreateEnvironment](#)
- [StartDeployment](#)
- [GetDeployment](#)

## Java

```
private void createDeployment() throws InterruptedException {
    AppConfigClient appconfig = AppConfigClient.create();

    // Create an application
    CreateApplicationResponse app = appconfig.createApplication(req ->
req.name("MyDemoApp"));

    // Create a hosted, freeform configuration profile
    CreateConfigurationProfileResponse configProfile =
appconfig.createConfigurationProfile(req -> req
        .applicationId(app.id())
        .name("MyConfigProfile")
        .locationUri("hosted")
        .type("AWS.Freeform"));

    // Create a hosted configuration version
    CreateHostedConfigurationVersionResponse hcv =
appconfig.createHostedConfigurationVersion(req -> req
        .applicationId(app.id())
        .configurationProfileId(configProfile.id())
        .contentType("text/plain; charset=utf-8")
        .content(SdkBytes.fromUtf8String("my config data")));

    // Create an environment
    CreateEnvironmentResponse env = appconfig.createEnvironment(req -> req
        .applicationId(app.id())
        .name("Beta")
        // If you have CloudWatch alarms that monitor the health of your
        // service, you can add them here and they
        // will trigger a rollback if they fire during an AppConfig deployment
        //.monitors(Monitor.builder().alarmArn("arn:aws:cloudwatch:us-
east-1:520900602629:alarm:MyAlarm")
        //
        .alarmRoleArn("arn:aws:iam::520900602629:role/MyAppConfigAlarmRole").build())
    );
}
```

```
// Start a deployment
StartDeploymentResponse deploymentResponse = appconfig.startDeployment(req -
> req
    .applicationId(app.id())
    .configurationProfileId(configProfile.id())
    .environmentId(env.id())
    .configurationVersion(hcv.versionNumber().toString())
    .deploymentStrategyId("AppConfig.Linear50PercentEvery30Seconds")
);

// Wait for deployment to complete
List<DeploymentState> nonFinalDeploymentStates = Arrays.asList(
    DeploymentState.DEPLOYING,
    DeploymentState.BAKING,
    DeploymentState.ROLLING_BACK,
    DeploymentState.VALIDATING);
GetDeploymentRequest getDeploymentRequest =
GetDeploymentRequest.builder().applicationId(app.id())

.environmentId(env.id())

.deploymentNumber(deploymentResponse.deploymentNumber()).build();
GetDeploymentResponse deployment =
appconfig.getDeployment(getDeploymentRequest);
while (nonFinalDeploymentStates.contains(deployment.state())) {
    System.out.println("Waiting for deployment to complete: " + deployment);
    Thread.sleep(1000L);
    deployment = appconfig.getDeployment(getDeploymentRequest);
}

System.out.println("Deployment complete: " + deployment);
}
```

## Python

```
import boto3

appconfig = boto3.client('appconfig')

# create an application
application = appconfig.create_application(Name='MyDemoApp')
```

```
# create an environment
environment = appconfig.create_environment(
    ApplicationId=application['Id'],
    Name='MyEnvironment')

# create a configuration profile
config_profile = appconfig.create_configuration_profile(
    ApplicationId=application['Id'],
    Name='MyConfigProfile',
    LocationUri='hosted',
    Type='AWS.Freeform')

# create a hosted configuration version
hcv = appconfig.create_hosted_configuration_version(
    ApplicationId=application['Id'],
    ConfigurationProfileId=config_profile['Id'],
    Content=b'my config data',
    ContentType='text/plain')

# start a deployment
deployment = appconfig.start_deployment(
    ApplicationId=application['Id'],
    EnvironmentId=environment['Id'],
    ConfigurationProfileId=config_profile['Id'],
    ConfigurationVersion=str(hcv['VersionNumber']),
    DeploymentStrategyId='AppConfig.Linear20PercentEvery6Minutes')
```

## JavaScript

```
import {
  AppConfigClient,
  CreateApplicationCommand,
  CreateEnvironmentCommand,
  CreateConfigurationProfileCommand,
  CreateHostedConfigurationVersionCommand,
  StartDeploymentCommand,
} from "@aws-sdk/client-appconfig";

const appconfig = new AppConfigClient();

// create an application
const application = await appconfig.send(
  new CreateApplicationCommand({ Name: "MyDemoApp" })
```

```
);

// create an environment
const environment = await appconfig.send(
  new CreateEnvironmentCommand({
    ApplicationId: application.Id,
    Name: "MyEnvironment",
  })
);

// create a configuration profile
const config_profile = await appconfig.send(
  new CreateConfigurationProfileCommand({
    ApplicationId: application.Id,
    Name: "MyConfigProfile",
    LocationUri: "hosted",
    Type: "AWS.Freeform",
  })
);

// create a hosted configuration version
const hcv = await appconfig.send(
  new CreateHostedConfigurationVersionCommand({
    ApplicationId: application.Id,
    ConfigurationProfileId: config_profile.Id,
    Content: "my config data",
    ContentType: "text/plain",
  })
);

// start a deployment
await appconfig.send(
  new StartDeploymentCommand({
    ApplicationId: application.Id,
    EnvironmentId: environment.Id,
    ConfigurationProfileId: config_profile.Id,
    ConfigurationVersion: hcv.VersionNumber.toString(),
    DeploymentStrategyId: "AppConfig.Linear20PercentEvery6Minutes",
  })
);
```

# Utilizzo di AWS AppConfig Agent per leggere un profilo di configurazione in formato libero

Ciascuno dei seguenti esempi include commenti sulle azioni eseguite dal codice.

## Java

```
public void retrieveConfigFromAgent() throws Exception {
    /*
    In this sample, we will retrieve configuration data from the AWS AppConfig
    Agent.
    The agent is a sidecar process that handles retrieving configuration data
    from AppConfig
    for you in a way that implements best practices like configuration caching.

    For more information about the agent, see How to use AWS AppConfig Agent
    */
    // The agent runs a local HTTP server that serves configuration data
    // Make a GET request to the agent's local server to retrieve the
    configuration data
    URL url = new URL("http://localhost:2772/applications/MyDemoApp/
environments/Beta/configurations/MyConfigProfile");
    HttpURLConnection con = (HttpURLConnection) url.openConnection();
    con.setRequestMethod("GET");
    StringBuilder content;
    try (BufferedReader in = new BufferedReader(new
InputStreamReader(con.getInputStream()))) {
        content = new StringBuilder();
        int ch;
        while ((ch = in.read()) != -1) {
            content.append((char) ch);
        }
    }
    con.disconnect();
    System.out.println("Configuration from agent via HTTP: " + content);
}
```

## Python

```
# in this sample, we will retrieve configuration data from the AWS AppConfig Agent.
```

```
# the agent is a sidecar process that handles retrieving configuration data from AWS
# AppConfig
# for you in a way that implements best practices like configuration caching.
#
# for more information about the agent, see
# How to use AWS AppConfig Agent
#
# import requests

application_name = 'MyDemoApp'
environment_name = 'MyEnvironment'
config_profile_name = 'MyConfigProfile'

# the agent runs a local HTTP server that serves configuration data
# make a GET request to the agent's local server to retrieve the configuration data
response = requests.get(f"http://localhost:2772/applications/{application_name}/
environments/{environment_name}/configurations/{config_profile_name}")
config = response.content
```

## JavaScript

```
// in this sample, we will retrieve configuration data from the AWS AppConfig Agent.
// the agent is a sidecar process that handles retrieving configuration data from
// AppConfig
// for you in a way that implements best practices like configuration caching.

// for more information about the agent, see
// How to use AWS AppConfig Agent

const application_name = "MyDemoApp";
const environment_name = "MyEnvironment";
const config_profile_name = "MyConfigProfile";

// the agent runs a local HTTP server that serves configuration data
// make a GET request to the agent's local server to retrieve the configuration data
const url = `http://localhost:2772/applications/${application_name}/environments/
${environment_name}/configurations/${config_profile_name}`;
const response = await fetch(url);
const config = await response.text(); // (use `await response.json()` if your config
is json)
```

# Utilizzo di AWS AppConfig Agent per leggere un indicatore di funzionalità specifico

Ciascuno dei seguenti esempi include commenti sulle azioni eseguite dal codice.

## Java

```
public void retrieveSingleFlagFromAgent() throws Exception {
    /*
     You can retrieve a single flag's data from the agent by providing the
     "flag" query string parameter.
     Note: the configuration's type must be AWS.AppConfig.FeatureFlags
    */

    URL url = new URL("http://localhost:2772/applications/MyDemoApp/
environments/Beta/configurations/MyFlagsProfile?flag=myFlagKey");
    HttpURLConnection con = (HttpURLConnection) url.openConnection();
    con.setRequestMethod("GET");
    StringBuilder content;
    try (BufferedReader in = new BufferedReader(new
InputStreamReader(con.getInputStream()))) {
        content = new StringBuilder();
        int ch;
        while ((ch = in.read()) != -1) {
            content.append((char) ch);
        }
    }
    con.disconnect();
    System.out.println("MyFlagName from agent: " + content);
}
```

## Python

```
import requests

application_name = 'MyDemoApp'
environment_name = 'MyEnvironment'
config_profile_name = 'MyConfigProfile'
flag_key = 'MyFlag'

# retrieve a single flag's data by providing the "flag" query string parameter
# note: the configuration's type must be AWS.AppConfig.FeatureFlags
```

```
response = requests.get(f"http://localhost:2772/applications/{application_name}/environments/{environment_name}/configurations/{config_profile_name}?flag={flag_key}")
config = response.content
```

## JavaScript

```
const application_name = "MyDemoApp";
const environment_name = "MyEnvironment";
const config_profile_name = "MyConfigProfile";
const flag_name = "MyFlag";

// retrieve a single flag's data by providing the "flag" query string parameter
// note: the configuration's type must be AWS.AppConfig.FeatureFlags
const url = `http://localhost:2772/applications/${application_name}/environments/${environment_name}/configurations/${config_profile_name}?flag=${flag_name}`;
const response = await fetch(url);
const flag = await response.json(); // { "enabled": true/false }
```

## Utilizzo di AWS AppConfig Agent per recuperare un flag di funzionalità con varianti

Ciascuno dei seguenti esempi include commenti sulle azioni eseguite dal codice.

### Java

```
public static void retrieveConfigFromAgentWithVariants() throws Exception {
    /*
    This sample retrieves feature flag configuration data
    containing variants from AWS AppConfig Agent.

    For more information about the agent, see How to use AWS AppConfig Agent
    */
    // Make a GET request to the agent's local server to retrieve the configuration
    // data
    URL url = new URL("http://localhost:2772/applications/MyDemoApp/environments/
    Beta/configurations/MyConfigProfile");
    HttpURLConnection con = (HttpURLConnection) url.openConnection();

    // Provide context in the 'Context' header
```

```
// In the header value, use '=' to separate context key from context value
// Note: Multiple context values may be passed either across
// multiple headers or as comma-separated values in a single header
con.setRequestProperty("Context", "country=US");

StringBuilder content;
try (BufferedReader in = new BufferedReader(new
InputStreamReader(con.getInputStream()))) {
    content = new StringBuilder();
    int ch;
    while ((ch = in.read()) != -1) {
        content.append((char) ch);
    }
}
con.disconnect();
System.out.println("Configuration from agent via HTTP: " + content);
}
```

## Python

```
# This sample retrieve features flag configuration data
# containing variants from AWS AppConfig Agent.

# For more information about the agent, see How to use AWS AppConfig Agent

import requests

application_name = 'MyDemoApp'
environment_name = 'Beta'
configuration_profile_name = 'MyConfigProfile'

# make a GET request to the agent's local server to retrieve the configuration data
response = requests.get(f"http://localhost:2772/applications/{application_name}/
environments/{environment_name}/configurations/{configuration_profile_name}",
                        headers = {
                            "Context": "country=US" # Provide context in the
                            'Context' header
                                            # In the header value, use '='
                            to separate context key from context value
                                            # Note: Multiple context values
                            may be passed either across
                                            # multiple headers or as comma-
                            separated values in a single header
```

```
        }  
    )  
    print("Configuration from agent via HTTP: ", response.json())
```

## JavaScript

```
// This sample retrieves feature flag configuration data  
// containing variants from AWS AppConfig Agent.  
  
// For more information about the agent, see How to use AWS AppConfig Agent  
  
const application_name = "MyDemoApp";  
const environment_name = "Beta";  
const configuration_profile_name = "MyConfigProfile";  
  
const url = `http://localhost:2772/applications/${application_name}/environments/  
${environment_name}/configurations/${configuration_profile_name}`;  
  
// make a GET request to the agent's local server to retrieve the configuration data  
const response = await fetch(url, {  
    method: 'GET',  
    headers: {  
        'Context': 'country=US' // Provide context in the 'Context' header  
                    // In the header value, use '=' to separate context  
key from context value  
                    // Note: Multiple context values may be passed  
either across  
                    // multiple headers or as comma-separated values in  
a single header  
    }  
});  
  
const config = await response.json();  
console.log("Configuration from agent via HTTP: ", config);
```

## Utilizzo dell'azione GetLatestConfiguration API per leggere un profilo di configurazione in formato libero

Ciascuno dei seguenti esempi include commenti sulle azioni eseguite dal codice. Gli esempi di questa sezione richiamano quanto segue APIs:

- [GetLatestConfiguration](#)
- [StartConfigurationSession](#)

## Java

```
/*
The example below uses two AWS AppConfig Data APIs: StartConfigurationSession and
GetLatestConfiguration.
For more information about these APIs, see AWS AppConfig Data.

This class is meant to be used as a singleton to retrieve the latest configuration
data from AWS AppConfig.
This class maintains a cache of the latest configuration data in addition to the
configuration token to be
passed to the next GetLatestConfiguration API call.
*/
public class AppConfigApiRetriever {

    /*
     * Set of AppConfig invalid parameter problems that require restarting the
     configuration session.
     * If the GetLatestConfiguration API call fails with any of these problems (e.g.
     token is EXPIRED or CORRUPTED),
     * we need to call StartConfigurationSession again to obtain a new configuration
     token before retrying.
     */
    private final Set<InvalidParameterProblem> SESSION_RESTART_REQUIRED =
        Stream.of(InvalidParameterProblem.EXPIRED,
        InvalidParameterProblem.CORRUPTED)
            .collect(Collectors.toSet());

    /* AWS AppConfig Data SDK client used to interact with the AWS AppConfig Data
    service.
    */
    private final AppConfigDataClient appConfigData;

    /*
     The configuration token to be passed to the next GetLatestConfiguration API
     call.
     */
    private String configurationToken;
```

```
/*
The cached configuration data to be returned when there is no new configuration
data available.
*/
private SdkBytes configuration;

public AppConfigApiRetriever() {
    this.appConfigData = AppConfigDataClient.create();
}

/*
Returns the latest configuration data stored in AWS AppConfig.
*/
public SdkBytes getConfig() {
    /*
    If there is no configuration token yet, get one by starting a new session
    with the StartConfigurationSession API.

    Note that this API does not return configuration data. Rather, it returns an
    initial configuration token that is
    subsequently passed to the GetLatestConfiguration API.
    */
    if (this.configurationToken == null) {
        startNewSession();
    }

    GetLatestConfigurationResponse response = null;

    try {
        /*
        Retrieve the configuration from the GetLatestConfiguration API,
        providing the current configuration token.

        If this caller does not yet have the latest configuration (e.g. this is
        the first call to GetLatestConfiguration
        or new configuration data has been deployed since the first call), the
        latest configuration data will be returned.

        Otherwise, the GetLatestConfiguration API will not return any data since
        the caller already has the latest.
        */
        response = appConfigData.getLatestConfiguration(
            GetLatestConfigurationRequest.builder()
                .configurationToken(this.configurationToken)
                .build());
    }
}
```

```
        } catch (ResourceNotFoundException e) {
            // Handle resource not found by refreshing the session
            System.err.println("Resource not found - refreshing session and
retrying...");

            startNewSession();
            response = appConfigData.getLatestConfiguration(
                GetLatestConfigurationRequest.builder()

                .configurationToken(this.configurationToken)
                .build());

        } catch (BadRequestException e) {
            // Handle expired or corrupted token by refreshing the session
            boolean needsNewSession = Optional.ofNullable(e.details())
                .map(details ->
details.invalidParameters()
                .values()
                .stream()
                .anyMatch(val ->
> SESSION_RESTART_REQUIRED.contains(val.problem())))
                .orElse(false);

            if (needsNewSession) {
                System.err.println("Configuration token expired or corrupted -
refreshing session and retrying...");

                startNewSession();
                response = appConfigData.getLatestConfiguration(
                    GetLatestConfigurationRequest.builder()

                    .configurationToken(this.configurationToken)
                    .build());

            } else {
                throw e; // rethrow if it's another kind of bad request
            }
        }

        if (response == null) {
            // Should not happen, but return cached config if no response
            return this.configuration;
        }

        /*
         Save the returned configuration token so that it can be passed to the next
         GetLatestConfiguration API call.

         Warning: Not persisting this token for use in the next
         GetLatestConfiguration API call may result in higher
    
```

```
        than expected usage costs.
    */
    this.configurationToken = response.nextPollConfigurationToken();

    /*
     If the GetLatestConfiguration API returned configuration data, update the
     cached configuration with the returned data.
     Otherwise, assume the configuration has not changed, and return the cached
     configuration.
    */
    SdkBytes configFromApi = response.configuration();
    if (configFromApi != null && configFromApi.asByteArray().length != 0) {
        this.configuration = configFromApi;
        System.out.println("Configuration contents have changed since the last
GetLatestConfiguration call, new contents = "
                           + this.configuration.asUtf8String());
    } else {
        System.out.println("GetLatestConfiguration returned an empty response
because we already have the latest configuration");
    }

    return this.configuration;
}

/*
 Starts a new session with AppConfig and retrieves an initial configuration
token.
*/
private void startNewSession() {
    StartConfigurationSessionResponse session =
appConfigData.startConfigurationSession(req -> req
        .applicationIdentifier("MyDemoApp")
        .configurationProfileIdentifier("MyConfig")
        .environmentIdentifier("Beta"));
    this.configurationToken = session.initialConfigurationToken();
}
}
```

## Python

```
# The example below uses two AWS AppConfig Data APIs: StartConfigurationSession and
GetLatestConfiguration.
# For more information about these APIs, see AWS AppConfig Data.
```

```
#  
# This class is meant to be used as a singleton to retrieve the latest configuration  
# data from AWS AppConfig.  
# This class maintains a cache of the latest configuration data in addition to the  
# configuration token to be  
# passed to the next GetLatestConfiguration API call.  
class AppConfigApiRetriever:  
    # Set of AppConfig invalid parameter problems that require restarting the  
    # configuration session.  
    # If the GetLatestConfiguration API call fails with any of these problems (e.g.  
    # token is EXPIRED or CORRUPTED),  
    # we need to call StartConfigurationSession again to obtain a new configuration  
    # token before retrying.  
    SESSION_RESTART_REQUIRED = {"EXPIRED", "CORRUPTED"}  
  
    def __init__(self):  
        # AWS AppConfig Data SDK client used to interact with the AWS AppConfig Data  
        # service.  
        self.appconfigdata = boto3.client('appconfigdata')  
  
        # The configuration token to be passed to the next GetLatestConfiguration  
        # API call.  
        self.configuration_token = None  
  
        # The cached configuration data to be returned when there is no new  
        # configuration data available.  
        self.configuration = None  
  
    # Returns the latest configuration data stored in AWS AppConfig.  
    def get_config(self):  
        # If there is no configuration token yet, get one by starting a new session  
        # with the StartConfigurationSession API.  
        # Note that this API does not return configuration data. Rather, it returns  
        # an initial configuration token that is  
        # subsequently passed to the GetLatestConfiguration API.  
        if not self.configuration_token:  
            self._start_new_session()  
  
            response = None  
            try:  
                # Retrieve the configuration from the GetLatestConfiguration API,  
                # providing the current configuration token.  
                # If this caller does not yet have the latest configuration (e.g. this  
                # is the first call to GetLatestConfiguration
```

```
        # or new configuration data has been deployed since the first call), the
        latest configuration data will be returned.

        # Otherwise, the GetLatestConfiguration API will not return any data
        since the caller already has the latest.

        response = self.appconfigdata.get_latest_configuration(
            ConfigurationToken=self.configuration_token
        )
    except ClientError as e:
        error_code = e.response.get("Error", {}).get("Code")
        # ResourceNotFoundException – usually means the token/session is invalid
        or expired
        if error_code == "ResourceNotFoundException":
            print("Resource not found – refreshing session and retrying...")
            self._start_new_session()
            response = self.appconfigdata.get_latest_configuration(
                ConfigurationToken=self.configuration_token
            )
        # BadRequestException – check if it's expired or corrupted token
        elif error_code == "BadRequestException":
            details = e.response.get("Error", {}).get("Details", {}) or {}
            invalid_params = details.get("InvalidParameters", {}) or {}
            needs_new_session = any(
                param.get("Problem") in self.SESSION_RESTART_REQUIRED
                for param in invalid_params.values()
            )
            if needs_new_session:
                print("Configuration token expired or corrupted – refreshing
session and retrying...")
                self._start_new_session()
                response = self.appconfigdata.get_latest_configuration(
                    ConfigurationToken=self.configuration_token
                )
            else:
                raise
        else:
            raise

    if response is None:
        # Should not happen, but return cached config if no response
        return self.configuration

    # Save the returned configuration token so that it can be passed to the next
    GetLatestConfiguration API call.
```

```
# Warning: Not persisting this token for use in the next
GetLatestConfiguration API call may result in higher
# than expected usage costs.
self.configuration_token = response['NextPollConfigurationToken']

# If the GetLatestConfiguration API returned configuration data, update the
# cached configuration with the returned data.
# Otherwise, assume the configuration has not changed, and return the cached
configuration.
config_stream = response.get('Configuration')
if config_stream:
    config_from_api = config_stream.read()
    if config_from_api:
        self.configuration = config_from_api
        print(
            'Configuration contents have changed since the last
GetLatestConfiguration call, new contents = '
            + self.configuration.decode('utf-8', errors='ignore')
        )
    else:
        print('GetLatestConfiguration returned an empty response because we
already have the latest configuration')

    return self.configuration

# Starts a new session with AppConfig and retrieves an initial configuration
# token.
def _start_new_session(self):
    session = self.appconfigdata.start_configuration_session(
        ApplicationIdentifier='MyDemoApp',
        ConfigurationProfileIdentifier='MyConfig',
        EnvironmentIdentifier='Beta'
    )
    self.configuration_token = session['InitialConfigurationToken']
```

## JavaScript

```
/*
The example below uses two AWS AppConfig Data APIs: StartConfigurationSession and
GetLatestConfiguration.
For more information about these APIs, see AWS AppConfig Data
.
```

```
This class is meant to be used as a singleton to retrieve the latest configuration
data from AWS AppConfig.

This class maintains a cache of the latest configuration data in addition to the
configuration token to be
passed to the next GetLatestConfiguration API call.

*/
class AppConfigApiRetriever {
    constructor() {
        /* AWS AppConfig Data SDK client used to interact with the AWS AppConfig Data
        service.
        */
        this.appconfigdata = new AppConfigDataClient();

        /*
        The configuration token to be passed to the next GetLatestConfiguration API
        call.
        */
        this.configurationToken = null;

        /*
        The cached configuration data to be returned when there is no new configuration
        data available.
        */
        this.configuration = null;
    }

    async startSession() {
        /*
        Starts a new session with the StartConfigurationSession API to get an initial
        configuration token.
        */
        const session = await this.appconfigdata.send(
            new StartConfigurationSessionCommand({
                ApplicationIdentifier: "MyDemoApp",
                ConfigurationProfileIdentifier: "MyConfig",
                EnvironmentIdentifier: "Beta"
            })
        );
        this.configurationToken = session.InitialConfigurationToken;
    }

    /*
    Returns the latest configuration data stored in AWS AppConfig.
    */
}
```

```
async getConfig() {
    /*
    If there is no configuration token yet, get one by starting a new session with
    the StartConfigurationSession API.

    Note that this API does not return configuration data. Rather, it returns an
    initial configuration token that is
    subsequently passed to the GetLatestConfiguration API.

    */
    if (!this.configurationToken) {
        await this.startSession();
    }

    let response;
    try {
        /*
        Retrieve the configuration from the GetLatestConfiguration API, providing the
        current configuration token.

        If this caller does not yet have the latest configuration (e.g. this is the
        first call to GetLatestConfiguration
        or new configuration data has been deployed since the first call), the latest
        configuration data will be returned.

        Otherwise, the GetLatestConfiguration API will not return any data since the
        caller already has the latest.

        */
        response = await this.appconfigdata.send(
            new GetLatestConfigurationCommand({
                ConfigurationToken: this.configurationToken
            })
        );
    } catch (err) {
        /*
        Add session restart logic – if the token is invalid or expired, restart the
        session and try once more.

        */
        if (err.name === "ResourceNotFoundException" || err.name ===
        "BadRequestException") {
            console.warn(
                "Configuration token invalid or expired. Restarting session..."
            );
            await this.startSession();
            response = await this.appconfigdata.send(
                new GetLatestConfigurationCommand({
                    ConfigurationToken: this.configurationToken
                })
            );
        }
    }
}
```

```
        );
    } else {
        throw err;
    }
}

/*
Save the returned configuration token so that it can be passed to the next
GetLatestConfiguration API call.
Warning: Not persisting this token for use in the next GetLatestConfiguration
API call may result in higher
than expected usage costs.
*/
this.configurationToken = response.NextPollConfigurationToken;

/*
If the GetLatestConfiguration API returned configuration data, update the cached
configuration with the returned data.
Otherwise, assume the configuration has not changed, and return the cached
configuration.
*/
const configFromApi = response.Configuration
    ? await response.Configuration.transformToString()
    : null;

if (configFromApi) {
    this.configuration = configFromApi;
    console.log(
        "Configuration contents have changed since the last GetLatestConfiguration
call, new contents = " +
        this.configuration
    );
} else {
    console.log(
        "GetLatestConfiguration returned an empty response because we already have
the latest configuration"
    );
}

return this.configuration;
}
}
```

## Pulizia dell'ambiente

Se hai eseguito uno o più esempi di codice in questa sezione, ti consigliamo di utilizzare uno dei seguenti esempi per individuare ed eliminare le AWS AppConfig risorse create da tali esempi di codice. Gli esempi in questa sezione richiamano quanto segue APIs:

- [ListApplications](#)
- [DeleteApplication](#)
- [ListEnvironments](#)
- [DeleteEnvironments](#)
- [ListConfigurationProfiles](#)
- [DeleteConfigurationProfile](#)
- [ListHostedConfigurationVersions](#)
- [DeleteHostedConfigurationVersion](#)

### Java

```
/*
 This sample provides cleanup code that deletes all the AWS AppConfig resources
 created in the samples above.

 WARNING: this code will permanently delete the given application and all of its
 sub-resources, including
 configuration profiles, hosted configuration versions, and environments. DO NOT
 run this code against
 an application that you may need in the future.

 */

public void cleanUpDemoResources() {
    AppConfigClient appconfig = AppConfigClient.create();

    // The name of the application to delete
    // IMPORTANT: verify this name corresponds to the application you wish to
    delete
    String applicationToDelete = "MyDemoApp";

    appconfig.listApplicationsPaginator(ListApplicationsRequest.builder().build()).items().forE
    -> {
```

```

        if (app.name().equals(applicationToDelete)) {
            System.out.println("Deleting App: " + app);
            appconfig.listConfigurationProfilesPaginator(req ->
req.applicationId(app.id()).items().forEach(cp -> {
            System.out.println("Deleting Profile: " + cp);
            appconfig
                .listHostedConfigurationVersionsPaginator(req -> req
                    .applicationId(app.id())
                    .configurationProfileId(cp.id()))
                .items()
                .forEach(hcv -> {
                    System.out.println("Deleting HCV: " + hcv);
                    appconfig.deleteHostedConfigurationVersion(req -> req
                        .applicationId(app.id())
                        .configurationProfileId(cp.id())
                        .versionNumber(hcv.versionNumber()));
                });
            appconfig.deleteConfigurationProfile(req -> req
                .applicationId(app.id())
                .configurationProfileId(cp.id()));
        });
    }

    appconfig.listEnvironmentsPaginator(req-
>req.applicationId(app.id()).items().forEach(env -> {
        System.out.println("Deleting Environment: " + env);
        appconfig.deleteEnvironment(req-
>req.applicationId(app.id()).environmentId(env.id()));
    });

    appconfig.deleteApplication(req -> req.applicationId(app.id()));
}
});
}
}

```

## Python

```

# this sample provides cleanup code that deletes all the AWS AppConfig resources
# created in the samples above.
#
# WARNING: this code will permanently delete the given application and all of its
# sub-resources, including
#   configuration profiles, hosted configuration versions, and environments. DO NOT
# run this code against

```

```
# an application that you may need in the future.
#



import boto3

# the name of the application to delete
# IMPORTANT: verify this name corresponds to the application you wish to delete
application_name = 'MyDemoApp'

# create and iterate over a list paginator such that we end up with a list of pages,
# which are themselves lists of applications
# e.g. [ [ {'Name': 'MyApp1', ...}, {'Name': 'MyApp2', ...}, [ {'Name': 'MyApp3', ...} ] ]
list_of_app_lists = [page['Items'] for page in
    appconfig.getPaginator('list_applications').paginate()]
# retrieve the target application from the list of lists
application = [app for apps in list_of_app_lists for app in apps if app['Name'] ==
    application_name][0]
print(f"deleting application {application['Name']} (id={application['Id']})")

# delete all configuration profiles
list_of_config_lists = [page['Items'] for page in
    appconfig.getPaginator('list_configuration_profiles').paginate(ApplicationId=application['Id'])]
for config_profile in [config for configs in list_of_config_lists for config in
    configs]:
    print(f"\tdeleting configuration profile {config_profile['Name']}"
        "(Id={config_profile['Id']})")

    # delete all hosted configuration versions
    list_of_hcv_lists = [page['Items'] for page in
        appconfig.getPaginator('list_hosted_configuration_versions').paginate(ApplicationId=application['Id'],
        ConfigurationProfileId=config_profile['Id'])]
    for hcv in [hcv for hcvs in list_of_hcv_lists for hcv in hcvs]:
        appconfig.delete_hosted_configuration_version(ApplicationId=application['Id'],
        ConfigurationProfileId=config_profile['Id'], VersionNumber=hcv['VersionNumber'])
        print(f"\t\tdeleted hosted configuration version {hcv['VersionNumber']}")

    # delete the config profile itself
    appconfig.delete_configuration_profile(ApplicationId=application['Id'],
    ConfigurationProfileId=config_profile['Id'])
    print(f"\tdeleted configuration profile {config_profile['Name']}"
        "(Id={config_profile['Id']})")

# delete all environments
```

```
list_of_env_lists = [page['Items'] for page in
    appconfig.getPaginator('list_environments').paginate(ApplicationId=application['Id'])]
for environment in [env for envs in list_of_env_lists for env in envs]:
    appconfig.delete_environment(ApplicationId=application['Id'],
        EnvironmentId=environment['Id'])
    print(f"\tdeleted environment {environment['Name']} (Id={environment['Id']})")

# delete the application itself
appconfig.delete_application(ApplicationId=application['Id'])
print(f"deleted application {application['Name']} (id={application['Id']})")
```

## JavaScript

```
// this sample provides cleanup code that deletes all the AWS AppConfig resources
// created in the samples above.

// WARNING: this code will permanently delete the given application and all of its
// sub-resources, including
//   configuration profiles, hosted configuration versions, and environments. DO NOT
//   run this code against
//   an application that you may need in the future.

import {
    AppConfigClient,
    paginateListApplications,
    DeleteApplicationCommand,
    paginateListConfigurationProfiles,
    DeleteConfigurationProfileCommand,
    paginateListHostedConfigurationVersions,
    DeleteHostedConfigurationVersionCommand,
    paginateListEnvironments,
    DeleteEnvironmentCommand,
} from "@aws-sdk/client-appconfig";

const client = new AppConfigClient();

// the name of the application to delete
// IMPORTANT: verify this name corresponds to the application you wish to delete
const application_name = "MyDemoApp";

// iterate over all applications, deleting ones that have the name defined above
for await (const app_page of paginateListApplications({ client }, {})) {
    for (const application of app_page.Items) {
```

```
// skip applications that dont have the name thats set
if (application.Name !== application_name) continue;

console.log(`deleting application ${application.Name} (id=${application.Id})`);

// delete all configuration profiles
for await (const config_page of paginateListConfigurationProfiles({ client },
{ ApplicationId: application.Id })) {
    for (const config_profile of config_page.Items) {
        console.log(`\tdeleting configuration profile ${config_profile.Name} (Id= ${config_profile.Id})`);

        // delete all hosted configuration versions
        for await (const hosted_page of
paginateListHostedConfigurationVersions({ client },
{ ApplicationId: application.Id, ConfigurationProfileId:
config_profile.Id })
) {
            for (const hosted_config_version of hosted_page.Items) {
                await client.send(
                    new DeleteHostedConfigurationVersionCommand({
                        ApplicationId: application.Id,
                        ConfigurationProfileId: config_profile.Id,
                        VersionNumber: hosted_config_version.VersionNumber,
                    })
                );
                console.log(`\t\tdeleted hosted configuration version
${hosted_config_version.VersionNumber}`);
            }
        }
    }
}

// delete the config profile itself
await client.send(
    new DeleteConfigurationProfileCommand({
        ApplicationId: application.Id,
        ConfigurationProfileId: config_profile.Id,
    })
);
console.log(`\tdeleted configuration profile ${config_profile.Name} (Id= ${config_profile.Id})`)

// delete all environments
```

```
    for await (const env_page of paginateListEnvironments({ client },
{ ApplicationId: application.Id })) {
      for (const environment of env_page.Items) {
        await client.send(
          new DeleteEnvironmentCommand({
            ApplicationId: application.Id,
            EnvironmentId: environment.Id,
          })
        );
        console.log(`\tdeleted environment ${environment.Name} (Id=${environment.Id})`)
      }
    }

    // delete the application itself
    await client.send(
      new DeleteApplicationCommand({ ApplicationId: application.Id })
    );
    console.log(`deleted application ${application.Name} (id=${application.Id})`)
  }
}
```

# Configurazione della protezione da AWS AppConfig eliminazione

AWS AppConfig fornisce un'impostazione dell'account per impedire agli utenti di eliminare involontariamente ambienti e profili di configurazione utilizzati attivamente. AWS AppConfig monitora le chiamate verso [GetLatestConfiguration](#) e tiene traccia dei profili [GetConfiguration](#) degli ambienti di configurazione inclusi in tali chiamate entro un intervallo di 60 minuti (impostazione predefinita). Qualsiasi profilo o ambiente di configurazione a cui è stato effettuato l'accesso entro tale intervallo verrà considerato attivo. Se si tenta di eliminare un profilo o un ambiente di configurazione attivo, viene AWS AppConfig restituito un errore. Se necessario, è possibile ignorare questo errore utilizzando il `DeletionProtectionCheck` parametro. Per ulteriori informazioni, consulta [Ignorare o forzare un controllo di protezione da eliminazione](#).

Configura la protezione dall'eliminazione utilizzando la console

Utilizzare la procedura seguente per configurare la protezione da eliminazione utilizzando la AWS Systems Manager console.

Per configurare la protezione da eliminazione (console)

1. Apri la AWS Systems Manager console su <https://console.aws.amazon.com/systems-manager/appconfig/>.
2. Nel pannello di navigazione scegli Impostazioni.
3. Usa l'interruttore per abilitare o disabilitare la protezione da eliminazione.
4. Per Periodo di protezione, imposta la definizione di risorsa attiva tra 15 e 1440 minuti.
5. Fare clic su Apply (Applica).

Configurare la protezione dall'eliminazione utilizzando il AWS CLI

Utilizzare la procedura seguente per configurare la protezione da eliminazione utilizzando AWS CLI. Sostituisci `value` i seguenti comandi con il valore che desideri utilizzare nel tuo ambiente.

### Note

Prima di iniziare, ti consigliamo di eseguire l'aggiornamento alla versione più recente di AWS CLI. Per ulteriori informazioni, consulta [Installare o aggiornare alla versione più recente di AWS CLI nella Guida per l'AWS Command Line Interface utente](#).

Per configurare la protezione da eliminazione (CLI)

1. Esegui il comando seguente per visualizzare le impostazioni correnti di protezione dall'eliminazione.

```
aws appconfig get-account-settings
```

2. Esegui il comando seguente per abilitare o disabilitare la protezione da eliminazione. `false` specificare se disabilitare la protezione `true` da eliminazione o attivarla.

```
aws appconfig update-account-settings --deletion-protection Enabled=value
```

3. È possibile aumentare l'intervallo predefinito fino a un massimo di 24 ore. Esegui il comando seguente per specificare un nuovo intervallo.

```
aws appconfig update-account-settings --deletion-protection  
Enabled=true,ProtectionPeriodInMinutes=a number between 15 and 1440
```

## Ignorare o forzare un controllo di protezione da eliminazione

Per aiutarti a gestire la protezione da eliminazione, [DeleteEnvironmentDeleteConfigurationProfile](#) APIs includi un parametro chiamato `DeletionProtectionCheck`. Questo parametro supporta i seguenti valori:

- **BYPASS**: Indica AWS AppConfig di ignorare il controllo di protezione da eliminazione e di eliminare un profilo di configurazione anche se la protezione da eliminazione lo avrebbe altrimenti impedito.
- **APPLY**: Indica l'esecuzione del controllo di protezione dall'eliminazione, anche se la protezione dall'eliminazione è disabilitata a livello di account. **APPLY** forza inoltre l'esecuzione del controllo di protezione dall'eliminazione sulle risorse create nell'ultima ora, che normalmente sono escluse dai controlli di protezione dall'eliminazione.

- ACCOUNT\_DEFAULT: L'impostazione predefinita, che indica di AWS AppConfig implementare il valore di protezione dall'eliminazione specificato nell'UpdateAccountSettingsAPI.

### Note

Per impostazione predefinita, `DeletionProtectionCheck` ignora i profili e gli ambienti di configurazione creati nell'ultima ora. La configurazione predefinita ha lo scopo di impedire che la protezione da eliminazione interferisca con test e demo che creano risorse di breve durata. È possibile ignorare questo comportamento passando `DeletionProtectionCheck=APPLY` quando si chiama o. `DeleteEnvironment` `DeleteConfigurationProfile`

La seguente procedura dettagliata CLI utilizza comandi di esempio per illustrare come utilizzare il parametro `DeletionProtectionCheck`. Sostituisci *ID* i seguenti comandi con l'ID dei tuoi artefatti. AWS AppConfig

1. Richiama una [GetLatestConfiguration](#) configurazione distribuita.

```
aws appconfigdata get-latest-configuration --configuration-token $(aws appconfigdata start-configuration-session --application-identifier ID --environment-identifier ID --configuration-profile-identifier ID --query InitialConfigurationToken) outfile.txt
```

2. Attendi 60 secondi AWS AppConfig per registrare che la configurazione è attiva.
3. Esegui il comando seguente per richiamare [DeleteEnvironment](#) applicare la protezione da eliminazione nell'ambiente.

```
aws appconfig delete-environment --environment-id ID --application-id ID --deletion-protection-check APPLY
```

Il comando dovrebbe restituire il seguente errore.

```
An error occurred (BadRequestException) when calling the DeleteEnvironment operation: Environment Beta is actively being used in your application and cannot be deleted.
```

4. Esegui il comando seguente per aggirare la protezione da eliminazione ed eliminare l'ambiente.

```
aws appconfig delete-environment --environment-id ID --application-id ID --  
deletion-protection-check BYPASS
```

# Sicurezza in AWS AppConfig

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il [modello di responsabilità condivisa](#) descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi in Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei [AWS Programmi di AWS conformità dei Programmi di conformità](#) dei di . Per ulteriori informazioni sui programmi di conformità applicabili AWS Systems Manager, consulta [AWS Servizi nell'ambito del programma di conformitàAWS](#) .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. L'utente è anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della propria azienda e le leggi e normative vigenti.

AWS AppConfig è uno strumento in AWS Systems Manager. Per capire come applicare il modello di responsabilità condivisa durante l'utilizzo AWS AppConfig, consulta [Security in AWS Systems Manager](#). Questa sezione descrive come configurare Systems Manager per soddisfare gli obiettivi di sicurezza e conformità per AWS AppConfig.

## Implementazione dell'accesso con privilegi minimi

Come best practice di sicurezza, concedi le autorizzazioni minime richieste dalle identità per eseguire azioni specifiche su risorse specifiche in condizioni specifiche. AWS AppConfig L'agente offre due funzionalità che consentono all'agente di accedere al file system di un'istanza o di un contenitore: backup e scrittura su disco. Se abiliti queste funzionalità, verifica che solo l' AWS AppConfig agente disponga delle autorizzazioni di scrittura sui file di configurazione designati sul file system. Verifica inoltre che solo i processi necessari per leggere da questi file di configurazione siano in grado di farlo. L'applicazione dell'accesso con privilegio minimo è fondamentale per ridurre i rischi di sicurezza e l'impatto risultante da errori o intenzioni dannose.

Per ulteriori informazioni sull'implementazione dell'accesso con privilegi minimi, vedere [SEC03-BP02 Garantire l'accesso con privilegi minimi](#) nella Guida per l'utente.AWS Well-Architected Tool Per

ulteriori informazioni sulle funzionalità dell' AWS AppConfig agente menzionate in questa sezione, vedere. [Utilizzo di un manifesto per abilitare funzionalità di recupero aggiuntive](#)

## Crittografia dei dati a riposo per AWS AppConfig

AWS AppConfig fornisce la crittografia di default per proteggere i dati dei clienti inutilizzati. Chiavi di proprietà di AWS

Chiavi di proprietà di AWS— AWS AppConfig utilizza queste chiavi per impostazione predefinita per crittografare automaticamente i dati distribuiti dal servizio e ospitati nell'archivio AWS AppConfig dati. Non è possibile visualizzare, gestire Chiavi di proprietà di AWS, utilizzare o controllare il loro utilizzo. Tuttavia, non è necessario effettuare alcuna operazione o modificare programmi per proteggere le chiavi che eseguono la crittografia dei dati. Per ulteriori informazioni, consulta la sezione [Chiavi di proprietà di AWS](#) nella Guida per gli sviluppatori di AWS Key Management Service .

Sebbene non sia possibile disabilitare questo livello di crittografia o selezionare un tipo di crittografia alternativo, è possibile specificare una chiave gestita dal cliente da utilizzare quando si salvano i dati di configurazione ospitati nel AWS AppConfig data store e quando si distribuiscono i dati di configurazione.

Chiavi gestite dal cliente: AWS AppConfig supporta l'uso di una chiave simmetrica gestita dal cliente che puoi creare, possedere e gestire per aggiungere un secondo livello di crittografia rispetto a quello esistente. Chiave di proprietà di AWS Avendo il pieno controllo di questo livello di crittografia, è possibile eseguire operazioni quali:

- Stabilire e mantenere politiche e sovvenzioni chiave
- Stabilire e mantenere le policy IAM
- Abilitare e disabilitare le policy delle chiavi
- Ruotare i materiali crittografici delle chiavi
- Aggiungere tag
- Creare alias delle chiavi
- Pianificare l'eliminazione delle chiavi

Per ulteriori informazioni, consulta [Customer managed key](#) nella AWS Key Management Service Developer Guide.

AWS AppConfig supporta chiavi gestite dal cliente

AWS AppConfig offre supporto per la crittografia a chiave gestita dal cliente per i dati di configurazione. Per le versioni di configurazione salvate nell'archivio dati AWS AppConfig ospitato, i clienti possono impostare un profilo di configurazione `KmsKeyIdentifier` sul profilo di configurazione corrispondente. Ogni volta che viene creata una nuova versione dei dati di configurazione utilizzando l'operazione `CreateHostedConfigurationVersion` API, AWS AppConfig genera una chiave AWS KMS dati da cui `KmsKeyIdentifier` crittografare i dati prima di archiviarli. Quando successivamente si accede ai dati, durante le operazioni `GetHostedConfigurationVersion` o tramite l'`StartDeploymentAPI`, AWS AppConfig decripta i dati di configurazione utilizzando le informazioni sulla chiave dati generata.

AWS AppConfig offre inoltre supporto per la crittografia a chiave gestita dal cliente per i dati di configurazione distribuiti. Per crittografare i dati di configurazione, i clienti possono fornire una procedura `KmsKeyIdentifier` di implementazione. AWS AppConfig genera la chiave AWS KMS dati con questa chiave `KmsKeyIdentifier` per crittografare i dati sul funzionamento dell'`StartDeploymentAPI`.

### AWS AppConfig accesso alla crittografia

Quando crei una chiave gestita dal cliente, utilizza la seguente politica di chiave per assicurarti che la chiave possa essere utilizzata.

### JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "Allow use of the key",  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::111122223333:role/role_name"  
      },  
      "Action": [  
        "kms:Decrypt",  
        "kms:GenerateDataKey"  
      ],  
      "Resource": "*"  
    }  
  ]  
}
```

Per crittografare i dati di configurazione ospitati con una chiave gestita dal cliente, la chiamata di identità `CreateHostedConfigurationVersion` richiede la seguente dichiarazione di politica, che può essere assegnata a un utente, gruppo o ruolo:

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "kms:GenerateDataKey",  
      "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-ID"  
    }  
  ]  
}
```

Se si utilizza un segreto di Secrets Manager o qualsiasi altro dato di configurazione crittografato con una chiave gestita dal cliente, `retrievalRoleArn` sarà necessario `kms:Decrypt` decrittografare e recuperare i dati.

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "kms:Decrypt",  
      "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-ID"  
    }  
  ]  
}
```

Quando si chiama l'operazione AWS AppConfig [StartDeployment](#) API, la chiamata di identità `StartDeployment` richiede la seguente policy IAM, che può essere assegnata a un utente, gruppo o ruolo:

## JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "kms:GenerateDataKey*"  
      ],  
      "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-ID"  
    }  
  ]  
}
```

Quando si chiama l'operazione AWS AppConfig [GetLatestConfiguration](#) API, la chiamata di identità GetLatestConfiguration richiede la seguente politica che può essere assegnata a un utente, gruppo o ruolo:

## JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "kms:Decrypt",  
      "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-ID"  
    }  
  ]  
}
```

## Contesto di crittografia

Un [contesto di crittografia](#) è un set facoltativo di coppie chiave-valore che contengono ulteriori informazioni contestuali sui dati.

AWS KMS utilizza il contesto di crittografia come dati autenticati aggiuntivi per supportare la crittografia autenticata. Quando includi un contesto di crittografia in una richiesta di crittografia dei

dati, AWS KMS associa il contesto di crittografia ai dati crittografati. Per decriptografare i dati, nella richiesta deve essere incluso lo stesso contesto di crittografia.

AWS AppConfig contesto di crittografia: AWS AppConfig utilizza un contesto di crittografia in tutte le operazioni AWS KMS crittografiche per i dati e le distribuzioni di configurazione ospitati crittografati. Il contesto contiene una chiave corrispondente al tipo di dati e un valore che identifica l'elemento di dati specifico.

## Monitoraggio delle chiavi di crittografia per AWS

Quando utilizzi chiavi gestite AWS KMS dal cliente con AWS AppConfig, puoi utilizzare AWS CloudTrail o Amazon CloudWatch Logs per tenere traccia delle richieste AWS AppConfig inviate a AWS KMS.

L'esempio seguente è un CloudTrail evento per Decrypt monitorare AWS KMS le operazioni richiamate per accedere AWS AppConfig ai dati crittografati dalla chiave gestita dal cliente:

```
{  
  "eventVersion": "1.08",  
  "userIdentity": {  
    "type": "AWSService",  
    "invokedBy": "appconfig.amazonaws.com"  
  },  
  "eventTime": "2023-01-03T02:22:28Z",  
  "eventSource": "kms.amazonaws.com",  
  "eventName": "Decrypt",  
  "awsRegion": "Region",  
  "sourceIPAddress": "172.12.34.56",  
  "userAgent": "ExampleDesktop/1.0 (V1; OS)",  
  "requestParameters": {  
    "encryptionContext": {  
      "aws:appconfig:deployment:arn":  
        "arn:aws:appconfig:Region:account_ID:application/application_ID/  
        environment/environment_ID/deployment/deployment_ID"  
    },  
    "keyId": "arn:aws:kms:Region:account_ID:key/key_ID",  
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT"  
  },  
  "responseElements": null,  
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",  
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",  
  "readOnly": true,  
}
```

```
"resources": [
  {
    "accountId": "account_ID",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:Region:account_ID:key_ID"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "account_ID",
"sharedEventID": "dc129381-1d94-49bd-b522-f56a3482d088"
}
```

## Accesso AWS AppConfig tramite un'interfaccia endpoint ()AWS PrivateLink

Puoi usarlo AWS PrivateLink per creare una connessione privata tra il tuo VPC e AWS AppConfig. Puoi accedere AWS AppConfig come se fosse nel tuo VPC, senza l'uso di un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione Direct Connect. Le istanze del tuo VPC non necessitano di indirizzi IP pubblici per accedervi. AWS AppConfig

Stabilisci questa connessione privata creando un endpoint di interfaccia attivato da AWS PrivateLink. In ciascuna sottorete viene creata un'interfaccia di rete endpoint da abilitare per l'endpoint di interfaccia. Queste sono interfacce di rete gestite dal richiedente che fungono da punto di ingresso per il traffico destinato a AWS AppConfig.

Per ulteriori informazioni, consulta la sezione [Accesso a Servizi AWS tramite AWS PrivateLink](#) nella Guida di AWS PrivateLink .

## Considerazioni per AWS AppConfig

Prima di configurare un endpoint di interfaccia per AWS AppConfig, consulta [le considerazioni nella Guida](#). AWS PrivateLink

AWS AppConfig supporta l'effettuazione di chiamate verso [appconfig](#) [appconfigdata](#) servizi tramite l'endpoint dell'interfaccia.

## Crea un endpoint dell'interfaccia per AWS AppConfig

Puoi creare un endpoint di interfaccia per AWS AppConfig utilizzando la console Amazon VPC o AWS Command Line Interface (AWS CLI). Per ulteriori informazioni, consulta la sezione [Creazione di un endpoint di interfaccia](#) nella Guida per l'utente di AWS PrivateLink .

Crea un endpoint di interfaccia per AWS AppConfig utilizzando i seguenti nomi di servizio:

com.amazonaws.*region*.appconfig

com.amazonaws.*region*.appconfigdata

Se abiliti il DNS privato per l'endpoint dell'interfaccia, puoi effettuare richieste API AWS AppConfig utilizzando il nome DNS regionale predefinito. Ad esempio appconfig.us-east-1.amazonaws.com e appconfigdata.us-east-1.amazonaws.com.

## Creazione di una policy dell'endpoint per l'endpoint dell'interfaccia

Una policy dell'endpoint è una risorsa IAM che è possibile allegare all'endpoint dell'interfaccia. La policy predefinita per gli endpoint consente l'accesso completo AWS AppConfig tramite l'endpoint dell'interfaccia. Per controllare l'accesso consentito AWS AppConfig dal tuo VPC, collega una policy endpoint personalizzata all'endpoint di interfaccia.

Una policy di endpoint specifica le informazioni riportate di seguito:

- I principali che possono eseguire azioni (Account AWS, utenti IAM e ruoli IAM).
- Le azioni che possono essere eseguite.
- Le risorse in cui è possibile eseguire le operazioni.

Per ulteriori informazioni, consulta la sezione [Controllo dell'accesso ai servizi con policy di endpoint](#) nella Guida di AWS PrivateLink .

Esempio: policy degli endpoint VPC per le azioni AWS AppConfig

Di seguito è riportato l'esempio di una policy dell'endpoint personalizzata. Se collegata a un endpoint dell'interfaccia, questa policy concede l'accesso alle operazioni AWS AppConfig elencate per tutti i principali su tutte le risorse.

```
{
```

```
"Statement": [  
  {  
    "Principal": "*",  
    "Effect": "Allow",  
    "Action": [  
      "appconfig:CreateApplication",  
      "appconfig:CreateEnvironment",  
      "appconfig:CreateConfigurationProfile",  
      "appconfig:StartDeployment",  
      "appconfig:GetLatestConfiguration"  
      "appconfig:StartConfigurationSession"  
    ],  
    "Resource": "*"  
  }  
]  
]
```

## Rotazione dei tasti di Secrets Manager

Questa sezione descrive importanti informazioni di sicurezza sull' AWS AppConfig integrazione con Secrets Manager. Per informazioni su Secrets Manager, vedi [Cos'è Gestione dei segreti AWS?](#) nella Guida Gestione dei segreti AWS per l'utente.

### Impostazione della rotazione automatica dei segreti di Secrets Manager distribuita da AWS AppConfig

La rotazione è il processo di aggiornamento periodico di un segreto archiviato in Secrets Manager. Quando si ruota un segreto, vengono aggiornati sia il segreto in Secrets Manager che le credenziali del database o del servizio. È possibile configurare la rotazione automatica dei segreti in Secrets Manager utilizzando una AWS Lambda funzione per aggiornare il segreto e il database. Per ulteriori informazioni, consulta [Ruotare Gestione dei segreti AWS i segreti](#) nella Guida per l'Gestione dei segreti AWS utente.

Per abilitare la rotazione delle chiavi dei segreti di Secrets Manager distribuiti da AWS AppConfig, aggiorna la funzione di rotazione Lambda e distribuisci il segreto ruotato.

#### Note

Implementa il tuo profilo di AWS AppConfig configurazione dopo che il segreto è stato ruotato e completamente aggiornato alla nuova versione. Puoi determinare se il segreto è ruotato

perché lo stato `VersionStage` cambia da `AWSPENDING` a `AWSCURRENT` il completamento della rotazione segreta avviene all'interno della `finish_secret` funzione Secrets Manager Rotation Templates.

Ecco un esempio di funzione che avvia una AWS AppConfig distribuzione dopo la rotazione di un segreto.

```
import time
import boto3
client = boto3.client('appconfig')

def finish_secret(service_client, arn, new_version):
    """Finish the rotation by marking the pending secret as current
    This method finishes the secret rotation by staging the secret staged AWSPENDING
    with the AWSCURRENT stage.

    Args:
        service_client (client): The secrets manager service client
        arn (string): The secret ARN or other identifier
        new_version (string): The new version to be associated with the secret
    """
    # First describe the secret to get the current version
    metadata = service_client.describe_secret(SecretId=arn)
    current_version = None
    for version in metadata["VersionIdsToStages"]:
        if "AWSCURRENT" in metadata["VersionIdsToStages"][version]:
            if version == new_version:
                # The correct version is already marked as current, return
                logger.info("finishSecret: Version %s already marked as AWSCURRENT for %s" % (version, arn))
                return
            current_version = version
            break

    # Finalize by staging the secret version current
    service_client.update_secret_version_stage(SecretId=arn, VersionStage="AWSCURRENT",
                                                MoveToVersionId=new_version, RemoveFromVersionId=current_version)

    # Deploy rotated secret
    response = client.start_deployment(
        ApplicationId='TestApp',
        EnvironmentId='TestEnvironment',
```

```
DeploymentStrategyId='TestStrategy',
ConfigurationProfileId='ConfigurationProfileId',
ConfigurationVersion=new_version,
KmsKeyIdentifier=key,
Description='Deploy secret rotated at ' + str(time.time())
)

logger.info("finishSecret: Successfully set AWSCURRENT stage to version %s for
secret %s." % (new_version, arn))
```

# Monitoraggio AWS AppConfig

Il monitoraggio è un elemento importante per mantenere l'affidabilità, la disponibilità e le prestazioni delle AWS AppConfig oltre AWS soluzioni esistenti. AWS fornisce i seguenti strumenti di monitoraggio per osservare AWS AppConfig, segnalare quando qualcosa non va e intraprendere azioni automatiche quando necessario:

- Amazon CloudWatch monitora AWS le tue risorse e le applicazioni su cui esegui AWS in tempo reale. Puoi raccogliere i parametri e tenerne traccia, creare pannelli di controllo personalizzati e impostare allarmi per inviare una notifica o intraprendere azioni quando un parametro specificato raggiunge una determinata soglia. Ad esempio, puoi tenere CloudWatch traccia dell'utilizzo della CPU o di altri parametri delle tue EC2 istanze Amazon e avviare automaticamente nuove istanze quando necessario. Per ulteriori informazioni, consulta la [Amazon CloudWatch User Guide](#).
- AWS CloudTrail acquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo AWS account e invia i file di log a un bucket Amazon S3 da te specificato. Puoi identificare quali utenti e account hanno chiamato AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute le chiamate. Per ulteriori informazioni, consulta la [AWS CloudTrail Guida per l'utente di](#).
- Amazon CloudWatch Logs ti consente di monitorare, archiviare e accedere ai tuoi file di registro da EC2 istanze Amazon e altre fonti. CloudTrail CloudWatch I log possono monitorare le informazioni nei file di registro e avvisarti quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la [Amazon CloudWatch Logs User Guide](#).
- Amazon EventBridge può essere utilizzato per automatizzare i AWS servizi e rispondere automaticamente agli eventi di sistema, come problemi di disponibilità delle applicazioni o modifiche delle risorse. Gli eventi AWS relativi ai servizi vengono forniti quasi EventBridge in tempo reale. Puoi compilare regole semplici che indichino quali eventi sono considerati di interesse per te e quali operazioni automatizzate intraprendere quando un evento corrisponde a una regola. Per ulteriori informazioni, consulta [Amazon EventBridge User Guide](#).

## Argomenti

- [Registrazione delle chiamate AWS AppConfig API utilizzando AWS CloudTrail](#)
- [Registrazione delle metriche per le chiamate sul piano AWS AppConfig dati](#)
- [Monitoraggio delle implementazioni per il rollback automatico](#)

# Registrazione delle chiamate AWS AppConfig API utilizzando AWS CloudTrail

AWS AppConfig è integrato con [AWS CloudTrail](#), un servizio che fornisce una registrazione delle azioni intraprese da un utente, ruolo o un Servizio AWS. CloudTrail acquisisce tutte le chiamate API AWS AppConfig come eventi. Le chiamate acquisite includono chiamate dalla AWS AppConfig console e chiamate di codice alle operazioni AWS AppConfig API. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare a quale richiesta è stata effettuata AWS AppConfig, l'indirizzo IP da cui è stata effettuata la richiesta, quando è stata effettuata e ulteriori dettagli.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali utente root o utente.
- Se la richiesta è stata effettuata per conto di un utente del Centro identità IAM.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro Servizio AWS.

CloudTrail è attivo nel tuo account Account AWS quando crei l'account e hai automaticamente accesso alla cronologia degli CloudTrail eventi. La cronologia CloudTrail degli eventi fornisce un record visualizzabile, ricercabile, scaricabile e immutabile degli ultimi 90 giorni di eventi di gestione registrati in un. Regione AWS Per ulteriori informazioni, consulta [Lavorare con la cronologia degli CloudTrail eventi](#) nella Guida per l'utente AWS CloudTrail. Non sono CloudTrail previsti costi per la visualizzazione della cronologia degli eventi.

Per una registrazione continua degli eventi degli Account AWS ultimi 90 giorni, crea un trail o un data store di eventi [CloudTrailLake](#).

## CloudTrail sentieri

Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Tutti i percorsi creati utilizzando il Console di gestione AWS sono multiregionali. È possibile creare un trail per una singola Regione o per più Regioni tramite AWS CLI. La creazione di un percorso multiregionale è consigliata in quanto consente di registrare l'intera attività del proprio Regioni AWS account.

Se si crea un trail per una singola Regione, è possibile visualizzare solo gli eventi registrati nella

Regione AWS del trail. Per ulteriori informazioni sui trail, consulta [Creating a trail for your Account AWS](#) e [Creating a trail for an organization](#) nella Guida per l'utente di AWS CloudTrail .

Puoi inviare gratuitamente una copia dei tuoi eventi di gestione in corso al tuo bucket Amazon S3 CloudTrail creando un percorso, tuttavia ci sono costi di storage di Amazon S3. [Per ulteriori informazioni sui CloudTrail prezzi, consulta la pagina Prezzi.AWS CloudTrail](#) Per informazioni sui prezzi di Amazon S3, consulta [Prezzi di Amazon S3](#).

## CloudTrail Archivi di dati sugli eventi di Lake

CloudTrail Lake ti consente di eseguire query basate su SQL sui tuoi eventi. CloudTrail [Lake converte gli eventi esistenti in formato JSON basato su righe in formato Apache ORC](#). ORC è un formato di archiviazione a colonne ottimizzato per il recupero rapido dei dati. Gli eventi vengono aggregati in archivi di dati degli eventi, che sono raccolte di eventi immutabili basate sui criteri selezionati applicando i [selettori di eventi avanzati](#). I selettori applicati a un archivio di dati degli eventi controllano quali eventi persistono e sono disponibili per l'esecuzione della query. Per ulteriori informazioni su CloudTrail Lake, consulta [Working with AWS CloudTrail Lake](#) nella Guida per l'utente.AWS CloudTrail

CloudTrail Gli archivi e le richieste di dati sugli eventi di Lake comportano dei costi. Quando crei un datastore di eventi, scegli l'[opzione di prezzo](#) da utilizzare per tale datastore. L'opzione di prezzo determina il costo per l'importazione e l'archiviazione degli eventi, nonché il periodo di conservazione predefinito e quello massimo per il datastore di eventi. [Per ulteriori informazioni sui CloudTrail prezzi, consulta la sezione Prezzi.AWS CloudTrail](#)

## AWS AppConfig eventi relativi ai dati in CloudTrail

[Gli eventi relativi ai dati](#) forniscono informazioni sulle operazioni eseguite sulle risorse su o all'interno di una risorsa (ad esempio, recuperando l'ultima configurazione distribuita tramite chiamata). GetLatestConfiguration Queste operazioni sono definite anche operazioni del piano dei dati. Gli eventi di dati sono spesso attività che interessano volumi elevati di dati. Per impostazione predefinita, CloudTrail non registra gli eventi relativi ai dati. La cronologia CloudTrail degli eventi non registra gli eventi relativi ai dati.

Per gli eventi di dati sono previsti costi aggiuntivi. Per ulteriori informazioni sui CloudTrail prezzi, consulta la sezione [AWS CloudTrail Prezzi](#).

Puoi registrare gli eventi relativi ai dati per i tipi di AWS AppConfig risorse utilizzando la CloudTrail console o AWS CLI le operazioni CloudTrail dell'API. La [tabella](#) in questa sezione mostra i tipi di risorse disponibili per AWS AppConfig.

- Per registrare gli eventi relativi ai dati utilizzando la CloudTrail console, crea un [percorso](#) o un [data store di eventi](#) per registrare gli eventi di dati oppure [aggiorna un trail o un data store di eventi esistente](#) per registrare gli eventi di dati.
  1. Scegli Data events per registrare gli eventi relativi ai dati.
  2. Dall'elenco dei tipi di eventi Data, scegli AWS AppConfig.
  3. Scegli il modello di selettore di registro che desideri utilizzare. Puoi registrare tutti gli eventi relativi ai dati per il tipo di risorsa, registrare tutti `readOnly` gli eventi, registrare tutti `writeOnly` gli eventi o creare un modello di selettore di registro personalizzato per filtrare i `readOnly` campi `eventName`, `resourcesARN`.
  4. Per il nome del selettore, immettere `AppConfigDataEvents`. Per informazioni sull'attivazione di Amazon CloudWatch Logs per il percorso degli eventi relativi ai dati, consulta [Registrazione delle metriche per le chiamate sul piano AWS AppConfig dati](#).
- Per registrare gli eventi relativi ai dati utilizzando il AWS CLI, configura il `--advanced-event-selectors` parametro in modo che il `eventCategory` campo sia uguale Data e il `resources.type` campo uguale al valore del tipo di risorsa (vedi [tabella](#)). È possibile aggiungere condizioni per filtrare i valori dei `resourcesARN` campi `readOnlyeventName`, e.
  - Per configurare un percorso per registrare gli eventi relativi ai dati, esegui [put-event-selectors](#) comando. Per ulteriori informazioni, vedere [Registrazione degli eventi relativi ai dati per i AWS CLI sentieri con](#).
  - Per configurare un Event Data Store per registrare gli eventi relativi ai dati, esegui il [create-event-data-store](#) comando per creare un nuovo archivio dati di eventi per registrare gli eventi di dati o eseguire il [update-event-data-store](#) comando per aggiornare un archivio dati di eventi esistente. Per ulteriori informazioni, vedere [Registrazione degli eventi di dati per gli archivi dati degli AWS CLI eventi con](#).

La tabella seguente elenca i tipi di AWS AppConfig risorse. La colonna Data event type (console) mostra il valore da scegliere dall'elenco Data event type sulla CloudTrail console. La colonna del valore `resources.type` mostra il `resources.type` valore da specificare durante la configurazione dei selettori di eventi avanzati utilizzando o. AWS CLI CloudTrail APIs La CloudTrail colonna Dati APIs registrati mostra le chiamate API registrate per il tipo di risorsa. CloudTrail

Tipo di evento di dati (console)	valore resources.type	Dati APIs registrati su* CloudTrail
AWS AppConfig	AWS::AppConfig::Configuration	<ul style="list-style-type: none"> <li>• <a href="#">GetLatestConfiguration</a></li> <li>• <a href="#">StartConfigurationSession</a></li> </ul>

\*Puoi configurare selettori di eventi avanzati per filtrare in base a eventName.readOnly, e resources.ARN campi per registrare solo gli eventi che ritieni importanti. Per ulteriori informazioni su questi campi, vedere [AdvancedFieldSelector](#).

## AWS AppConfig eventi di gestione in CloudTrail

AWS AppConfig registra tutte le operazioni AWS AppConfig del piano di controllo come eventi di gestione. Per un elenco delle operazioni del piano di AWS AppConfig controllo a cui si AWS AppConfig effettua l'accesso CloudTrail, consulta l'[AWS AppConfig API](#) Reference.

### AWS AppConfig esempi di eventi

Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'operazione API richiesta, la data e l'ora dell'operazione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi gli eventi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra un CloudTrail evento che dimostra l'[StartConfigurationSession](#) operazione.

```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/Administrator",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {},
      "attributes": {
        "creationDate": "2024-01-11T14:37:02Z",
        "mfaAuthenticated": false
      }
    }
  }
}
```

```
        "mfaAuthenticated": "false"
    }
}
},
"eventTime": "2024-01-11T14:45:15Z",
"eventSource": "appconfig.amazonaws.com",
"eventName": "StartConfigurationSession",
"awsRegion": "us-east-1",
"sourceIPAddress": "203.0.113.0",
"userAgent": "Boto3/1.34.11 md/Botocore#1.34.11 ua/2.0 os/macos#22.6.0
md/arch#x86_64 lang/python#3.11.4 md/pyimpl#CPython cfg/retry-mode#legacy
Botocore/1.34.11",
"requestParameters": {
    "applicationIdentifier": "rrfexample",
    "environmentIdentifier": "mexampleqe0",
    "configurationProfileIdentifier": "3eexampleu1"
},
"responseElements": null,
"requestID": "a1b2c3d4-5678-90ab-cdef-aaaaaaEXAMPLE",
"eventID": "a1b2c3d4-5678-90ab-cdef-bbbbEXAMPLE",
"readOnly": false,
"resources": [
{
    "accountId": "123456789012",
    "type": "AWS::AppConfig::Configuration",
    "ARN": "arn:aws:appconfig:us-east-1:123456789012:application/rrfexample/
environment/mexampleqe0/configuration/3eexampleu1"
}
],
"eventType": "AwsApiCall",
"managementEvent": false,
"recipientAccountId": "123456789012",
"eventCategory": "Data",
"tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_128_GCM_SHA256",
    "clientProvidedHostHeader": "appconfigdata.us-east-1.amazonaws.com"
}
}
```

Per informazioni sul contenuto dei CloudTrail record, consultate il [contenuto dei CloudTrail record](#) nella Guida per l'AWS CloudTrail utente.

# Registrazione delle metriche per le chiamate sul piano AWS AppConfig dati

Se hai configurato AWS CloudTrail per registrare gli eventi AWS AppConfig relativi ai dati, puoi consentire ad Amazon CloudWatch Logs di registrare i parametri per le chiamate sul piano AWS AppConfig dati. Puoi quindi cercare e filtrare i dati di log in CloudWatch Logs creando uno o più filtri metrici. I filtri metrici definiscono i termini e i modelli da cercare nei dati di registro quando vengono inviati ai registri. CloudWatch CloudWatch Logs utilizza filtri metrici per trasformare i dati di registro in metriche numeriche. CloudWatch Puoi rappresentare graficamente le metriche o configurarle con un allarme.

## Prima di iniziare

Abilita la registrazione degli eventi AWS AppConfig relativi ai dati in. AWS CloudTrail La procedura seguente descrive come abilitare la registrazione metrica per un trail in esistente AWS AppConfig . CloudTrail Per informazioni su come abilitare la CloudTrail registrazione per le chiamate al piano AWS AppConfig dati, vedere. [AWS AppConfig eventi relativi ai dati in CloudTrail](#)

Utilizzare la procedura seguente per consentire a CloudWatch Logs di registrare le metriche per le chiamate al AWS AppConfig piano dati.

Per consentire a CloudWatch Logs di registrare le metriche per le chiamate sul piano dati AWS AppConfig

1. Apri la CloudTrail console all'indirizzo. <https://console.aws.amazon.com/cloudtrail/>
2. Nella dashboard, scegli il tuo AWS AppConfig percorso.
3. Nella sezione CloudWatch Registri, scegli Modifica.
4. Scegli Enabled (Abilitato).
5. Per il nome del gruppo di log, lascia il nome predefinito o inserisci un nome. Prendere nota del nome. Sceglierai il gruppo di log nella console CloudWatch Logs in un secondo momento.
6. In Role name (Nome ruolo), immettere un nome.
7. Scegli Save changes (Salva modifiche).

Utilizza la procedura seguente per creare una metrica e un filtro metrico per AWS AppConfig Logs. CloudWatch La procedura descrive come creare un filtro metrico per le chiamate da operation e (facoltativamente) le chiamate da e. operation Amazon Resource Name (ARN)

Per creare una metrica e un filtro metrico per in Logs AWS AppConfig CloudWatch

1. Apri la console all' CloudWatch indirizzo. <https://console.aws.amazon.com/cloudwatch/>
2. Nel pannello di navigazione scegli Log, quindi Gruppi di log.
3. Scegli la casella di controllo accanto al gruppo di AWS AppConfig log.
4. Scegli Actions (Operazioni) e quindi Create metric filter (Crea filtro parametri).
5. Per Nome del filtro, inserisci un nome.
6. Per Pattern di filtro, inserisci quanto segue:

```
{ $.eventSource = "appconfig.amazonaws.com" }
```

7. (Facoltativo) Nella sezione Schema di test, scegliete il gruppo di log dall'elenco Selezione i dati di registro da testare. Se CloudTrail non ha registrato alcuna chiamata, puoi saltare questo passaggio.
8. Scegli Next (Successivo).
9. Per Metric namespace, inserisci **AWS AppConfig**
10. Per Nome parametro, inserire **Calls**.
11. In Metric value (Valore parametro), inserisci **1**.
12. Ignora il valore e l'unità predefiniti.
13. Per Nome dimensione, immettere **operation**.
14. Per Valore della dimensione, immettere **\$.eventName**.

(Facoltativo) Puoi inserire una seconda dimensione che include l'Amazon Resource Name (ARN) che effettua la chiamata. Per aggiungere una seconda dimensione, in Nome dimensione, inserisci **resource**. Per Valore della dimensione, immettere **\$.resources[0].ARN**.

Scegli Next (Successivo).

15. Controlla i dettagli del filtro e crea un filtro metrico.

(Facoltativo) Puoi ripetere questa procedura per creare un nuovo filtro metrico per un codice di errore specifico come AccessDenied. Se lo fai, inserisci i seguenti dettagli:

1. Per Nome filtro, inserisci un nome.
2. Per Pattern di filtro, inserisci quanto segue:

```
{ $.errorCode = "codename" }
```

Ad esempio

```
{ $.errorCode = "AccessDenied" }
```

3. Per Metric namespace, immettere **AWS AppConfig**
4. Per Nome parametro, inserire **Errors**.
5. In Metric value (Valore parametro), inserisci **1**.
6. Per Valore predefinito, inserite uno zero (0).
7. Ignora unità, dimensioni e allarmi.

Dopo aver CloudTrail registrato le chiamate API, puoi visualizzare le metriche in CloudWatch Per ulteriori informazioni, consulta [Visualizzazione delle metriche e dei log nella console nella Amazon CloudWatch User Guide](#). Per informazioni su come individuare una metrica che hai creato, consulta [Cerca metriche disponibili](#).

 Note

Se imposta la metrica di errore senza dimensione, come descritto qui, puoi visualizzare tali metriche nella pagina Metriche senza dimensione.

## Creazione di un allarme per una metrica CloudWatch

Dopo aver creato le metriche, puoi creare allarmi metrici in CloudWatch Ad esempio, puoi creare un allarme per la metrica delle AWS AppConfig chiamate creata nella procedura precedente. In particolare, è possibile creare un allarme per le chiamate all'azione dell' AWS AppConfig StartConfigurationSessionAPI che superano una soglia. Per informazioni su come creare un allarme per una metrica, consulta [Creare un CloudWatch allarme basato su una soglia statica](#) nella Amazon CloudWatch User Guide. Per informazioni sui limiti predefiniti per le chiamate al piano AWS AppConfig dati, consulta [Limiti predefiniti del piano dati](#) nel Riferimenti generali di Amazon Web Services.

# Monitoraggio delle implementazioni per il rollback automatico

Durante una distribuzione, puoi mitigare le situazioni in cui dati di configurazione errati o non corretti causano errori nell'applicazione utilizzando una combinazione di [strategie di AWS AppConfig distribuzione](#) e rollback automatici basati sugli allarmi Amazon. CloudWatch Una volta configurato, se uno o più CloudWatch allarmi entrano nello INSUFFICIENT\_DATA stato ALARM o durante una distribuzione, ripristina AWS AppConfig automaticamente i dati di configurazione alla versione precedente, evitando così interruzioni o errori dell'applicazione.

## Note

Una distribuzione non viene ripristinata automaticamente se le azioni sono state disabilitate in un allarme associato CloudWatch .

Puoi disabilitare e abilitare gli allarmi utilizzando le azioni [DisableAlarmActions](#) [EnableAlarmActions](#) API o i [enable-alarm-actions](#) comandi [disable-alarm-actions](#) and in. AWS CLI

Puoi anche ripristinare una configurazione richiamando l'operazione [StopDeployment](#) API mentre è ancora in corso una distribuzione.

## Important

Per le distribuzioni che vengono completate correttamente, supporta AWS AppConfig anche il ripristino dei dati di configurazione a una versione precedente utilizzando il `AllowRevert` parametro con l'[StopDeployment](#) operazione API. Per alcuni clienti, il ripristino di una configurazione precedente dopo una corretta implementazione garantisce che i dati rimangano gli stessi di prima della distribuzione. Il ripristino ignora anche i monitor degli allarmi, il che può impedire il proseguimento del rollforward durante un'emergenza dell'applicazione. Per ulteriori informazioni, consulta [Ripristino di una configurazione](#).

Per configurare i rollback automatici, specifichi l'Amazon Resource Name (ARN) di uno o CloudWatch più parametri nel campo degli allarmi quando CloudWatch crei (o modifichi) un ambiente. AWS AppConfig Per ulteriori informazioni, consulta [Creazione di ambienti per l'applicazione in AWS AppConfig](#).

### Note

Se utilizzi una soluzione di monitoraggio di terze parti (ad esempio Datadog), puoi creare un' AWS AppConfig estensione che verifica la presenza di allarmi nel punto di AT\_DEPLOYMENT\_TICK azione e, come barriera di sicurezza, annulla l'implementazione se ha attivato un allarme. Per ulteriori informazioni sulle estensioni, consulta [AWS AppConfig Estendere i AWS AppConfig flussi di lavoro utilizzando le estensioni](#). Per ulteriori informazioni sulle estensioni personalizzate, consulta [Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig](#). Per visualizzare un esempio di codice di un' AWS AppConfig estensione che utilizza il punto di AT\_DEPLOYMENT\_TICK azione per l'integrazione con Datadog, consulta [aws-samples/-for-datadog on. aws-appconfig-tick-extn GitHub](#)

## Metriche consigliate da monitorare per il rollback automatico

Le metriche che scegli di monitorare dipenderanno dall'hardware e dal software utilizzati dalle tue applicazioni. AWS AppConfig i clienti spesso monitorano le seguenti metriche. Per un elenco completo delle metriche consigliate raggruppate per Servizio AWS, consulta [Allarmi consigliati](#) nella Amazon CloudWatch User Guide.

Dopo aver determinato le metriche che desideri monitorare, CloudWatch usale per configurare gli allarmi. Per ulteriori informazioni, consulta [Usare gli CloudWatch allarmi Amazon](#).

Servizio	Parametro	Informazioni
<a href="#">Gateway Amazon API</a>	4 XXError	Questo allarme rileva un elevato tasso di errori lato client. Ciò può indicare un problema nei parametri di autorizzazione o di richiesta del client. Potrebbe anche significare che una risorsa è stata rimossa o che un client ne sta richiedendo una che non esiste. Valuta la possibilità di abilitare Amazon CloudWatch Logs e di verificar

Servizio	Parametro	Informazioni
		<p>e la presenza di eventuali errori che potrebbero causare gli errori 4XX. Inoltre, valuta la possibilità di abilitare CloudWatch parametri dettagliati per visualizzare questi parametri per risorsa e metodo e restringere l'origine degli errori. Gli errori potrebbero essere causati anche dal superamento del limite di limitazione della larghezza di banda della rete impostato.</p>
<a href="#"><u>Gateway Amazon API</u></a>	5XXError	Questo allarme aiuta a rilevare un elevato tasso di errori lato server. Ciò può indicare un errore nel back-end dell'API, nella rete o nell'integrazione tra il gateway API e l'API di back-end.

Servizio	Parametro	Informazioni
<a href="#">Gateway Amazon API</a>	Latenza	<p>Questo allarme rileva un'elevata latenza in una fase. Trova il valore del parametro <code>IntegrationLatency</code> per verificare la latenza del back-end dell'API. Se i due parametri sono per lo più allineati, il back-end dell'API è l'origine della latenza più elevata e dovrebbe essere esaminato per individuare eventuali problemi. Prendi in considerazione anche l'attivazione CloudWatch dei log e il controllo degli errori che potrebbero causare l'elevata latenza.</p>
<a href="#">Amazon EC2 Auto Scaling</a>	<code>GroupInServiceCapacity</code>	<p>Questo allarme aiuta a rilevare quando la capacità del gruppo è inferiore alla capacità desiderata richiesta per il carico di lavoro. Per risolvere il problema, controlla le attività di dimensionamento per individuare eventuali errori di avvio e verifica che la configurazione della capacità desiderata sia corretta.</p>

Servizio	Parametro	Informazioni
<a href="#">Amazon EC2</a>	CPUUtilization	Questo allarme aiuta a monitorare l'utilizzo della CPU di un' EC2 istanza. A seconda dell'applicazione, potrebbero essere normali livelli di utilizzo costantemente elevati. Tuttavia, se le prestazioni si deteriorano e l'applicazione non è limitata dall'I/O del disco, dalla memoria o dalle risorse di rete, una CPU al massimo potrebbe indicare un collo di bottiglia in termini di risorse o problemi di prestazioni delle applicazioni.
<a href="#">Amazon ECS</a>	CPUReservation	Questo allarme consente di rilevare un'elevata prenotazione della CPU del cluster ECS. Una prenotazione della CPU elevata potrebbe indicare che il cluster si sta esaurendo o è registrato CPUs per l'attività.
<a href="#">Amazon ECS</a>	HTTPCode_target_5xx_count	Questo allarme consente di rilevare un elevato numero di errori lato server per il servizio ECS. Ciò può indicare che sono presenti errori che impediscono al server di evadere le richieste.

Servizio	Parametro	Informazioni
<a href="#">Amazon EKS con Container Insights</a>	node_cpu_utilization	Questo allarme aiuta a rilevare un elevato utilizzo della CPU nei nodi di lavoro del cluster Amazon EKS. Un utilizzo costantemente elevato potrebbe indicare la necessità di sostituire i nodi worker con istanze con una CPU maggiore o la necessità di eseguire un dimensionamento orizzontale del sistema.
<a href="#">Amazon EKS con Container Insights</a>	node_memory_utilization	Questo allarme aiuta a rilevare un elevato utilizzo della memoria nei nodi di lavoro del cluster Amazon EKS. Se l'utilizzo è costantemente elevato, potrebbe indicare la necessità di dimensionare il numero di repliche dei pod o ottimizzare l'applicazione.
<a href="#">Amazon EKS con Container Insights</a>	pod_cpu_utilization_over_pod_limit	Questo allarme aiuta a rilevare un elevato utilizzo della CPU nei pod del cluster Amazon EKS. Se l'utilizzo è costantemente elevato, potrebbe indicare la necessità di aumentare il limite di CPU per il pod interessato.

Servizio	Parametro	Informazioni
<a href="#">Amazon EKS con Container Insights</a>	pod_memory_utilization_over_pod_limit	Questo allarme aiuta a rilevare un elevato utilizzo della CPU nei pod del cluster Amazon EKS. Se l'utilizzo è costantemente elevato, potrebbe indicare la necessità di aumentare il limite di CPU per il pod interessato.
<a href="#">AWS Lambda</a>	Errori	Questo allarme rileva un numero elevato di errori. Gli errori includono eccezioni generate dal codice e eccezioni generate dal runtime Lambda.
<a href="#">AWS Lambda</a>	Throttles	Questo allarme rileva un numero elevato di richieste di invocazione limitate. La limitazione della larghezza di banda della rete si verifica quando non è disponibile simultaneità per l'aumento.
<a href="#">Lambda Insight</a>	memory_utilization	Questo allarme viene utilizzato per rilevare se l'utilizzo della memoria di una funzione lambda si avvicina al limite configurato.

Servizio	Parametro	Informazioni
<a href="#">Amazon S3</a>	4xxErrors	Questo allarme ci aiuta a segnalare il numero totale di 4xx codici di stato di errore che vengono creati in risposta alle richieste dei client. I codici di errore 403 potrebbero indicare una politica IAM errata e i codici di errore 404 potrebbero indicare un comportamento scorretto dell'applicazione client, ad esempio.
<a href="#">Amazon S3</a>	5xxErrors	Questo allarme consente di rilevare un numero elevato di errori sul lato server. Questi errori indicano che un client ha effettuato una richiesta che il server non è riuscito a completare. Questo può aiutarti a stabilire un collegamento con il problema che la tua applicazione sta riscontrando a causa di S3.

# AWS AppConfig Cronologia dei documenti della Guida dell'utente

La tabella seguente descrive le modifiche importanti alla documentazione dall'ultima versione di AWS AppConfig.

Versione attuale dell'API: 2019-10-09

Modifica	Descrizione	Data
<a href="#">IPv6 supporto</a>	Tutto AWS AppConfig APIs ora supporto IPv4 e IPv6 chiamate completi. Per ulteriori informazioni, vedere <a href="#">Understanding IPv6 support.</a>	23 aprile 2025
<a href="#">Nuovo argomento: salvataggio di una versione precedente di Feature Flag in una nuova versione</a>	Quando aggiorni un feature flag, salva AWS AppConfig automaticamente le modifiche apportate a una nuova versione. Se desiderate utilizzare una versione precedente del feature flag, dovete copiarla in una versione bozza e salvarla. Non potete modificare e salvare le modifiche a una versione precedente del flag senza salvarla in una nuova versione. Per ulteriori informazioni, consulta <a href="#">Salvare una versione precedente del Feature Flag in una nuova versione.</a>	15 aprile 2025

[Nuovo argomento: esempi di feature flag per la modalità di sviluppo locale di AWS AppConfig Agent](#)

AWS AppConfig Agent supporta una [modalità di sviluppo locale](#). Se si abilita la modalità di sviluppo locale, l'agente legge i dati di configurazione da una directory specificata su disco. Non recupera i dati di configurazione da AWS AppConfig. Per aiutarvi a comprendere meglio come utilizzare la modalità di sviluppo locale, questa guida contiene ora un argomento con esempi di feature flag. Per ulteriori informazioni, consulta [Esempi di Feature flag per la modalità di sviluppo locale di AWS AppConfig Agent](#).

[Nuovo argomento: creazione di un profilo di configurazione per fonti di dati non native](#)

L'argomento descrive il processo di alto livello per l'utilizzo di un' AWS AppConfig estensione per recuperare i dati di configurazione da fonti che non sono supportate nativamente, inclusi altri AWS servizi come Amazon RDS e Amazon DynamoDB, oltre a fonti di terze parti come, o un repository locale. GitHub GitLab [Per ulteriori informazioni, consulta Creazione di un profilo di configurazione per fonti di dati non native](#)

18 febbraio 2025

19 dicembre 2024

## [Argomento aggiornato: regex fisso nel riferimento al tipo di flag di funzionalità](#)

Lo schema json nel riferimento al tipo di flag di funzionalità mostrava in precedenza il seguente schema regex in vari punti: "^[a-z][a-zA-Z]\d-\_\]{0,63}\$" Il modello regex corretto è "^[a-z][a-zA-Z]\d-\_\]{0,63}\$" Il trattino è elencato dopo il trattino basso. Per ulteriori informazioni, consulta [Understanding the type reference for AWS AppConfig. FeatureFlags](#)

18 dicembre 2024

## [Argomenti aggiornati: Aggiunti esempi di variabili di ambiente](#)

Le tabelle che descrivono le variabili di ambiente nei seguenti argomenti sono state aggiornate per includere esempi:

12 dicembre 2024

- [\(Facoltativo\) Utilizzo di variabili di ambiente per configurare AWS AppConfig Agent for Amazon ECS e Amazon EKS](#)
- [\(Facoltativo\) Utilizzo di variabili di ambiente per configurare AWS AppConfig Agent for Amazon EC2](#)
- [Configurazione dell'estensione AWS AppConfig Agent Lambda](#)

## [Nuova sezione: Comprendere l'operatore split](#)

Una nuova sezione utilizza esempi per spiegare come funziona l'operator `split` per una regola di feature flag multivariante. Per ulteriori informazioni, vedere [Comprendere le regole dei feature flag multivarianti](#).

22 novembre 2024

## [Nuovo punto d'azione per l'estensione: AT\\_DEPLOYMENT\\_TICK](#)

AWS AppConfig ha lanciato un nuovo punto di azione per gli utenti che creano estensioni personalizzate. Il punto di AT\_DEPLOYMENT\_TICK azione supporta l'integrazione del monitoraggio di terze parti. AT\_DEPLOYMENT\_TICK viene richiamato durante l'orchestrazione dell'elaborazione della configurazione, della distribuzione. Se utilizzi una soluzione di monitoraggio di terze parti (ad esempio Datadog), puoi creare un' AWS AppConfig estensione che verifica la presenza di allarmi nel punto di AT\_DEPLOYMENT\_TICK azione e, come barriera di sicurezza, ripristina l'implementazione se ha attivato un allarme. [Per ulteriori informazioni sulle AWS AppConfig estensioni, consulta Estendere i flussi di lavoro utilizzando le estensioni AWS AppConfig](#) Per ulteriori informazioni sulle estensioni personalizzate, consulta [Procedura dettagliata: creazione di estensioni personalizzate](#). AWS AppConfig Per visualizzare un esempio di codice di un' AWS AppConfig estensione che utilizza il punto di

22 novembre 2024

	AT_DEPLOYMENT_TICK azione per l'integrazione con Datadog, consulta <a href="#">aws-samples/-for-datadog on. aws-appconfig-tick-extn GitHub</a>	
<a href="#">Nuovo AWS AppConfig argomento: considerazioni sull'uso dei dispositivi mobili</a>	Un nuovo argomento di questa guida descrive importanti considerazioni sull'utilizzo dei flag di AWS AppConfig funzionalità con i dispositivi mobili. Per ulteriori informazioni, consulta Considerazioni sull'uso dei <a href="#">AWS AppConfig dispositivi mobili</a> .	21 novembre 2024
<a href="#">Nuova funzionalità: protezione e dalla AWS AppConfig cancellazione</a>	AWS AppConfig ora fornisce un'impostazione dell'account per impedire agli utenti di eliminare involontariamente ambienti e profili di configurazione utilizzati attivamente. <a href="#">Per ulteriori informazioni, vedere Configurazione della protezione da eliminazione.</a> <a href="#">AWS AppConfig</a>	28 agosto 2024
<a href="#">Nuova versione dell'estensione AWS AppConfig Agent Lambda</a>	L'agente è stato aggiornato con miglioramenti minori e correzioni di bug. Per visualizzare i nuovi Amazon Resource Names (ARNs) per l'estensione, consulta <a href="#">Versioni disponibili dell'estensione AWS AppConfig Agent Lambda</a> .	9 agosto 2024

<a href="#"><u>Nuovi esempi di codice per il recupero delle varianti di bandiera</u></a>	Per ulteriori informazioni, vedete <a href="#"><u>Using AWS AppConfig Agent to retrieve a feature flag with variant</u></a> .	9 agosto 2024
<a href="#"><u>Nuova versione dell'estensione AWS AppConfig Agent Lambda</u></a>	L'agente è stato aggiornato per supportare obiettivi, varianti e suddivisioni con feature flag. Per visualizzare i nuovi Amazon Resource Names (ARNs) per l'estensione, consulta <a href="#"><u>Versioni disponibili dell'estensione AWS AppConfig Agent Lambda</u></a> .	23 luglio 2024

<a href="#"><u>Nuova funzionalità: bandiere di funzionalità multivarianti</u></a>	I flag di funzionalità multivarianti consentono di definire una serie di possibili valori di flag da restituire per una richiesta. È inoltre possibile configurare diversi stati (abilitati o disabilitati) per i flag multivarianti. Quando si richiede un flag configurato con varianti, l'applicazione fornisce un contesto che AWS AppConfig valuta in base a una serie di regole definite dall'utente. A seconda del contesto specificato nella richiesta e delle regole definite per la variante, AWS AppConfig restituisce valori di flag diversi all'applicazione. Per ulteriori informazioni, vedete <a href="#"><u>Creazione di flag di funzionalità multivarianti</u></a> .	23 luglio 2024
<a href="#"><u>Nuova versione dell'estensione AWS AppConfig Agent Lambda</u></a>	L'agente è stato aggiornato con miglioramenti minori e correzioni di bug. Per visualizzare i nuovi Amazon Resource Names (ARNs) per l'estensione, consulta <a href="#"><u>Versioni disponibili dell'estensione AWS AppConfig Agent Lambda</u></a> .	28 febbraio 2024

## [AWS AppConfig esempi di estensioni personalizzate](#)

L'argomento [Procedura dettagliata: creazione di AWS AppConfig estensioni personalizzate](#) ora include collegamenti alle seguenti estensioni di esempio su: GitHub

28 febbraio 2024

- [Estensione di esempio che impedisce le implementazioni con un calendario di blocked day moratoria utilizzando Systems Manager Change Calendar](#)
- [Estensione di esempio che impedisce la divulgazione di segreti nei dati di configurazione utilizzando git-secrets](#)
- [Estensione di esempio che impedisce la fuoriuscita di informazioni di identificazione personale \(PII\) nei dati di configurazione utilizzando Amazon Comprehend](#)

[Nuovo argomento: registrazione delle chiamate AWS AppConfig API utilizzando AWS CloudTrail](#)

AWS AppConfig è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, ruolo o AWS servizio in AWS AppConfig. CloudTrail acquisisce tutte le chiamate API AWS AppConfig come eventi. Questo nuovo argomento fornisce contenuti AWS AppConfig specifici anziché collegamenti al contenuto corrispondente nella Guida per l'utente. AWS Systems Manager Per ulteriori informazioni, consulta [Registrazione delle chiamate AWS AppConfig API utilizzando AWS CloudTrail](#).

18 gennaio 2024

## [AWS AppConfig ora supporta AWS PrivateLink](#)

Puoi usarlo AWS PrivateLink per creare una connessione privata tra il tuo VPC e AWS AppConfig. Puoi accedere AWS AppConfig come se fosse nel tuo VPC, senza l'uso di un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione Direct Connect. Le istanze del tuo VPC non necessitano di indirizzi IP pubblici per accedervi. AWS AppConfig Per ulteriori informazioni, consulta [Accedere AWS AppConfig utilizzando un endpoint di interfaccia](#) (AWS PrivateLink).

6 dicembre 2023

## Funzionalità aggiuntive di recupero degli AWS AppConfig agenti e una nuova modalità di sviluppo locale

AWS AppConfig Agent offre le seguenti funzionalità aggiuntive per aiutarvi a recuperare le configurazioni per le vostre applicazioni.

1 dicembre 2023

### Funzionalità di recupero aggiuntive

- Recupero di più account: utilizza AWS AppConfig Agent da un sistema primario o di recupero Account AWS per recuperare i dati di configurazione da più account fornitore.
- Scrittura della copia della configurazione su disco: utilizza AWS AppConfig Agent per scrivere i dati di configurazione su disco. Questa funzionalità consente l'integrazione con i clienti con applicazioni che leggono i dati di configurazione dal disco AWS AppConfig.

#### Note

La configurazione di scrittura su disco non è progettata come funzionalità di backup della configurazione. AWS

AppConfig L'agente non legge i file di configurazione copiati su disco. Se desideri eseguire il backup delle configurazioni su disco, consulta le variabili di PRELOAD\_BACkUP ambiente per [Using AWS AppConfig Agent with Amazon EC2](#) o [Using AWS AppConfig Agent with Amazon ECS](#) [BACKUP\\_DIRECTORY](#) e Amazon EKS.

## Modalità di sviluppo locale

AWS AppConfig L'agente supporta una modalità di sviluppo locale. Se si abilita la modalità di sviluppo locale, l'agente legge i dati di configurazione da una directory specificata su disco. Non recupera i dati di configurazione da AWS AppConfig. È possibile simulare le distribuzioni di configurazione aggiornando i file nella directory specificata. Consiglio mo la modalità di sviluppo locale per i seguenti casi d'uso:

- Prova diverse versioni di configurazione prima di distribuirle utilizzando AWS AppConfig.
- Prova diverse opzioni di configurazione per una nuova funzionalità prima di apportare modifiche al tuo repository di codice.
- Prova diversi scenari di configurazione per verificare che funzionino come previsto.

[Nuovo argomento relativo agli esempi di codice](#)

È stato aggiunto un nuovo argomento [sugli esempi di codice](#) a questa guida. L'argomento include esempi in Java, Python e JavaScript per eseguire a livello di codice sei azioni comuni. AWS AppConfig

[Sommario rivisto per riflettere meglio il flusso di lavoro AWS AppConfig](#)

Il contenuto di questa guida per l'utente è ora raggruppato sotto i titoli Creazione, distribuzione, recupero ed estensione dei flussi di lavoro. Questa organizzazione riflette meglio il flusso di lavoro per l'utilizzo AWS AppConfig e mira a rendere i contenuti più reperibili.

17 novembre 2023

7 novembre 2023

[È stato aggiunto il riferimento al payload](#)

L'argomento [Creazione di una funzione Lambda per un'AWS AppConfig estensione personalizzata](#) ora include un riferimento al payload di richiesta e risposta.

7 novembre 2023

[Nuova strategia di AWS implementazione predefinita](#)

AWS AppConfig ora offre e consiglia la strategia di implementazione AppConfig `.Linear20PercentEvery6Minutes` predefinita. Per ulteriori informazioni, consulta Strategie di [distribuzione predefinite](#).

11 agosto 2023

[AWS AppConfig integrazione con Amazon EC2](#)

Puoi integrarti AWS AppConfig con le applicazioni in esecuzione sulle tue istanze Amazon Elastic Compute Cloud (Amazon EC2) Linux utilizzando AWS AppConfig Agent. L'agente supporta x86\_64 e ARM64 architetture per Amazon EC2. Per ulteriori informazioni, consulta [AWS AppConfig Integrazione con Amazon EC2](#).

20 luglio 2023

## [CloudFormation supporto per nuove AWS AppConfig risorse e un esempio di feature flag](#)

AWS CloudFormation ora supporta le [AWS::AppConfig::ExtensionAssociation](#) risorse [AWS::AppConfig::Extension](#) e le risorse per aiutarti a iniziare con AWS AppConfig le estensioni.

12 aprile 2023

Le risorse [AWS::AppConfig::ConfigurationProfile](#) and [AWS::AppConfig::HostedConfigurationVersion](#) ora includono un esempio di creazione di un profilo di configurazione Feature Flag nell'archivio di configurazione AWS AppConfig ospitato.

## [AWS AppConfig integrazione con Gestione dei segreti AWS](#)

AWS AppConfig si integra con Gestione dei segreti AWS.

2 febbraio 2023

Secrets Manager consente di crittografare, archiviare e recuperare in modo sicuro le credenziali per i database e altri servizi. Invece di inserire le credenziali nelle app, puoi effettuare chiamate a Secrets Manager per recuperare le credenziali ogni volta che ne hai bisogno. Secrets Manager ti aiuta a proteggere l'accesso alle risorse e ai dati IT consentendoti di ruotare e gestire l'accesso ai tuoi segreti.

Quando si crea un profilo di configurazione in formato libero, è possibile scegliere Secrets Manager come origine dei dati di configurazione. È necessario effettuare l'onboarding con Secrets Manager e creare un segreto prima di creare il profilo di configurazione. Per ulteriori informazioni su Secrets Manager, vedi [Cos'è Gestione dei segreti AWS?](#) nella Guida Gestione dei segreti AWS per l'utente. Per informazioni sulla creazione di un profilo di configurazione, vedere [Creazione di un profilo di](#)

configurazione in formato  
libero.

## [AWS AppConfig integrazione con Amazon ECS e Amazon EKS](#)

Puoi effettuare l'integrazione AWS AppConfig con Amazon Elastic Container Service (Amazon ECS) e Amazon Elastic Kubernetes Service (Amazon EKS) utilizzando l'agente AWS AppConfig. L'agente funziona come un contenitore secondario che funziona insieme alle applicazioni container Amazon ECS e Amazon EKS. L'agente migliora l'elaborazione e la gestione delle applicazioni containerizzate nei seguenti modi:

- L'agente chiama AWS AppConfig per conto dell'utente utilizzando un ruolo AWS Identity and Access Management (IAM) e gestendo una cache locale di dati di configurazione. Estraendo i dati di configurazione dalla cache locale, l'applicazione richiede meno aggiornamenti del codice per gestire i dati di configurazione, recupera i dati di configurazione in millisecondi e non è interessata da problemi di rete che possono interrompere le chiamate per tali dati.

2 dicembre 2022

- L'agente offre un'esperienza nativa per il recupero e la risoluzione dei flag di funzionalità. AWS AppConfig
- Immediatamente, l'agente fornisce le migliori pratiche per le strategie di memorizzazione nella cache, gli intervalli di polling e la disponibilità dei dati di configurazione locali, tenendo traccia dei token di configurazione necessari per le successive chiamate di servizio.
- Durante l'esecuzione in background, l'agente analizza periodicamente il piano dati per verificare la presenza di aggiornamenti dei AWS AppConfig dati di configurazione. L'applicazione containerizzata può recuperare i dati connessi a localhost sulla porta 2772 (un valore di porta predefinito personalizzabile) e chiamando HTTP GET per recuperare i dati.
- L'AWS AppConfig agente aggiorna i dati di configurazione nei contenitori senza dover riavviare o riciclare tali contenitori.

Per ulteriori informazioni,  
consulta [AWS AppConfig](#)  
[l'integrazione con Amazon](#)  
[ECS e Amazon EKS.](#)

## [Nuova estensione: AWS AppConfig estensione per CloudWatch Evidently](#)

Puoi usare Amazon CloudWatch Evidently per convalidare nuove funzionalità in modo sicuro offrendole disponibili a una percentuale specifica di utenti durante il rollout della funzionalità. È possibile monitorare le prestazioni della nuova funzionalità per aiutarti a decidere quando aumentare il traffico verso gli utenti. Ciò consente di ridurre i rischi e identificare le conseguenze non intenzionali prima di avviare completamente la funzionalità. Puoi anche condurre A/B esperimenti per prendere decisioni sulla progettazione delle funzionalità sulla base di prove e dati.

L'AWS AppConfig estensione per CloudWatch Evidently consente all'applicazione di assegnare variazioni alle sessioni utente localmente anziché richiamare l'[EvaluateFeature](#) operazione. Una sessione locale mitiga i rischi di latenza e disponibilità associati a una chiamata API. Per informazioni su come configurare e utilizzare l'estensione, consulta [Esegui lanci ed A/B esperimenti con](#)

13 settembre 2022

[CloudWatch Evidently](#) nella Amazon CloudWatch User Guide.

[Deprecazione dell'azione dell'API GetConfiguration](#)

Il 18 novembre 2021, AWS AppConfig ha rilasciato un nuovo servizio di piano dati. Questo servizio sostituisce il precedente processo di recupero dei dati di configurazione utilizzando l'azione API. `GetConfiguration` Il servizio data plane utilizza due nuove azioni API e. [StartConfigurationSession](#) [GetLatestConfiguration](#) Il servizio data plane utilizza anche [nuovi endpoint](#).

13 settembre 2022

Per ulteriori informazioni, vedere [Informazioni sul servizio AWS AppConfig Data Plane](#).

[Nuova versione dell'estensione AWS AppConfig Agent Lambda](#)

La versione 2.0.122 dell'estensione AWS AppConfig Agent Lambda è ora disponibile. La nuova estensione utilizza diversi Amazon Resource Names (ARNs). Per ulteriori informazioni, vedere le note di [rilascio dell'estensione AWS AppConfig Agent Lambda](#).

23 agosto 2022

<a href="#"><u>Lancio delle estensioni AWS AppConfig</u></a>	Un'estensione aumenta la capacità di inserire logica o comportamento in punti diversi durante il AWS AppConfig flusso di lavoro di creazione o distribuzione di una configurazione. Puoi usare estensioni AWS create da te o crearne di tue. Per ulteriori informazioni, consulta <a href="#"><u>Lavorare con AWS AppConfig</u></a> le estensioni.	12 luglio 2022
<a href="#"><u>Nuova versione dell'estensione AWS AppConfig Agent Lambda</u></a>	La versione 2.0.58 dell'estensione AWS AppConfig Agent Lambda è ora disponibile. La nuova estensione utilizza diversi Amazon Resource Names (ARNs). Per ulteriori informazioni, consulta <a href="#"><u>Versioni disponibili dell'estensione AWS AppConfig Lambda</u></a> .	3 maggio 2022

## [AWS AppConfig integrazione con Atlassian Jira](#)

L'integrazione con Atlassian Jira consente di AWS AppConfig creare e aggiornare i problemi nella console Atlassian ogni volta che si apportano modifiche a un feature flag nel modulo specificato. Account AWS Regione AWS Ogni numero di Jira include il nome del flag, l'ID dell'applicazione, l'ID del profilo di configurazione e i valori dei flag. Dopo aver aggiornato, salvato e distribuito le modifiche ai flag, Jira aggiorna i problemi esistenti con i dettagli della modifica. Per ulteriori informazioni, consulta [AWS AppConfig Integrazione con Atlassian Jira](#).

7 aprile 2022

[Disponibilità generale dei flag delle funzionalità e del supporto dell'estensione Lambda ARM64 per i processori \(Graviton2\)](#)

Con i flag di AWS AppConfig funzionalità, puoi sviluppar e una nuova funzionalità e distribuirla in produzione nascondendola agli utenti. Iniziate aggiungendo il flag ai dati di configurazione AWS AppConfig . Una volta che la funzionalità è pronta per essere rilasciata, puoi aggiornare i dati di configura zione del flag senza distribui re alcun codice. Questa funzionalità migliora la sicurezza dell'ambiente dev-ops perché non è necessari o distribuire nuovo codice per rilasciare la funzionalità. Per ulteriori informazioni, consulta [Creazione di un profilo di configurazione del feature flag.](#)

15 marzo 2022

La disponibilità generale dei flag di funzionalità AWS AppConfig include i seguenti miglioramenti:

- La console include un'opzione per designare una bandiera come bandiera a breve termine. È possibile filtrare e ordinare l'elenco delle bandiere in base alle bandiere a breve termine.
- Per i clienti che utilizzano i flag di funzionalità in AWS Lambda, la nuova estension

e Lambda consente di richiamare i singoli flag di funzionalità utilizzando un endpoint HTTP. Per ulteriori informazioni, consulta [Recupero di uno o più flag da una configurazione di feature flag](#).

Questo aggiornamento fornisce anche il supporto per AWS Lambda le estensioni sviluppate per i processori ARM64 (Graviton2). Per ulteriori informazioni, consulta [Versioni disponibili dell'estensione AWS AppConfig Lambda](#).

### [L'azione GetConfiguration API è obsoleta](#)

L'azione GetConfiguration API è obsoleta. Le chiamate per ricevere i dati di configurazione devono invece utilizzare il StartConfigurationSession comando and. GetLatest Configuration APIs Per ulteriori informazioni su questi APIs e su come utilizzarli, vedi [Recupero della configurazione](#).

28 gennaio 2022

<a href="#"><u>Nuova regione ARN per l'estensione Lambda AWS AppConfig</u></a>	AWS AppConfig L'estensione Lambda è disponibile nella nuova regione Asia Pacifico (Osaka). L'Amazon Resource Name (ARN) è necessario per creare una Lambda nella regione. Per ulteriori informazioni sull'ARN della regione Asia Pacifico (Osaka), <a href="#"><u>vedi Aggiungere l'estensione AWS AppConfig Lambda.</u></a>	4 marzo 2021
<a href="#"><u>AWS AppConfig Estensione Lambda</u></a>	Se utilizzi AWS AppConfig per gestire le configurazioni per una funzione Lambda, ti consigliamo di aggiungere l'estensione AWS AppConfig Lambda. Questa estensione include le migliori pratiche che semplificano l'utilizzo riducendo al AWS AppConfig contempo i costi. I costi ridotti derivano da un minor numero di chiamate API al AWS AppConfig servizio e, separatamente, dalla riduzione dei costi grazie ai tempi di elaborazione delle funzioni Lambda più brevi. Per ulteriori informazioni, consulta <a href="#"><u>AWS AppConfig Integrazione con le estensioni Lambda.</u></a>	8 ottobre 2020

<a href="#"><u>Nuova sezione</u></a>	È stata aggiunta una nuova sezione che fornisce istruzioni per la configurazione AWS AppConfig. Per ulteriori informazioni, vedere <a href="#"><u>Configurazione AWS AppConfig</u></a> .	30 settembre 2020
<a href="#"><u>Sono state aggiunte procedure da riga di comando</u></a>	Le procedure di questa guida per l'utente ora includono i passaggi della riga di comando per AWS Command Line Interface (AWS CLI) e Tools for Windows. PowerShell. Per ulteriori informazioni, vedere <a href="#"><u>Lavorare con AWS AppConfig</u></a>	30 settembre 2020
<a href="#"><u>Lancio della guida AWS AppConfig per l'utente</u></a>	Utilizza AWS AppConfig, uno strumento per creare AWS Systems Manager, gestire e distribuire rapidamente le configurazioni delle applicazioni. AWS AppConfig supporta implementazioni controllate su applicazioni di qualsiasi dimensione e include controlli di convalida e monitoraggio integrati. Puoi utilizzarlo AWS AppConfig con applicazioni ospitate su EC2 istanze AWS Lambda, contenitori, applicazioni mobili o dispositivi IoT.	31 luglio 2020

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.