



Guida per gli sviluppatori

AWS Key Management Service



AWS Key Management Service: Guida per gli sviluppatori

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 discrediti 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

AWS Key Management Service	1
Perché usare AWS KMS?	1
AWS KMS in Regioni AWS	2
AWS KMS prezzi	2
AWS KMS contratto sul livello di servizio	2
Accedendo AWS Key Management Service	3
Console di gestione AWS	3
Autorizzazioni necessarie per utilizzare la console AWS KMS	3
AWS Command Line Interface	4
AWS KMS REST API	4
AWS SDKs	4
Lavorare con AWS SDKs	5
AWS Encryption SDK	6
AWS KMS eventuale coerenza	6
Protocollo TLS post-quantistico ibrido	7
Informazioni sul protocollo TLS post quantistico	8
Come utilizzarlo	9
Configura il TLS post-quantistico ibrido	9
Ulteriori informazioni	11
Connect a AWS KMS tramite un endpoint VPC	12
Crea un endpoint VPC per AWS KMS	13
Connect a un endpoint VPC	14
Usa gli endpoint VPC per controllare l'accesso alle risorse AWS KMS	15
Registrazione delle AWS KMS richieste che utilizzano un endpoint VPC	19
Endpoint dual-stack	20
Funzionalità non disponibili su IPv6	21
Concetti	22
Introduzione	22
Obiettivi di progettazione di	23
AWS KMS keys	24
Chiavi gestite dal cliente	27
Chiavi gestite da AWS	28
Chiavi di proprietà di AWS	29
AWS KMS key gerarchia	30

Identificatori chiave () KeyId	32
Chiavi asimmetriche	35
Chiavi HMAC	37
Chiavi ML-DSA	39
Chiavi multi-regione	40
Materiale della chiave importato	51
Chiavi KMS in un archivio di chiavi CloudHSM	58
Chiavi KMS negli archivi di chiavi esterni	61
AWS KMS elementi essenziali della crittografia	64
Entropia e generazione di numeri casuali	64
Operazioni con chiavi simmetriche (solo crittografia)	64
Operazioni con chiave asimmetrica (crittografia, firma digitale e verifica della firma)	65
Funzioni di derivazione chiave	66
AWS KMS uso interno delle firme digitali	66
Crittografia envelope	66
Operazioni di crittografia	68
Algoritmi supportati	70
algoritmi crittografici	70
Accesso e autorizzazioni alle chiavi KMS	74
Politiche chiave KMS	74
Concessioni chiave KMS	75
Policy delle chiavi	76
Creazione di una policy delle chiavi	76
Policy delle chiavi predefinita	84
Visualizza una politica chiave	100
Modificare una policy della chiave	103
Autorizzazioni per i servizi AWS	106
Policy IAM	107
Autorizzazione per più principali IAM di accedere a una chiave KMS	109
Best practice per le policy IAM	109
Specificazione delle chiavi KMS nelle istruzioni della policy IAM	112
Esempi	115
Politiche di controllo delle risorse	122
Concessioni	125
Concetti delle concessioni	126
Best practice	131

Controllo dell'accesso alle concessioni	133
Creazione di concessioni	134
Visualizzazione di concessioni	143
Utilizzo di un token di concessione	144
Ritirare e revocare le concessioni	145
Chiavi di condizione	146
AWS chiavi di condizione globali	147
AWS KMS tasti di condizione	150
AWS KMS chiavi di condizione per piattaforme attestate	220
Autorizzazioni con privilegi minimi	226
Implementazione di autorizzazioni con privilegio minimo	227
Controllo degli accessi basato su attributi (ABAC)	231
Chiavi di condizione ABAC per AWS KMS	232
Tag o alias?	235
Risoluzione dei problemi ABAC per AWS KMS	237
Controllo degli accessi basato sui ruoli (RBAC)	241
Accesso multi-account	243
Fase 1: aggiungere una dichiarazione di policy delle chiavi nell'account locale	246
Fase 2: aggiungere le policy IAM nell'account esterno	249
Autorizzazione per l'utilizzo di chiavi KMS esterne con Servizi AWS	251
Utilizzo delle chiavi KMS in altri account	251
Controlla l'accesso alle chiavi multiregionali	252
Nozioni di base sull'autorizzazione per le chiavi multi-regione	253
Autorizzazione degli amministratori e degli utenti delle chiavi multi-regione	255
Determinazione dell'accesso	259
Analisi della policy delle chiavi	260
Analisi delle policy IAM	263
Analisi delle concessioni	265
Contesto di crittografia	266
Regole sul contesto di crittografia	267
Contesto di crittografia nelle policy	268
Contesto di crittografia nelle concessioni	269
Registrazione del contesto di crittografia	270
Archiviazione del contesto di crittografia	270
Test delle autorizzazioni	270
Che cos'è DryRun?	271

Specificazione DryRun con l'API	272
Risoluzione dei problemi relativi alle autorizzazioni AWS KMS	272
Esempio 1: all'utente viene negato l'accesso a una chiave KMS nel proprio Account AWS ..	275
Esempio 2: L'utente assume un ruolo con il permesso di utilizzare una chiave KMS in un altro Account AWS	277
Glossario	281
Autenticazione	281
Autorizzazione	281
Autenticazione con identità	281
Gestione dell'accesso tramite policy	283
AWS KMS risorse	284
Crea una chiave KMS	286
Autorizzazioni per la creazione di chiavi KMS	288
Scelta del tipo di chiave KMS da creare	289
Creazione di una chiave KMS di crittografia simmetrica	292
Creazione di una chiave KMS asimmetrica	297
Creazione di una chiave KMS HMAC	303
Crea chiavi primarie multiregionali	308
Crea chiavi di replica multiregionali	314
Passaggio 1: Scegli le regioni di replica	314
Fase 2: Creare chiavi di replica	315
Crea una chiave KMS con materiale chiave importato	321
Autorizzazioni per l'importazione del materiale della chiave	322
Requisiti per il materiale della chiave importato	324
Fase AWS KMS key 1: Creare un materiale senza chiave	327
Fase 2: download della chiave pubblica di wrapping e del token di importazione	330
Fase 3: crittografare il materiale delle chiavi	339
Fase 4: importare il materiale delle chiavi	348
Crea una chiave KMS in un AWS CloudHSM archivio di chiavi	356
Crea una nuova chiave KMS nel tuo key store CloudHSM	357
Crea una chiave KMS in archivi di chiavi esterni	363
Requisiti per una chiave KMS in un archivio delle chiavi esterne	365
Crea una nuova chiave KMS nel tuo archivio di chiavi esterno	366
Identifica e visualizza le chiavi	375
Trova l'ID della chiave e l'ARN della chiave	375
Accedi ed elenca i dettagli chiave KMS	377

Identifica diversi tipi di chiave	387
Identifica le chiavi KMS asimmetriche	387
Identifica le chiavi HMAC KMS	388
Identifica le chiavi KMS multiregionali	389
Identifica le chiavi KMS con materiale chiave importato	390
Identifica le chiavi KMS negli AWS CloudHSM archivi delle chiavi	391
Identifica le chiavi KMS negli archivi di chiavi esterni	391
Personalizza la visualizzazione della console	392
Ordina e filtra le tue chiavi KMS	393
Personalizza le tue tabelle chiave KMS	396
Trova le chiavi KMS e il materiale chiave in un archivio di AWS CloudHSM chiavi	398
Trova le chiavi KMS in un AWS CloudHSM archivio di chiavi	399
Trova tutte le chiavi di un archivio di AWS CloudHSM chiavi	400
Trova la chiave KMS per una chiave AWS CloudHSM	402
Trova la AWS CloudHSM chiave per una chiave KMS	407
Attivazione e disattivazione delle chiavi	410
Rotazione delle chiavi	413
Perché ruotare le chiavi KMS?	415
Come funziona la rotazione dei tasti	416
Abilita la rotazione automatica dei tasti	420
Disabilita la rotazione automatica delle chiavi	423
Esegui la rotazione dei tasti su richiesta	425
Avvio della rotazione dei tasti su richiesta (console)	426
Avvio della rotazione delle chiavi su richiesta (API)AWS KMS	427
Elenca le rotazioni e i materiali chiave	428
Elenca le rotazioni e i materiali chiave (console)	428
Rotazioni degli elenchi e materiali chiave (API)AWS KMS	429
Ruota i tasti manualmente	430
Cambia la chiave primaria in un set di chiavi multiregionali	433
Aggiorna la regione principale	435
Eliminazione delle chiavi	438
Informazioni sul periodo di attesa	439
Considerazioni speciali	439
Controlla l'accesso all'eliminazione delle chiavi	443
Consenti agli amministratori delle chiavi di pianificare e annullare l'eliminazione delle chiavi	443

Pianifica l'eliminazione delle chiavi	446
.....	446
Annula l'eliminazione della chiave	447
Creazione di un allarme	449
Determina l'utilizzo passato di una chiave KMS	451
Esamina le autorizzazioni delle chiavi KMS per determinare l'ambito del potenziale utilizzo .	451
Esamina AWS CloudTrail i log per determinare l'utilizzo effettivo	451
Eliminare il materiale chiave importato	454
Generazione di chiavi dati	458
Crea una chiave di chiavi	458
Come funzionano le operazioni crittografiche con chiavi dati	459
Crittografia dei dati con una chiave di dati	459
Decrittografare i dati con una chiave di dati	460
In che modo le chiavi KMS inutilizzabili influiscono sulle chiavi dati	461
Genera coppie di chiavi di dati	463
Creare una coppia di chiave di dati	463
Come funzionano le operazioni crittografiche con coppie di chiavi di dati	464
Crittografia dei dati con una coppia di chiavi di dati	465
Decrittografia dei dati con una coppia di chiave di dati	465
Firmare messaggi con una coppia di chiavi di dati	466
Verificare una firma con una coppia di chiavi di dati	467
Ricava un segreto condiviso con coppie di chiavi di dati	468
Esegui operazioni offline con chiavi pubbliche	469
Considerazioni speciali per il download delle chiavi pubbliche	470
Scarica la chiave pubblica	471
Esempi di operazioni offline	473
Ricevere segreti condivisi offline	473
Verifica offline con coppie di chiavi ML-DSA	474
Verifica offline con coppie di SM2 chiavi (solo regioni della Cina)	477
Tasti del monitor	483
Strumenti di monitoraggio	484
Strumenti automatici	484
Strumenti manuali	484
Registrazione con AWS CloudTrail	485
Ricerca delle voci di AWS KMS registro in CloudTrail	486
Esclusione di AWS KMS eventi da un percorso	488

Esempi di voci di AWS KMS registro	489
Monitora le chiavi con CloudWatch	574
AWS KMS metriche e dimensioni	575
Crea un CloudWatch allarme per la scadenza del materiale chiave importato	584
Crea CloudWatch allarmi per archivi di chiavi esterni	585
Monitora le chiavi con Amazon EventBridge	590
Rotazione KMS CMK	590
Scadenza del materiale chiave importato di KMS	591
Eliminazione di KMS CMK	592
Alias	593
Come funzionano gli alias	593
Controllo dell'accesso agli alias	597
kms:CreateAlias	597
kms:ListAliases	598
kms:UpdateAlias	599
kms>DeleteAlias	600
Limitazione delle autorizzazioni alias	601
Creare alias	603
Trova il nome dell'alias e l'alias ARN	605
Aggiornare gli alias	610
Eliminare un alias	611
Usa gli alias per controllare l'accesso alle chiavi KMS	612
kms:RequestAlias	614
kms:ResourceAliases	615
Scopri come utilizzare gli alias nelle tue applicazioni	616
Trova gli alias nei log AWS CloudTrail	618
Tag	620
Controllo degli accessi ai tag	621
Autorizzazioni ad assegnare tag nelle policy	622
Limitazione delle autorizzazioni ad assegnare tag	624
Aggiungi tag	626
Aggiungere tag durante la creazione di una chiave KMS	626
Aggiungi tag alle chiavi KMS esistenti	628
Modifica i tag	629
Rimuovi i tag	630
Visualizzazione dei tag	631

Usa i tag per controllare l'accesso alle chiavi KMS	633
Negozi chiave	638
AWS KMS archivio chiavi standard	638
AWS KMS archivio chiavi standard con materiale chiave importato	638
AWS KMS archivi di chiavi personalizzati	640
AWS CloudHSM archivio chiavi	641
Archivio delle chiavi esterne	641
AWS CloudHSM negozi chiave	641
AWS CloudHSM concetti di archivio chiave	645
Controlla l'accesso al tuo AWS CloudHSM key store	648
Crea un archivio di AWS CloudHSM chiavi	649
Visualizza un archivio di AWS CloudHSM chiavi	656
Modifica le impostazioni del AWS CloudHSM key store	660
Connect a AWS CloudHSM key store	663
Disconnetti un archivio AWS CloudHSM chiavi	667
Eliminare un archivio AWS CloudHSM chiavi	671
Risoluzione di problemi relativi a store delle chiavi personalizzate	673
Archivi delle chiavi esterne	689
Concetti fondamentali sull'archivio delle chiavi esterne	694
Funzionamento degli archivi delle chiavi esterne	703
Controlla l'accesso al tuo archivio di chiavi esterno	705
Scegli un'opzione di connettività proxy	711
Creare un archivio di chiavi esterno	724
Modifica delle proprietà dell'archivio chiavi esterno	738
Visualizza gli archivi di chiavi esterni	745
Monitora gli archivi di chiavi esterni	751
Connect e disconnetti gli archivi di chiavi esterni	764
Eliminare un archivio chiavi esterno	774
Risoluzione dei problemi relativi all'archivio delle chiavi esterne	776
Sicurezza	803
Protezione dei dati	803
Protezione del materiale della chiave	804
Crittografia dei dati	805
Riservatezza di Internet	807
Gestione dell'identità e degli accessi	808
AWS politiche gestite	808

Ruoli collegati ai servizi	812
Registrazione di log e monitoraggio	818
Convalida della conformità	819
Documenti di conformità e sicurezza	820
Ulteriori informazioni	820
Resilienza	821
Isolamento regionale	821
Design multi-tenant	822
Le migliori pratiche di resilienza in AWS KMS	822
Sicurezza dell'infrastruttura	823
Isolamento di host fisici	824
Quote	826
Quote delle risorse	826
AWS KMS keys: 100.000	827
Alias per chiave KMS: 50	827
Concessioni per chiave KMS: 50.000	828
Quote di risorse per gli archivi delle chiavi personalizzate: 10	828
Rotazione su richiesta: 10	828
Quote di richieste	829
Richiedi quote per ogni operazione API AWS KMS	829
Applicazione delle quote di richieste	836
Quote condivise per le operazioni di crittografia	837
Richieste API eseguite per tuo conto	839
Richieste tra account	839
Quote di richiesta per l'archivio delle chiavi personalizzate	839
Limitazione delle richieste	841
Esempi di codice	843
Nozioni di base	847
Ciao AWS KMS	848
Informazioni di base	851
Azioni	925
Scenari	1076
Utilizzo della crittografia delle tabelle	1076
Supporto per l'attestazione crittografica in AWS KMS	1079
Come effettuare chiamate certificate a AWS KMS	1081
Monitoraggio delle richieste attestate	1081

Richieste di monitoraggio per enclavi Nitro	1082
Richieste di monitoraggio per NitroTPM	1087
Servizi di crittografia AWS	1092
Amazon Elastic Block Store (Amazon EBS)	1093
Crittografia Amazon EBS	1093
Utilizzo di chiavi KMS e chiavi dati	1094
Contesto di crittografia di Amazon EBS	1094
Rilevamento degli errori Amazon EBS	1095
Utilizzo AWS CloudFormation per creare volumi Amazon EBS crittografati	1096
Amazon EMR	1096
Crittografia dei dati su EMR File System (EMRFS)	1097
Crittografia dei dati su volumi di storage di nodi cluster	1099
Contesto di crittografia	1100
Amazon Redshift	1101
Crittografia di Amazon Redshift	1102
Contesto di crittografia	1103
Documentazione di riferimento	1104
Riferimento degli stati chiave	1105
Stato chiave e tipi di chiave KMS	1105
Tabella dello stato delle chiavi	1106
Documentazione di riferimento dei tipi di chiave	1115
Tabella dei tipi di chiave	1115
Tabella delle caratteristiche speciali	1124
Riferimento alle specifiche chiave	1132
Specifiche della chiave SYMMETRIC_DEFAULT	1134
Specifiche della chiave RSA	1135
Specifiche della chiave basata su curva ellittica	1140
Specifiche della chiave per le chiavi KMS HMAC	1142
Specifiche principali di ML-DSA	1143
SM2 specifiche chiave (solo regioni della Cina)	1143
Riferimento per le autorizzazioni	1144
Descrizioni delle colonne	1192
AWS KMS operazioni interne	1194
Domini e stato del dominio	1195
Sicurezza delle comunicazioni interne	1199
Processo di replica per chiavi multi-regione	1202

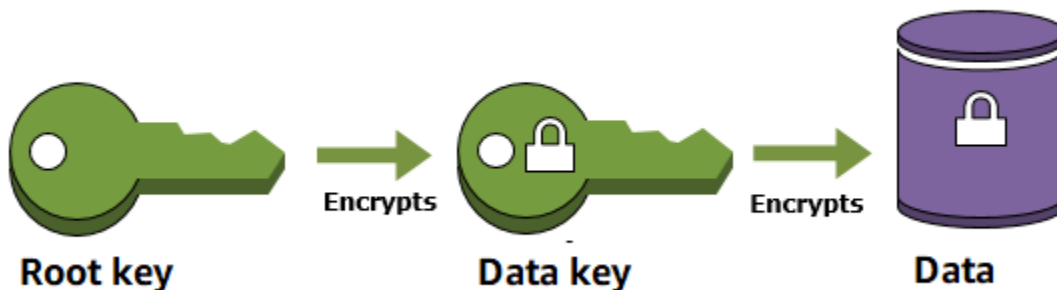
Protezione della durabilità	1203
Cronologia dei documenti	1204
Aggiornamenti recenti	1204
Aggiornamenti precedenti	1211
.....	mccxvi

AWS Key Management Service

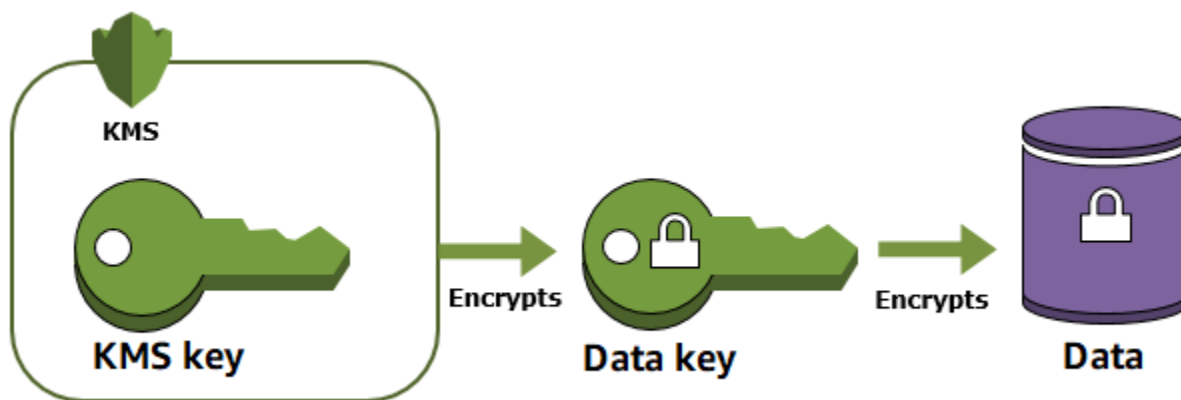
AWS Key Management Service (AWS KMS) è un servizio AWS gestito che semplifica la creazione e il controllo delle chiavi utilizzate per crittografare e firmare i dati. I AWS KMS keys file creati AWS KMS sono protetti da [moduli di sicurezza hardware \(HSM\) convalidati FIPS 140-3 di livello di sicurezza 3](#). Non escono mai senza crittografia. AWS KMS Per utilizzare o gestire le tue chiavi KMS, interagisci con. AWS KMS

Perché usare AWS KMS?

Quando si crittografano i dati, è necessario proteggere la chiave di crittografia. Se si crittografa la chiave, è necessario proteggere la relativa chiave di crittografia. Alla fine, è necessario proteggere la chiave di crittografia di livello più alto (nota come chiave radice) nella gerarchia che protegge i dati. È qui che AWS KMS entra in gioco.



AWS KMS protegge le chiavi principali. Le chiavi KMS vengono create, gestite, utilizzate ed eliminate interamente all'interno AWS KMS. Non lasciano mai il servizio non crittografato. Per utilizzare o gestire le tue chiavi KMS, chiama. AWS KMS



Inoltre, puoi creare e gestire [le politiche chiave](#) in AWS KMS, assicurando che solo gli utenti fidati abbiano accesso alle chiavi KMS.

AWS KMS in Regioni AWS

Le Regioni AWS funzionalità supportate AWS KMS sono elencate in [AWS Key Management Service Endpoints e Quotas](#). Se una AWS KMS funzionalità non è supportata da una funzionalità Regione AWS che la AWS KMS supporta, la differenza regionale è descritta nell'argomento relativo alla funzionalità.

AWS KMS prezzi

Come per altri AWS prodotti, l'utilizzo AWS KMS non richiede contratti o acquisti minimi. Per ulteriori informazioni sui AWS KMS prezzi, consulta la sezione [AWS Key Management Service Prezzi](#).

AWS KMS contratto sul livello di servizio

AWS Key Management Service è supportato da un [accordo sul livello di servizio](#) che definisce la nostra politica di disponibilità del servizio.

Accedendo AWS Key Management Service

È possibile lavorare con AWS KMS nei seguenti modi:

Argomenti

- [Console di gestione AWS](#)
- [AWS Command Line Interface](#)
- [AWS KMS REST API](#)
- [AWS SDKs](#)
- [Utilizzo di questo servizio con un AWS SDK](#)
- [AWS Encryption SDK](#)
- [AWS KMS eventuale coerenza](#)
- [Utilizzo del TLS post-quantistico ibrido con AWS KMS](#)
- [Connect a AWS KMS tramite un endpoint VPC](#)
- [Supporto per endpoint dual-stack](#)

Console di gestione AWS

La console è un'interfaccia utente basata sul Web per la gestione AWS KMS e AWS le risorse. Se ti sei registrato a Account AWS, puoi accedere alla AWS KMS console accedendo Console di gestione AWS e selezionando AWS KMS dalla Console di gestione AWS home page.

Autorizzazioni necessarie per utilizzare la console AWS KMS

Per utilizzare la AWS KMS console, gli utenti devono disporre di un set minimo di autorizzazioni che consentano loro di utilizzare le AWS KMS risorse disponibili. Account AWS Oltre a queste autorizzazioni AWS KMS , gli utenti devono disporre delle autorizzazioni per elencare gli utenti IAM e i ruoli IAM. Se decidi di creare una policy IAM più restrittiva delle autorizzazioni minime richieste, la console AWS KMS non funzionerà come previsto per gli utenti con tale policy IAM.

Per le autorizzazioni minime necessarie per consentire a un utente l'accesso in sola lettura alla console AWS KMS , consulta [Consenti a un utente di visualizzare le chiavi KMS nella console AWS KMS](#).

Per consentire agli utenti di utilizzare la AWS KMS console per creare e gestire le chiavi KMS, allega la policy `AWSKeyManagementServicePowerUser` gestita all'utente, come descritto in [AWS politiche gestite per AWS Key Management Service](#)

Non è necessario consentire autorizzazioni minime per la console agli utenti che utilizzano l' AWS KMS API tramite [AWS SDKsAWS Command Line Interface](#), o [AWS Strumenti per PowerShell](#). Tuttavia, è necessario concedere a questi utenti l'autorizzazione per utilizzare l'API. Per ulteriori informazioni, consulta [Riferimento per le autorizzazioni](#).

AWS Command Line Interface

È possibile utilizzare gli AWS CLI strumenti per impartire comandi o creare script dalla riga di comando del sistema per eseguire AWS (incluse AWS KMS) attività.

Per ulteriori informazioni sull'utilizzo AWS KMS tramite AWS CLI, consulta la Guida di riferimento ai [AWS CLI comandi](#)

AWS KMS REST API

L'architettura di AWS KMS è progettata per essere indipendente dal linguaggio di programmazione. REST APIÈ un'interfaccia HTTP per. AWS KMS ConREST API, si utilizzano richieste HTTP standard per creare, recuperare ed eliminare le chiavi.

Per ulteriori informazioni sull'utilizzo di AWS KMS REST API, consulta l'[AWS Key Management Service API Reference](#)

AWS SDKs

AWS fornisce SDKs (kit di sviluppo software) costituiti da librerie e codice di esempio per linguaggi e piattaforme di programmazione comuni (Java JavaScript, C, Python e così via). AWS SDKs Forniscono un modo conveniente per creare l'accesso programmatico a e. AWS KMS AWS KMS è un REST servizio. È possibile inviare richieste AWS KMS utilizzando le librerie AWS SDK, che racchiudono le informazioni sottostanti AWS KMS REST API e semplificano le attività di programmazione. Per informazioni su AWS SDKs, incluso come scaricarle e installarle, consulta [Tools to Build on AWS](#).

[Esempi di codice per AWS KMS l'utilizzo AWS SDKs](#)Fornisce un buon punto di partenza per l'utilizzo AWS KMS tramite AWS SDKs.

Utilizzo di questo servizio con un AWS SDK

AWS i kit di sviluppo software (SDKs) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice, e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

Documentazione sugli SDK	Esempi di codice
AWS SDK per C++	AWS SDK per C++ esempi di codice
AWS CLI	AWS CLI esempi di codice
AWS SDK per Go	AWS SDK per Go esempi di codice
AWS SDK per Java	AWS SDK per Java esempi di codice
AWS SDK per JavaScript	AWS SDK per JavaScript esempi di codice
AWS SDK per Kotlin	AWS SDK per Kotlin esempi di codice
AWS SDK per .NET	AWS SDK per .NET esempi di codice
AWS SDK per PHP	AWS SDK per PHP esempi di codice
AWS Strumenti per PowerShell	AWS Strumenti per PowerShell esempi di codice
AWS SDK per Python (Boto3)	AWS SDK per Python (Boto3) esempi di codice
AWS SDK per Ruby	AWS SDK per Ruby esempi di codice
AWS SDK for Rust	AWS SDK for Rust esempi di codice
SDK AWS per SAP ABAP	SDK AWS per SAP ABAP esempi di codice
SDK AWS per Swift	SDK AWS per Swift esempi di codice

Esempio di disponibilità

Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link Fornisci un feedback nella parte inferiore di questa pagina.

AWS Encryption SDK

AWS Encryption SDK è uno strumento per implementare la crittografia lato client nell'applicazione. Non fornisce l'accesso completo a KMS, ma si integra o può essere utilizzato come AWS KMS SDK autonomo senza fare riferimento alle chiavi KMS. Le librerie sono disponibili per Java JavaScript, C, Python e altri linguaggi di programmazione.

Per ulteriori informazioni, consulta la [Guida per gli sviluppatori di AWS Encryption SDK](#).

AWS KMS key politiche e politiche IAM

AWS KMS eventuale coerenza

L' AWS KMS API segue un [eventuale modello di coerenza](#) dovuto alla natura distribuita del sistema. Di conseguenza, le modifiche alle AWS KMS risorse potrebbero non essere immediatamente visibili ai comandi successivi eseguiti.

Quando si eseguono chiamate AWS KMS API, potrebbe verificarsi un breve ritardo prima che la modifica sia disponibile per tutta la durata AWS KMS. Generalmente, occorrono pochissimi secondi per la propagazione della modifica in tutto il sistema, ma in alcuni casi possono essere necessari diversi minuti. Durante questo periodo, potrebbero verificarsi errori imprevisti, ad esempio `NotFoundException` o `InvalidStateException`. Ad esempio, AWS KMS potrebbe restituire un messaggio `NotFoundException` se si chiama `GetParametersForImport` subito dopo la chiamata `CreateKey`.

Ti consigliamo di configurare una strategia di ripetizione dei tentativi sui tuoi AWS KMS client per riprovare automaticamente le operazioni dopo un breve periodo di attesa. Per ulteriori informazioni, consulta [Retry behavior nella AWS SDKs and Tools Reference](#) Guide.

Per le chiamate API correlate all'autorizzazione, puoi [usare un token di concessione](#) per evitare potenziali ritardi e utilizzare subito le autorizzazioni incluse in una concessione. Per ulteriori informazioni, consulta [Consistenza finale \(per concessioni\)](#).

Utilizzo del TLS post-quantistico ibrido con AWS KMS

AWS Key Management Service (AWS KMS) supporta un'opzione ibrida di scambio di chiavi post-quantistiche per il protocollo di crittografia di rete Transport Layer Security (TLS). È possibile utilizzare questa opzione TLS quando ci si connette agli endpoint API AWS KMS. Queste caratteristiche opzionali di scambio di chiavi post-quantistiche ibride sono sicure almeno quanto la crittografia TLS che utilizziamo oggi e potrebbero fornire ulteriori vantaggi per la sicurezza a lungo termine. Tuttavia, influenzano la latenza e il throughput rispetto ai protocolli di scambio di chiavi classici in uso oggi.

I dati inviati a AWS Key Management Service (AWS KMS) sono protetti in transito dalla crittografia fornita da una connessione Transport Layer Security (TLS). Le classiche suite di crittografia supportate da AWS KMS per le sessioni TLS rendono gli attacchi di forza bruta sui meccanismi di scambio delle chiavi irrealizzabili con la tecnologia attuale. Tuttavia, se il calcolo quantistico su larga scala diventa una realtà in futuro, le classiche suite di crittografia utilizzate nei meccanismi di scambio delle chiavi TLS saranno suscettibili a questi attacchi. Se state sviluppando applicazioni che si basano sulla riservatezza a lungo termine dei dati trasmessi tramite una connessione TLS, dovrete prendere in considerazione un piano di migrazione alla crittografia post-quantistica prima che i computer quantistici su larga scala diventino disponibili per l'uso. AWS sta lavorando per prepararsi a questo futuro e vogliamo che anche voi siate ben preparati.

Per proteggere i dati crittografati oggi da potenziali attacchi futuri, AWS partecipa con la comunità crittografica allo sviluppo di algoritmi quantistici resistenti o post-quantistici. Abbiamo implementato suite di crittografia ibride post-quantistiche a scambio di chiavi AWS KMS che combinano elementi classici e post-quantistici per garantire che la connessione TLS sia almeno altrettanto potente come lo sarebbe con le suite di crittografia classiche.

[Queste suite di crittografia ibride sono disponibili per l'uso sui carichi di lavoro di produzione nella maggior parte dei casi. Regioni AWS](#) Tuttavia, poiché le caratteristiche prestazionali e i requisiti di larghezza di banda delle suite di crittografia ibride sono diversi da quelli dei classici meccanismi di scambio di chiavi, consigliamo di [testarli sulle](#) chiamate API in condizioni diverse. AWS KMS

Feedback

Come sempre, la tua opinione e la partecipazione nei nostri repository open source è molto importante. Vorremmo soprattutto sapere come la tua infrastruttura interagisce con questa nuova variante del traffico TLS.

- Per fornire un feedback su questo argomento, usa il link Feedback nell'angolo in alto a destra di questa pagina.
- Stiamo sviluppando queste suite di crittografia ibrida in open source nel repository di [s2n-tls](#) GitHub. Per fornire feedback sulla fruibilità delle suite crittografia o condividere nuove condizioni o risultati di test, [creare un problema](#) nel repository s2n-tls.
- Stiamo scrivendo esempi di codice per l'utilizzo del TLS post-quantistico ibrido nel repository. AWS KMS [aws-kms-pq-tls-example](#) GitHub [Per porre domande o condividere idee sulla configurazione del client o AWS KMS del client HTTP per l'utilizzo delle suite di crittografia ibrida, crea un problema nel repository.](#) [aws-kms-pq-tls-example](#)

Supportato Regioni AWS

Post-quantum TLS for AWS KMS è disponibile in tutti i Regioni AWS supporti. AWS KMS

Per un elenco di AWS KMS endpoint per ciascuno Regione AWS, consulta [AWS Key Management Service endpoint](#) e quote in. Riferimenti generali di Amazon Web Services. Per ulteriori informazioni sugli endpoint FIPS, consulta [Endpoint FIPS](#) nella Riferimenti generali di Amazon Web Services.

Informazioni sullo scambio di chiavi post-quantistiche ibride in TLS

AWS KMS supporta suite di cifratura ibride post-quantistiche a scambio di chiavi. È possibile utilizzare AWS SDK for Java 2.x e AWS Common Runtime sui sistemi Linux per configurare un client HTTP che utilizza queste suite di crittografia. Quindi, ogni volta che ci si connette a un AWS KMS endpoint con il client HTTP, vengono utilizzate le suite di crittografia ibride.

Questo client HTTP utilizza [s2n-tls](#), che è un'implementazione open source del protocollo TLS. Le suite di crittografia ibride utilizzate da s2n-tls sono implementate solo per lo scambio di chiavi, non per la crittografia dei dati diretta. Durante lo scambio di chiavi, il client e il server calcolano la chiave che utilizzeranno per crittografare e decrittografare i dati in rete.

Gli algoritmi s2n-tls utilizzati sono un ibrido che combina [Elliptic Curve Diffie-Hellman](#) (ECDH), un classico algoritmo di scambio di chiavi utilizzato oggi in TLS, con il [meccanismo di incapsulamento delle chiavi basato su Module-Lattice](#) (ML-KEM), un algoritmo di crittografia e definizione delle chiavi a chiave pubblica che il National Institute for Standards and Technology (NIST) [ha designato come primo algoritmo di accordo di chiavi post-quantistiche standard](#). Questo algoritmo ibrido utilizza ciascuno degli algoritmi in modo indipendente per generare una chiave. Quindi combina crittograficamente le due chiavi. Con s2n-tls, puoi [configurare un client HTTP](#) con una preferenza per

il protocollo TLS post-quantistico, che inserisce ECDH con ML-KEM in prima posizione nell'elenco delle preferenze. Gli algoritmi di scambio di chiavi classici sono inclusi nell'elenco delle preferenze per garantire la compatibilità, ma sono inferiori nell'ordine delle preferenze.

Utilizzo del TLS post-quantistico ibrido con AWS KMS

Puoi utilizzare il TLS post-quantistico ibrido per le tue chiamate a AWS KMS. Quando si configura l'ambiente di test del client HTTP, tenere presente le seguenti informazioni:

Crittografia in transito

Le suite di crittografia ibrida in s2n-tls vengono utilizzate solo per la crittografia in transito. Proteggono i tuoi dati mentre viaggiano dal client all'endpoint. AWS KMS non utilizza queste suite di crittografia per crittografare i dati con AWS KMS keys.

Invece, quando AWS KMS crittografa i dati con chiavi KMS, utilizza la crittografia simmetrica con chiavi a 256 bit e l'algoritmo Advanced Encryption Standard in Galois Counter Mode (AES-GCM), che è già resistente ai calcoli quantistici. Attacchi teorici futuri di calcolo quantistico su larga scala su testi cifrati creati con chiavi AES-GCM a 256 bit [riducono l'effettiva sicurezza della chiave a 128 bit](#). Questo livello di sicurezza è sufficiente a rendere impossibili gli attacchi di forza bruta ai testi cifrati.

AWS KMS

Sistemi supportati

L'uso delle suite di crittografia ibride in s2n-tls al momento è supportato solo su sistemi Linux. Inoltre, queste suite di crittografia sono supportate solo se supportano Common Runtime, come SDKs ad esempio. AWS SDK for Java 2.x Per vedere un esempio, consulta [Configura il TLS post-quantistico ibrido](#).

AWS KMS Endpoints

AWS KMS [supporta il TLS post-quantistico ibrido su tutti gli endpoint, inclusi gli endpoint convalidati FIPS 140-3](#).

Configura il TLS post-quantistico ibrido

In questa procedura, aggiungi una dipendenza Maven per il client HTTP Common Runtime. AWS Quindi, configura un client HTTP che preferisca il protocollo TLS post-quantistico. Quindi, crea un AWS KMS client che utilizzi il client HTTP.

Per un esempio completo di configurazione e utilizzo del protocollo TLS post-quantistico ibrido con AWS KMS, consultare il repository [aws-kms-pq-tls-example](#).

Note

Il AWS Common Runtime HTTP Client, disponibile in anteprima, è diventato disponibile a livello generale nel febbraio 2023. In tale versione, la classe `TlsCipherPreference` e il parametro del metodo `TlsCipherPreference()` vengono sostituiti dal parametro del metodo `PostQuantumTlsEnabled()`. Se stavi usando questo esempio durante l'anteprima, devi aggiornare il codice.

1. Aggiungi il client AWS Common Runtime alle tue dipendenze Maven. Si consiglia di utilizzare l'ultima versione disponibile.

Ad esempio, questa istruzione aggiunge la versione 2.30.22 del client AWS Common Runtime alle dipendenze Maven.

```
<dependency>
  <groupId>software.amazon.awssdk</groupId>
  <artifactId>aws-crt-client</artifactId>
  <version>2.30.22</version>
</dependency>
```

2. Per abilitare le suite ibride di cifratura post-quantistica, aggiungile al progetto e inizializzalo. AWS SDK for Java 2.x Quindi abilita le suite di crittografia post-quantistica ibrida come mostrato nell'esempio seguente.

Questo codice utilizza il parametro `PostQuantumTlsEnabled()` method per configurare un [client HTTP di runtime AWS comune](#) che preferisce la suite di cifratura post-quantistica ibrida consigliata, ECDH with. ML-KEM Quindi utilizza il client HTTP configurato per creare un'istanza del client asincrono, AWS KMS [KmsAsyncClient](#) Una volta completato questo codice, tutte le richieste [API AWS KMS](#) sull'istanza `KmsAsyncClient` utilizzano un protocollo TLS post-quantistico ibrido.

```
// Configure HTTP client
SdkAsyncHttpClient awsCrtHttpClient = AwsCrtAsyncHttpClient.builder()
    .postQuantumTlsEnabled(true)
    .build();
```

```
// Create the AWS KMS async client
KmsAsyncClient kmsAsync = KmsAsyncClient.builder()
    .httpClient(awsCrtHttpClient)
    .build();
```

3. Metti alla prova le tue AWS KMS chiamate con il TLS post-quantistico ibrido.

Quando richiami le operazioni AWS KMS API sul AWS KMS client configurato, le chiamate vengono trasmesse all' AWS KMS endpoint utilizzando il TLS post-quantistico ibrido. Per testare la tua configurazione, chiama un' AWS KMS API, ad esempio. [ListKeys](#)

```
ListKeysResponse keys = kmsAsync.listKeys().get();
```

Testa la tua configurazione TLS ibrida post-quantistica

Valuta la possibilità di eseguire i seguenti test con suite di crittografia ibrida sulle applicazioni che effettuano chiamate. AWS KMS

- Visualizza la `tlsDetails` sezione nella voce di CloudTrail registro relativa a una chiamata AWS KMS API effettuata dalla tua applicazione. Il `keyExchange` campo deve menzionare un algoritmo ibrido come `X25519MLKEM768`. Per vedere un esempio, consulta [Decrittografia con una chiave di crittografia simmetrica standard su una connessione TLS post-quantistica](#).
- Esegui benchmark utilizzando TLS post-quantistico ibrido. Lo scambio di chiavi ibride aumenta le dimensioni e il tempo di elaborazione di alcuni messaggi nell'handshake TLS, ma l'impatto complessivo sulle prestazioni dovrebbe essere impercettibile nella maggior parte dei casi.
- Provare a connettersi da posizioni diverse. A seconda del percorso di rete seguito dalla richiesta, potresti scoprire che host intermedi, proxy o firewall legacy con ispezione approfondita dei pacchetti (DPI) bloccano la richiesta. Ciò potrebbe derivare dall'utilizzo dei nuovi gruppi di scambio di chiavi nella [ClientHello](#) parte dell'handshake TLS o dai messaggi di scambio di chiavi più grandi. Se hai problemi a risolvere questi problemi, collabora con il tuo team di sicurezza o gli amministratori IT per aggiornare la configurazione pertinente e sbloccare i nuovi gruppi di scambio di chiavi TLS.

Scopri di più sul TLS post-quantistico in AWS KMS

Per ulteriori informazioni sull'utilizzo del TLS ibrido post-quantistico in AWS KMS, consulta le seguenti risorse.

- [Per ulteriori informazioni sulla crittografia post-quantistica all'indirizzo AWS, compresi i collegamenti ai post di blog e ai documenti di ricerca, consulta Crittografia post-quantistica.](#)
- Per informazioni su s2n-tls, consultare [Introduzione di s2n-tls, una nuova implementazione TLS open source](#) e [Utilizzo di s2n-tls](#).
- Per informazioni sul client HTTP AWS Common Runtime, consulta [Configurazione del client HTTP basato su AWS CRT nella Guida per gli sviluppatori](#).AWS SDK for Java 2.x
- Per informazioni sul progetto di crittografia post-quantistica presso il National Institute for Standards and Technology (NIST), consultare [Post-Quantum Cryptography](#) (Crittografia post-quantistica).
- Per informazioni sulla standardizzazione della crittografia post-quantistica del NIST, consulta la sezione [Standardizzazione della crittografia post-quantistica](#).

Connect a AWS KMS tramite un endpoint VPC

Puoi connetterti direttamente AWS KMS tramite un endpoint con interfaccia privata nel tuo cloud privato virtuale (VPC). Quando si utilizza un endpoint VPC di interfaccia, la comunicazione tra il VPC e il VPC AWS KMS viene condotta interamente all'interno della rete. AWS

AWS KMS supporta gli endpoint Amazon Virtual Private Cloud (Amazon VPC) con tecnologia. [AWS PrivateLink](#) Ogni endpoint VPC è rappresentato da una o più [interfacce di rete elastiche](#) (ENIs) con indirizzi IP privati nelle sottoreti VPC.

L'interfaccia VPC endpoint collega il tuo VPC direttamente AWS KMS senza un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione. AWS Direct Connect Le istanze del tuo VPC non necessitano di indirizzi IP pubblici con cui comunicare. AWS KMS

Regioni

AWS KMS supporta gli endpoint VPC e le policy degli endpoint VPC in tutti i casi in cui è supportato. Regioni AWS [AWS KMS](#)

Considerazioni sugli endpoint AWS KMS VPC

Prima di configurare un endpoint VPC di interfaccia per AWS KMS, consulta l'argomento [Proprietà e limitazioni dell'endpoint dell'interfaccia](#) nella Guida.AWS PrivateLink

AWS KMS il supporto per un endpoint VPC include quanto segue.

- Puoi utilizzare l'endpoint VPC per richiamare tutte le [operazioni API AWS KMS](#) dal VPC.

- [È possibile creare un endpoint VPC di interfaccia che si connette a un endpoint AWS KMS regionale o a un endpoint FIPS.AWS KMS](#)
- Puoi utilizzare AWS CloudTrail i log per verificare l'utilizzo delle chiavi KMS tramite l'endpoint VPC. Per informazioni dettagliate, vedi [Registrazione delle AWS KMS richieste che utilizzano un endpoint VPC](#).

Argomenti

- [Crea un endpoint VPC per AWS KMS](#)
- [Connect a un AWS KMS endpoint VPC](#)
- [Usa gli endpoint VPC per controllare l'accesso alle risorse AWS KMS](#)
- [Registrazione delle AWS KMS richieste che utilizzano un endpoint VPC](#)

Crea un endpoint VPC per AWS KMS

Puoi creare un endpoint VPC per AWS KMS utilizzando la console Amazon VPC o l'API Amazon VPC. Segui le procedure per [creare un endpoint di interfaccia](#) utilizzando uno dei seguenti valori.

- Per creare un endpoint VPC per AWS KMS, usa il seguente nome di servizio:

```
com.amazonaws.region.kms
```

Ad esempio, nella regione Stati Uniti occidentali (Oregon) (us-west-2), il nome del servizio sarebbe:

```
com.amazonaws.us-west-2.kms
```

- Per creare un endpoint VPC che si collega a un [endpoint FIPS AWS KMS](#), usa il seguente nome di servizio:

```
com.amazonaws.region.kms-fips
```

Ad esempio, nella regione Stati Uniti occidentali (Oregon) (us-west-2), il nome del servizio sarebbe:

```
com.amazonaws.us-west-2.kms-fips
```

Per semplificare l'utilizzo dell'endpoint VPC, puoi abilitare un [nome DNS privato](#) per l'endpoint VPC. Se selezioni l'opzione Enable DNS Name (Abilita nome DNS), il nome host DNS standard AWS KMS si risolve nell'endpoint VPC. Ad esempio, `https://kms.us-west-2.amazonaws.com` si risolverebbe in un endpoint VPC connesso al nome del servizio `com.amazonaws.us-west-2.kms`.

Questa opzione rende più semplice utilizzare l'endpoint VPC. Per impostazione predefinita, AWS CLI utilizza il nome host AWS KMS DNS standard, quindi non è necessario specificare l'URL dell'endpoint VPC nelle applicazioni e nei comandi. AWS SDKs

Per ulteriori informazioni, consulta la sezione [Accesso a un servizio tramite un endpoint di interfaccia](#) nella Guida di AWS PrivateLink .

Connect a un AWS KMS endpoint VPC

Puoi connetterti AWS KMS tramite l'endpoint VPC utilizzando un AWS SDK, o. AWS CLI AWS Strumenti per PowerShell Per specificare l'endpoint VPC, utilizzare il nome DNS.

Ad esempio, il comando [list-keys](#) utilizza il parametro `endpoint-url` per specificare l'endpoint VPC. Per utilizzare un comando come questo, sostituisci l'ID dell'endpoint VPC con uno presente nel tuo account.

```
$ aws kms list-keys --endpoint-url https://vpce-1234abcd5678c90a-09p7654s-us-east-1a.ec2.us-east-1.vpce.amazonaws.com
```

Autorizzazioni richieste

Affinché una AWS KMS richiesta che utilizza un endpoint VPC abbia esito positivo, il principale richiede le autorizzazioni da due fonti:

- Una [policy chiave](#), una [policy IAM](#) o una [concessione](#) deve concedere l'autorizzazione al principale per chiamare l'operazione sulla risorsa (chiave KMS o alias).
- Una policy di endpoint VPC deve concedere l'autorizzazione al principale per utilizzare l'endpoint per effettuare la richiesta.

Ad esempio, una policy chiave potrebbe concedere al principale un'autorizzazione per chiamare [Decrypt](#) in una determinata chiave KMS. Tuttavia, la policy di endpoint VPC potrebbe non consentire a tale principale di chiamare Decrypt sulla chiave KMS utilizzando l'endpoint.

Oppure una policy [DisableKey](#) sugli endpoint VPC potrebbe consentire a un principale di utilizzare l'endpoint per richiamare determinate chiavi KMS. Tuttavia se il principale non dispone di tali

autorizzazioni da una policy delle chiavi, una policy IAM o una concessione, la richiesta non riesce.

Puoi creare una policy di endpoint VPC quando crei l'endpoint e puoi modificare la policy di endpoint VPC in qualsiasi momento. Utilizza la console di gestione VPC o le operazioni [CreateVpcEndpoint](#) [ModifyVpcEndpoint](#). Puoi anche creare e modificare una policy per gli endpoint VPC [utilizzando](#) un modello. AWS CloudFormation Per informazioni sull'utilizzo della console di gestione VPC, consulta la sezione [Creazione di un endpoint di interfaccia](#) e [Modifica di un endpoint di interfaccia](#) nella Guida di AWS PrivateLink .

Hostname privati

Se hai attivato nomi host privati al momento della creazione dell'endpoint VPC, non è necessario specificare l'URL dell'endpoint VPC nella configurazione dell'applicazione o nei comandi della CLI. Il nome host DNS standard AWS KMS viene risolto nell'endpoint VPC. Seleziona AWS CLI e SDKs utilizza questo nome host per impostazione predefinita, in modo da poter iniziare a utilizzare l'endpoint VPC per connetterti a AWS KMS un endpoint regionale senza modificare nulla negli script e nelle applicazioni.

Per utilizzare nomi host privati, gli attributi `enableDnsHostnames` e `enableDnsSupport` del VPC devono essere impostati su `true`. Per impostare questi attributi, usa l'operazione [ModifyVpcAttribute](#). Per informazioni dettagliate, consulta la sezione [Visualizzazione e aggiornamento degli attributi DNS per il VPC](#) nella Guida per l'utente di Amazon VPC.

Usa gli endpoint VPC per controllare l'accesso alle risorse AWS KMS

Puoi controllare l'accesso alle AWS KMS risorse e alle operazioni quando la richiesta proviene da VPC o utilizza un endpoint VPC. A tale scopo, utilizza una delle seguenti [chiavi di condizione globale](#) in una [policy delle chiavi](#) o una [policy IAM](#).

- Usa la chiave di condizione `aws:sourceVpce` per concedere o limitare l'accesso in base all'endpoint VPC.
- Usa la chiave di condizione `aws:sourceVpc` per concedere o limitare l'accesso in base al VPC che ospita l'endpoint privato.

Note

Fai attenzione durante la creazione delle policy IAM e delle policy delle chiavi basate sull'endpoint VPC. Se una dichiarazione di policy richiede che le richieste provengano da un particolare VPC o endpoint VPC, le richieste provenienti da AWS servizi integrati che utilizzano una AWS KMS risorsa per conto dell'utente potrebbero non riuscire. Per assistenza, consulta [Utilizzo delle condizioni di endpoint VPC nelle policy con autorizzazioni AWS KMS](#).

Inoltre, la chiave di condizione `aws:sourceIP` non è efficace quando la richiesta proviene da un [endpoint Amazon VPC](#). Per limitare le richieste a un endpoint VPC, utilizza il comando `aws:sourceVpce` o le chiavi di condizione `aws:sourceVpc`. Per ulteriori informazioni, consulta la sezione [Gestione delle identità e degli accessi per endpoint VPC e servizi endpoint VPC](#) nella Guida di AWS PrivateLink .

È possibile utilizzare queste chiavi di condizione globali per controllare l'accesso a AWS KMS keys (chiavi KMS), gli alias e a operazioni del genere [CreateKey](#) che non dipendono da alcuna risorsa particolare.

Ad esempio, la seguente policy delle chiavi di esempio consente a un utente di eseguire alcune operazioni di crittografia con una chiave KMS solo quando la richiesta utilizza l'endpoint VPC specificato. Quando un utente effettua una richiesta a AWS KMS, l'ID dell'endpoint VPC nella richiesta viene confrontato con il valore della chiave di `aws:sourceVpce` condizione nella policy. Se non corrisponde, la richiesta viene rifiutata.

Per utilizzare una politica come questa, sostituisci l' Account AWS ID segnaposto e l'endpoint VPC IDs con valori validi per il tuo account.

JSON

```
{
  "Id": "example-key-1",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnableIAMpolicies",
      "Effect": "Allow",
      "Principal": {"AWS": ["111122223333"]},
      "Action": ["kms:*"],
```

```

        "Resource": "*"
    },
    {
        "Sid": "Restrict usage to my VPC endpoint",
        "Effect": "Deny",
        "Principal": "*",
        "Action": [
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*"
        ],
        "Resource": "*",
        "Condition": {
            "StringNotEquals": {
                "aws:sourceVpce": "vpce-1234abcdef5678c90a"
            }
        }
    }
}
]
}

```

Puoi anche utilizzare la chiave di condizione `aws:sourceVpc` per limitare l'accesso alle tue chiavi KMS in base al VPC in cui risiede l'endpoint VPC.

I seguenti comandi di esempio della policy delle chiavi consentono i comandi che gestiscono la chiave KMS solo quando la loro provenienza è `vpc-12345678`. Inoltre, consente i comandi che utilizzano la chiave KMS per le operazioni di crittografia solo quando provengono da `vpc-2b2b2b2b`. Puoi usare una policy come questa se un'applicazione è in esecuzione in un VPC, ma devi utilizzare un secondo VPC separato per le funzioni di gestione.

Per utilizzare una politica come questa, sostituisci l' Account AWS ID segnaposto e l'endpoint VPC IDs con valori validi per il tuo account.

JSON

```

{
    "Id": "example-key-2",
    "Version": "2012-10-17",
    "Statement": [

```

```

{
  "Sid": "AllowAdministrativeActionsFromVPC",
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": [
    "kms:Create*",
    "kms:Enable*",
    "kms:Put*",
    "kms:Update*",
    "kms:Revoke*",
    "kms:Disable*",
    "kms>Delete*",
    "kms:TagResource",
    "kms:UntagResource"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:sourceVpc": "vpc-12345678"
    }
  }
},
{
  "Sid": "AllowKeyUsageFromVPC2b2b2b2b",
  "Effect": "Allow",
  "Principal": {
    "AWS": "111122223333"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:sourceVpc": "vpc-2b2b2b2b"
    }
  }
},
{
  "Sid": "AllowReadActionsEverywhere",

```

```
    "Effect": "Allow",
    "Principal": {
      "AWS": "111122223333"
    },
    "Action": [
      "kms:Describe*",
      "kms:List*",
      "kms:Get*"
    ],
    "Resource": "*"
  }
]
```

Registrazione delle AWS KMS richieste che utilizzano un endpoint VPC

AWS CloudTrail registra tutte le operazioni che utilizzano l'endpoint VPC. Quando una richiesta AWS KMS utilizza un endpoint VPC, l'ID dell'endpoint VPC viene visualizzato nella voce di [AWS CloudTrail registro che registra](#) la richiesta. Puoi utilizzare l'ID endpoint per verificare l'uso del tuo endpoint AWS KMS VPC.

Tuttavia, CloudTrail i registri non includono le operazioni richieste dai responsabili in altri account o le richieste di AWS KMS operazioni su chiavi e alias KMS in altri account. Inoltre, per proteggere il tuo VPC, le richieste che vengono rifiutate da una policy sugli endpoint VPC, ma che altrimenti sarebbero state consentite, non vengono registrate in. [AWS CloudTrail](#)

Ad esempio, questa voce di log di esempio registra una richiesta [GenerateDataKey](#) che utilizza l'endpoint VPC. Il campo `vpcEndpointId` viene visualizzato alla fine della voce di log.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "accountId": "111122223333",
    "userName": "Alice"
  },
  "eventTime": "2018-01-16T05:46:57Z",
  "eventSource": "kms.amazonaws.com",
```

```

"eventName": "GenerateDataKey",
"awsRegion": "eu-west-1",
"sourceIPAddress": "172.01.01.001",
"userAgent": "aws-cli/1.14.23 Python/2.7.12 Linux/4.9.75-25.55.amzn1.x86_64
botocore/1.8.27",
"requestParameters": {
  "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "numberOfBytes": 128
},
"responseElements": null,
"requestID": "a9fff0bf-fa80-11e7-a13c-afcabff2f04c",
"eventID": "77274901-88bc-4e3f-9bb6-acf1c16f6a7c",
"readOnly": true,
"resources": [{
  "ARN": "arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "accountId": "111122223333",
  "type": "AWS::KMS::Key"
}],
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333",
"vpcEndpointId": "vpce-1234abcdf5678c90a"
}

```

Supporto per endpoint dual-stack

AWS KMS fornisce un endpoint pubblico dual-stack che supporta sia i client che i client. IPv4 IPv6
Un endpoint dual-stack consente ai client di comunicare con uno o più indirizzi. AWS KMS IPv4
IPv6 [Per ulteriori informazioni sugli endpoint, consulta AWS KMS Endpoints and Quotas.](#) [AWS Key Management Service](#)

L'endpoint pubblico AWS KMS dual-stack supporta sia i client che i client. <https://kms.your-region.api.aws> IPv4 IPv6 AWS KMS è anche accessibile privatamente IPv6 da IPv4 e verso il tuo cloud privato virtuale (VPC) utilizzando. AWS PrivateLink Per ulteriori informazioni sulla creazione di endpoint VPC con interfaccia privata, vedere. AWS KMS [Connect a AWS KMS tramite un endpoint VPC](#)

Per ulteriori informazioni sull'IPv6 indirizzamento per te VPCs, consulta [How Amazon VPC works](#) nella Amazon Virtual Private Cloud User Guide. Per ulteriori informazioni su come configurare il tuo VPC per la modalità dual-stack, consulta [Indirizzamento IP per le tue sottoreti nella Guida per l'utente VPCs di Amazon Virtual Private Cloud](#).

Funzionalità non disponibili su IPv6

AWS KMS non può comunicare IPv6 con gli archivi AWS CloudHSM chiave o gli archivi di chiavi esterni. Questa limitazione non impedisce di AWS KMS APIs effettuare chiamate IPv6.

AWS KMS concetti

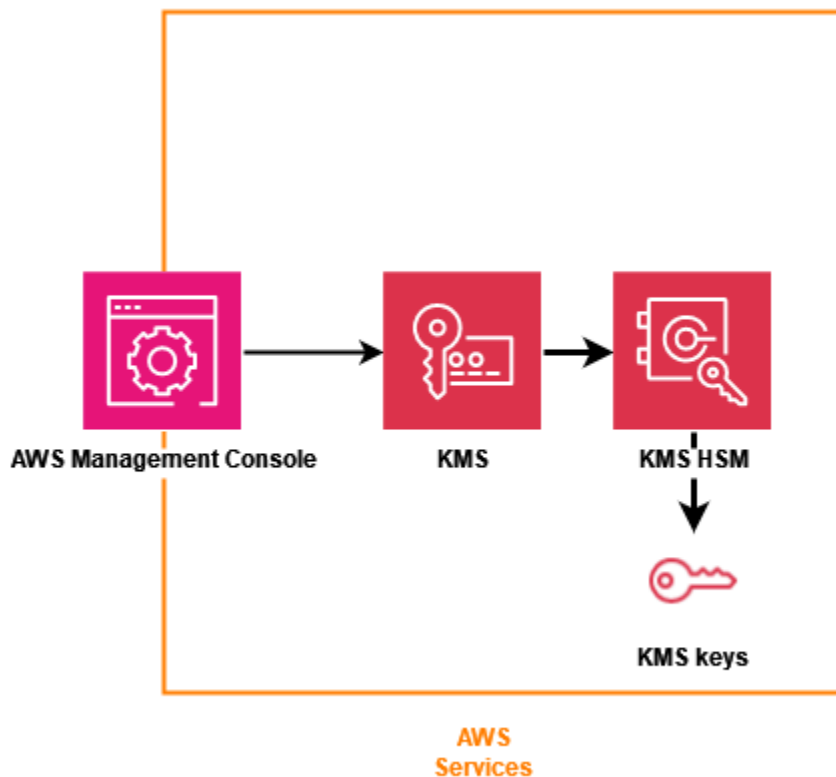
Scopri i termini e i concetti di base utilizzati in AWS Key Management Service (AWS KMS) e come interagiscono per proteggere i tuoi dati.

Introduzione a AWS KMS

AWS Key Management Service (AWS KMS) fornisce un'interfaccia web per generare e gestire chiavi crittografiche e opera come fornitore di servizi crittografici per la protezione dei dati. AWS KMS offre servizi tradizionali di gestione delle chiavi integrati con AWS servizi per fornire una visione coerente delle chiavi dei clienti su tutti i fronti AWS, con gestione e controllo centralizzati.

AWS KMS include un'interfaccia web tramite interfaccia a riga di comando e operazioni RESTful API per richiedere le operazioni crittografiche di una flotta distribuita di moduli di sicurezza hardware convalidati FIPS 140-3 (). Console di gestione AWS HSMs L' AWS KMS HSM è un dispositivo crittografico hardware standalone multichip progettato per fornire funzioni crittografiche dedicate per soddisfare i requisiti di sicurezza e scalabilità di. AWS KMS È possibile stabilire la propria gerarchia crittografica basata su HSM nelle chiavi gestite come AWS KMS keys. Queste chiavi sono disponibili solo su HSMs e solo in memoria per il tempo necessario all'elaborazione della richiesta crittografica. È possibile creare più chiavi KMS, ognuna rappresentata dal proprio ID chiave. Solo con i ruoli e gli account AWS IAM amministrati da ciascun cliente è possibile creare, eliminare o utilizzare chiavi KMS gestite dal cliente per crittografare, decrittografare, firmare o verificare i dati. Puoi definire i controlli di accesso su chi può gestire and/or l'uso delle chiavi KMS creando una policy allegata alla chiave. Tali policy consentono di definire usi specifici dell'applicazione per le chiavi per ogni operazione API.

Inoltre, la maggior parte dei AWS servizi supporta la crittografia dei dati inattivi utilizzando le chiavi KMS. Questa funzionalità consente ai clienti di controllare come e quando AWS i servizi possono accedere ai dati crittografati controllando come e quando è possibile accedere alle chiavi KMS.



AWS KMS è un servizio a più livelli composto da AWS KMS host rivolti al Web e da un livello di HSMs. Il raggruppamento di questi host a più livelli costituisce lo stack. AWS KMS. Tutte le richieste AWS KMS devono essere effettuate tramite il protocollo Transport Layer Security (TLS) e terminate su un host. AWS KMS [AWS KMS gli host consentono TLS solo con una suite di crittografia che fornisce una perfetta segretezza di inoltro](#). AWS KMS autentica e autorizza le richieste utilizzando gli stessi meccanismi di credenziali e policy di AWS Identity and Access Management (IAM) disponibili per tutte le altre operazioni API. AWS

AWS KMS obiettivi di progettazione

AWS KMS è progettato per soddisfare i seguenti requisiti.

Durabilità

La durabilità delle chiavi crittografiche è progettata per eguagliare quella dei servizi di massima durabilità in AWS. Una singola chiave di crittografia può crittografare grandi volumi di dati accumulati per un lungo periodo di tempo.

Affidabile

L'utilizzo delle chiavi è protetto alle policy di controllo accessi definite e gestite dall'utente. Non esiste alcun meccanismo per esportare le chiavi KMS in chiaro. La riservatezza delle chiavi di crittografia è fondamentale. Sono necessari più dipendenti Amazon con accesso specifico per ruolo ai controlli di accesso basati sul quorum per eseguire azioni amministrative su HSMs

Bassa latenza e velocità effettiva elevata

AWS KMS fornisce operazioni crittografiche a livelli di latenza e velocità effettiva adatti all'uso da parte di altri servizi in AWS

Regioni indipendenti

AWS fornisce regioni indipendenti per i clienti che devono limitare l'accesso ai dati in diverse regioni. L'utilizzo delle chiavi può essere isolato all'interno di una Regione AWS.

Fonte sicura di numeri casuali

Poiché una crittografia avanzata dipende da una generazione di numeri casuali davvero imprevedibile, AWS KMS fornisce una fonte di numeri casuali convalidata e di alta qualità.

Audit

AWS KMS registra l'uso e la gestione delle chiavi crittografiche nei log. AWS CloudTrail È possibile utilizzare AWS CloudTrail i log per controllare l'uso delle chiavi crittografiche, incluso l'uso delle chiavi da parte AWS dei servizi che operano per conto dell'utente.

Per raggiungere questi obiettivi, il AWS KMS sistema include una serie di AWS KMS operatori e operatori di service host (collettivamente, «operatori») che amministrano i «domini». Un dominio è un insieme di AWS KMS server e operatori definito a livello regionale. HSMs Ogni AWS KMS operatore dispone di un token hardware che contiene una coppia di chiavi privata e una pubblica che viene utilizzata per autenticare le sue azioni. HSMs Dispongono di un'ulteriore coppia di chiavi private e pubbliche per stabilire chiavi di crittografia che proteggono la sincronizzazione dello stato HSM.

AWS KMS keys

Le chiavi KMS create e gestite per essere utilizzate nelle vostre applicazioni crittografiche sono di un tipo noto come chiavi gestite dal cliente. Le chiavi gestite dal cliente possono essere utilizzate anche in combinazione con AWS servizi che utilizzano le chiavi KMS per crittografare i dati archiviati dal servizio per conto dell'utente. Le chiavi gestite dal cliente sono consigliate ai clienti che desiderano il

pieno controllo sul ciclo di vita e sull'utilizzo delle proprie chiavi. È previsto un costo mensile per avere una chiave gestita dal cliente nel proprio account. Inoltre, le richieste relative all'utilizzo e alla and/or gestione della chiave comportano un costo di utilizzo. Per maggiori dettagli, consulta la sezione [AWS Key Management Service Prezzi](#).

Ci sono casi in cui un cliente potrebbe desiderare che un AWS servizio crittografi i propri dati, ma non vuole il sovraccarico di gestione delle chiavi e non vuole pagare per una chiave. Una Chiave gestita da AWS è una chiave KMS presente nel tuo account, ma può essere utilizzata solo in determinate circostanze. In particolare, può essere utilizzata solo nel contesto del AWS servizio in cui operi e può essere utilizzata solo dai responsabili all'interno dell'account in cui esiste la chiave. Non puoi gestire nulla sul ciclo di vita o sulle autorizzazioni di queste chiavi. <service code>Come potete notare, quando utilizzate le funzionalità di crittografia nei AWS servizi Chiavi gestite da AWS, questi utilizzano un alias nel formato «aws». Ad esempio, una aws/ebs chiave può essere utilizzata solo per crittografare i volumi EBS e solo per i volumi utilizzati dai responsabili IAM nello stesso account della chiave. Pensa a una Chiave gestita da AWS che sia destinata ad essere utilizzata solo dagli utenti del tuo account per le risorse del tuo account. Non puoi condividere risorse crittografate con e Chiave gestita da AWS con altri account. Sebbene nel tuo account possa esistere gratuitamente, ogni utilizzo di questo tipo di chiave ti viene addebitato dal AWS servizio assegnato alla chiave. Chiave gestita da AWS

Chiavi gestite da AWS sono un tipo di chiave legacy che non viene più creato per nuovi AWS servizi a partire dal 2021. Invece, i AWS servizi nuovi (e legacy) utilizzano il cosiddetto «an» Chiave di proprietà di AWS per crittografare i dati dei clienti per impostazione predefinita. Una Chiave di proprietà di AWS è una chiave KMS contenuta in un account gestito dal AWS servizio, in modo che gli operatori del servizio abbiano la possibilità di gestirne il ciclo di vita e le autorizzazioni di utilizzo. Grazie all'utilizzo Chiavi di proprietà di AWS, AWS i servizi possono crittografare i dati in modo trasparente e consentire una facile condivisione dei dati tra account o tra regioni diverse senza che l'utente debba preoccuparsi delle autorizzazioni chiave. Utilizzate Chiavi di proprietà di AWS per encryption-by-default carichi di lavoro che forniscono una protezione dei dati più semplice e automatizzata. Poiché queste chiavi sono possedute e gestite da AWS, non ti viene addebitato alcun costo per la loro esistenza o il loro utilizzo, non puoi modificarne le politiche, non puoi controllare le attività su queste chiavi e non puoi eliminarle. Utilizzate le chiavi gestite dal cliente quando il controllo è importante, ma usatele Chiavi di proprietà di AWS quando la praticità è più importante.

Chiavi gestite dal
cliente

Chiavi gestite da AWS

Chiavi di proprietà di
AWS

Policy della chiave	Controllato esclusivamente dal cliente	Controllato dal servizio; visualizzabile dal cliente	Controllato esclusivamente e visualizzabile solo dal AWS servizio che crittografa i dati
Registrazione di log	CloudTrail percorso dei clienti o archivio dati sugli eventi	CloudTrail percorso dei clienti o archivio dati sugli eventi	Non visualizzabile dal cliente
Gestione del ciclo di vita	Il cliente gestisce la rotazione, l'eliminazione e l'ubicazione regionale	AWS KMS gestisce la rotazione (annuale), l'eliminazione e la sede regionale	Servizio AWS gestisce la rotazione, l'eliminazione e la posizione regionale
Prezzi	Tariffa mensile per l'esistenza delle chiavi (ripartita proporzionalmente su base oraria). Addebitato anche per l'utilizzo delle chiavi	Nessun canone mensile, ma al chiamante viene addebitato l'utilizzo dell'API su queste chiavi	Nessun addebito per il cliente

Le chiavi KMS create dall'utente sono [chiavi gestite dal cliente](#). I Servizi AWS che utilizzano le chiavi KMS per crittografare le risorse di servizio spesso creano le chiavi per conto dell'utente. Le chiavi KMS Servizi AWS create nel tuo AWS account sono [Chiavi gestite da AWS](#). Le chiavi KMS che vengono Servizi AWS create in un account di servizio sono. [Chiavi di proprietà di AWS](#)

Tipo di chiave KMS	Può visualizzare i metadati della chiave KMS	Può gestire la chiave KMS	Usato solo per il mio Account AWS	Rotazione automatica	Prezzi
Chiave gestita dal cliente	Sì	Sì	Sì	Opzionale.	Canone mensile (proporzionale a ora)

Tipo di chiave KMS	Può visualizzare i metadati della chiave KMS	Può gestire la chiave KMS	Usato solo per il mio Account AWS	Rotazione automatica	Prezzi
					Tariffa per uso
Chiave gestita da AWS	Sì	No	Sì	Obbligatorio. Ogni anno (circa 365 giorni).	Nessuna tariffa mensile Tariffa per utilizzo (alcuni Servizi AWS pagano questa tariffa per te)
Chiave di proprietà di AWS	No	No	No	Servizio AWS Gestisce la strategia di rotazione	Nessuna tariffa

[AWS i servizi che si AWS KMS integrano con](#) differiscono nel supporto per le chiavi KMS. Per impostazione predefinita, alcuni AWS servizi crittografano i dati con un Chiave di proprietà di AWS o un. Chiave gestita da AWS Alcuni AWS servizi supportano le chiavi gestite dai clienti. Altri AWS servizi supportano tutti i tipi di chiavi KMS per consentire la facilità di utilizzo Chiave di proprietà di AWS, la visibilità o il controllo di una chiave gestita dal cliente. Chiave gestita da AWS Per informazioni dettagliate sulle opzioni di crittografia offerte da un AWS servizio, consulta l'argomento Encryption at Rest nella guida per l'utente o la guida per sviluppatori del servizio.

Chiavi gestite dal cliente

Le chiavi KMS create dall'utente sono chiavi gestite dal cliente. Le chiavi gestite dal cliente sono chiavi KMS Account AWS che create, possedete e gestite dall'utente. L'utente ha il controllo completo su queste chiavi KMS, tra cui la definizione e il mantenimento delle [policy chiave, delle policy IAM](#)

[e delle concessioni](#), la loro [attivazione e disattivazione](#), la [rotazione del materiale crittografico](#), [l'aggiunta di tag](#), la [creazione di alias](#) relativi alle chiavi KMS e la [programmazione di chiavi KMS per l'eliminazione](#).

Le chiavi gestite dal cliente vengono visualizzate nella pagina chiavi gestite dal cliente della Console di gestione AWS per AWS KMS. Per identificare in modo definitivo una chiave gestita dal cliente, utilizza l'[DescribeKey](#) operazione. Per le chiavi gestite dal cliente, il valore del campo KeyManager della risposta di DescribeKey è CUSTOMER.

Si possono utilizzare chiavi gestite dal cliente in operazioni di crittografia e verificarne l'uso nei registri AWS CloudTrail. Inoltre, molti [servizi AWS che si integrano con AWS KMS](#) consentono di specificare una chiave gestita dal cliente per proteggere i dati archiviati e gestiti per l'utente.

Le chiavi gestite dal cliente sono soggette a una tariffa mensile e a una tariffa qualora l'utilizzo superi i termini del piano gratuito. Vengono conteggiati nelle AWS KMS [quote](#) del tuo account. Per i dettagli, vedere le sezioni [Prezzi AWS Key Management Service](#) e [Quote](#).

Chiavi gestite da AWS

Chiavi gestite da AWS sono chiavi KMS del tuo account che vengono create, gestite e utilizzate per tuo conto da un [AWS servizio integrato](#) con AWS KMS.

Alcuni AWS servizi consentono di scegliere una chiave Chiave gestita da AWS o una chiave gestita dal cliente per proteggere le risorse di quel servizio. In generale, a meno che non sia necessario controllare la chiave di crittografia che protegge le risorse, una Chiave gestita da AWS è una buona scelta. Non è necessario creare o mantenere la chiave o la relativa policy delle chiavi e non è mai previsto un canone mensile per una Chiave gestita da AWS.

Hai il permesso di [visualizzarle Chiavi gestite da AWS](#) nel tuo account, [visualizzare le relative politiche chiave](#) e [controllarne l'utilizzo](#) nei AWS CloudTrail log. Tuttavia, non è possibile modificare alcuna proprietà Chiavi gestite da AWS, ruotarle, modificarne le politiche chiave o pianificarne l'eliminazione. Inoltre, non è possibile utilizzarle direttamente Chiavi gestite da AWS nelle operazioni crittografiche; il servizio che le crea le utilizza per conto dell'utente.

[Le politiche di controllo delle risorse](#) dell'organizzazione non si applicano a Chiavi gestite da AWS.

Chiavi gestite da AWS appaiono nella Chiavi gestite da AWS pagina del Console di gestione AWS modulo AWS KMS. È inoltre possibile identificarli Chiavi gestite da AWS tramite i relativi alias, che hanno il formato `aws/service-name`, ad esempio. `aws/redshift` Per identificare definitivamente

un Chiavi gestite da AWS, utilizzate l'[DescribeKey](#) operazione. Per le Chiavi gestite da AWS, il valore del campo KeyManager della risposta DescribeKey è AWS.

Tutti Chiavi gestite da AWS vengono ruotati automaticamente ogni anno. Non è possibile modificare questo programma di rotazione.

Note

A maggio 2022, AWS KMS ha modificato il programma di rotazione Chiavi gestite da AWS da ogni tre anni (circa 1.095 giorni) a ogni anno (circa 365 giorni).

Non è previsto alcun canone mensile per Chiavi gestite da AWS. Il loro utilizzo può essere soggetto a tariffe superiori a quelle del piano gratuito, ma alcuni AWS servizi coprono questi costi per te. Per informazioni dettagliate, consulta l'argomento Crittografia dei dati inattivi nella guida per l'utente o nella guida per gli sviluppatori del servizio. Per informazioni dettagliate, consulta [Prezzi di AWS Key Management Service](#).

Chiavi gestite da AWS non conteggiate ai fini delle quote di risorse sul numero di chiavi KMS in ciascuna regione del vostro account. Tuttavia, quando vengono utilizzate per conto di un principale nel tuo account, le chiavi KMS vengono conteggiate ai fini delle quote di richiesta. Per informazioni dettagliate, vedi [Quote](#).

Chiavi di proprietà di AWS

Chiavi di proprietà di AWS sono una raccolta di chiavi KMS possedute e gestite da un AWS servizio e utilizzabili in più lingue. Account AWS Sebbene non Chiavi di proprietà di AWS siano presenti nel tuo account Account AWS, un AWS servizio può Chiave di proprietà di AWS utilizzarlo per proteggere le risorse del tuo account.

Alcuni AWS servizi consentono di scegliere una chiave Chiave di proprietà di AWS o una chiave gestita dal cliente. In generale, a meno che non sia necessario verificare o controllare la chiave di crittografia che protegge le risorse, un Chiave di proprietà di AWS è una buona scelta. Chiavi di proprietà di AWS sono completamente gratuiti (senza canoni mensili o costi di utilizzo), non influiscono sulle [AWS KMS quote](#) del tuo account e sono facili da usare. Non è necessario creare o mantenere la chiave o la relativa policy delle chiavi.

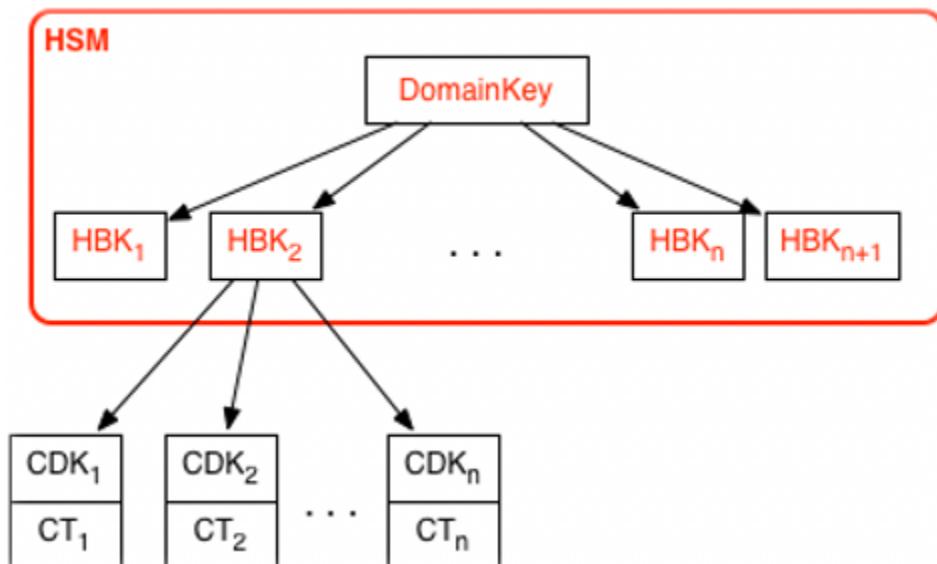
La rotazione di Chiavi di proprietà di AWS varia a seconda dei servizi. Per informazioni sulla rotazione di un determinato servizio Chiave di proprietà di AWS, consulta l'argomento Encryption at Rest nella guida per l'utente o nella guida per sviluppatori del servizio.

AWS KMS key gerarchia

La tua gerarchia delle chiavi inizia con una chiave logica di primo livello, un. AWS KMS key Una chiave KMS rappresenta un container per il materiale della chiave di primo livello ed è definita in modo univoco all'interno dello spazio dei nomi del servizio AWS con un Amazon Resource Name (ARN). L'ARN include un identificatore di chiave generato in modo univoco, un ID chiave. Una chiave KMS viene creata sulla base di una richiesta avviata dall'utente tramite. AWS KMS Alla ricezione, AWS KMS richiede la creazione di una chiave di supporto HSM iniziale (HBK) da inserire nel contenitore delle chiavi KMS. L'HBK viene generata su una HSM nel dominio ed è progettata per non essere mai esportata da HSM in testo normale. Invece, l'HBK viene esportata crittografata in chiavi di dominio gestite da HSM. Questi token esportati HBKs sono denominati token chiave esportati (). EKTs

L'EKT viene esportato in uno spazio di archiviazione altamente durevole e a bassa latenza. Si supponga, ad esempio, di ricevere un ARN per la chiave logica KMS. Questo rappresenta la parte superiore di una gerarchia di chiavi, o contesto crittografico. Puoi creare più chiavi KMS all'interno del tuo account e impostare politiche sulle tue chiavi KMS come qualsiasi altra risorsa denominata. AWS

All'interno della gerarchia di una chiave KMS specifica, l'HBK può essere considerata come una versione della chiave KMS. Quando desideri ruotare la chiave KMS AWS KMS, viene creato un nuovo HBK e associato alla chiave KMS come HBK attivo per la chiave KMS. Le versioni precedenti HBKs vengono conservate e possono essere utilizzate per decrittografare e verificare i dati precedentemente protetti. Ma solo la chiave di crittografia attiva può essere utilizzata per proteggere nuove informazioni.



Puoi richiedere di utilizzare le tue chiavi KMS AWS KMS per proteggere direttamente le informazioni o richiedere chiavi aggiuntive generate da HSM protette dalla tua chiave KMS. Queste chiavi sono chiamate chiavi dei dati del cliente o. CDKs CDKs possono essere restituite crittografate come testo cifrato (CT), in testo non crittografato o entrambi. Tutti gli oggetti crittografati con una chiave KMS (dati forniti dal cliente o chiavi generate da HSM) possono essere decrittografati solo su un HSM tramite una chiamata. AWS KMS

Il testo cifrato restituito, o il payload decrittografato, non viene mai archiviato all'interno. AWS KMS Le informazioni vengono restituite tramite la connessione TLS a AWS KMS. Ciò vale anche per le chiamate effettuate dai AWS servizi per conto dell'utente.

La gerarchia delle chiavi e le proprietà della chiave specifiche vengono visualizzate nella tabella seguente.

Chiave	Description	Ciclo di vita
Chiave di dominio	Una chiave AES-GCM a 256 bit solo in memoria di un HSM utilizzato per avvolgere le versioni delle chiavi KMS, le chiavi di supporto HSM.	Rotazione giornaliera ¹
Materiale della chiave HSM	Una chiave simmetrica a 256 bit o RSA o chiave privata della curva ellittica, utilizzata per proteggere i dati e le chiavi dei clienti e archiviata crittografata con le chiavi di dominio. Una o più chiavi di supporto HSM comprendono la chiave KMS, rappresentata da keyId.	Rotazione annuale ² (config. facoltativa)
Chiave di crittografia derivata	Una chiave AES-GCM a 256 bit solo in memoria di un HSM utilizzato per crittografare i dati e le chiavi dei clienti. Derivato da una HBK per ogni crittografia.	Usato una volta per crittografare e rigenerato sulla decrittografia

Chiave	Description	Ciclo di vita
Chiave dei dati del cliente	Chiave simmetrica o asimmetrica definita dall'utente esportata da HSM in testo normale e cifrato. Crittografata con una chiave di supporto HSM e restituita agli utenti autorizzati sul canale TLS.	Rotazione e utilizzo controllati dall'applicazione

Di tanto in tanto AWS KMS potresti ridurre la rotazione delle chiavi di dominio a una rotazione settimanale al massimo per tenere conto delle attività di amministrazione e configurazione del dominio.

² Le impostazioni predefinite Chiavi gestite da AWS create e gestite da AWS KMS per conto dell'utente vengono ruotate automaticamente ogni anno.

Identificatori chiave () KeyId

Gli identificatori delle chiavi fungono da nomi per le tue chiavi KMS. Consentono di riconoscere le chiavi KMS nella console. Puoi utilizzarli per indicare quali chiavi KMS vuoi utilizzare nelle operazioni API AWS KMS, nelle policy chiave, nelle policy IAM e nelle concessioni. Gli identificatori delle chiavi non sono in alcun modo correlati al materiale chiave associato alla chiave KMS.

AWS KMS definisce diversi identificatori chiave. Quando crei una chiave KMS, AWS KMS genera un ARN e un ID chiave, che sono proprietà della chiave KMS. Quando crei un [alias](#), AWS KMS genera un alias ARN in base al nome dell'alias che definisci. È possibile visualizzare gli identificatori di chiave e alias nella e nell'API. Console di gestione AWS AWS KMS

Nella AWS KMS console, puoi visualizzare e filtrare le chiavi KMS in base all'ARN della chiave, all'ID della chiave o al nome alias e ordinare per ID chiave e nome alias. Per informazioni su come individuare gli identificatori della chiave nella console, consulta [the section called “Trova l'ID della chiave e l'ARN della chiave”](#).

Nell' AWS KMS API, i parametri utilizzati per identificare una chiave KMS hanno un nome KeyId o una variante, ad esempio `TargetKeyId` `DestinationKeyId`. Tuttavia, i valori di tali parametri non si limitano alla chiave. IDs Alcuni possono prendere qualsiasi identificatore di chiave valido. Per informazioni sui valori di ciascun parametro, consulta la descrizione del parametro nell' AWS Key Management Service API Reference.

 Note

Quando usi l' AWS KMS API, fai attenzione all'identificatore di chiave che usi. Diversi APIs richiedono identificatori chiave diversi. In generale, utilizza l'identificatore di chiave più completo e pratico per il processo.

AWS KMS supporta i seguenti identificatori chiave.

ARN della chiave

L'ARN di chiave è il nome della risorsa Amazon (ARN) di una chiave KMS. Si tratta di un identificatore univoco e completo per la chiave KMS. L'ARN di una chiave include la Account AWS regione e l'ID della chiave. Per informazioni su come individuare l'ARN di una chiave KMS, consulta [the section called “Trova l'ID della chiave e l'ARN della chiave”](#).

Il formato di un ARN della chiave è il seguente:

```
arn:<partition>:kms:<region>:<account-id>:key/<key-id>
```

Di seguito è riportato un esempio di ARN della chiave per una chiave KMS per una singola Regione.

```
arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

L'*key-id* elemento della chiave ARNs delle chiavi [multiregionali](#) inizia con il mrk- prefisso. Di seguito è riportato un esempio di ARN della chiave per una chiave KMS in più Regioni.

```
arn:aws:kms:us-west-2:111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab
```

ID chiave

L'ID chiave identifica in modo univoco una chiave KMS all'interno di un account e di una Regione. Per informazioni su come individuare l'ID chiave di una chiave KMS, consulta [the section called “Trova l'ID della chiave e l'ARN della chiave”](#).

Di seguito è riportato un esempio di ID chiave per una chiave KMS per una singola Regione.

```
1234abcd-12ab-34cd-56ef-1234567890ab
```

La chiave IDs delle chiavi [multiregionali](#) inizia con il prefisso. `mrk-` Di seguito è riportato un esempio di ARN della chiave per una chiave KMS in più Regioni.

```
mrk-1234abcd12ab34cd56ef1234567890ab
```

ARN dell'alias

L'alias ARN è l'Amazon Resource Name (ARN) di un alias. AWS KMS Si tratta di un identificatore univoco e completo per l'alias e per la chiave KMS che rappresenta. Un alias ARN include, Region e Account AWS il nome dell'alias.

In qualsiasi momento, un ARN di alias identifica una particolare chiave KMS. Tuttavia, poiché puoi modificare la chiave KMS associata all'alias, l'ARN di alias può identificare chiavi KMS diverse in momenti diversi. Per informazioni su come individuare l'ARN di alias di una chiave KMS, consulta [Trova il nome dell'alias e l'alias ARN per una chiave KMS](#).

Il formato di un ARN dell'alias è il seguente:

```
arn:<partition>:kms:<region>:<account-id>:alias/<alias-name>
```

Di seguito è riportato l'ARN dell'alias per un `ExampleAlias` fittizio.

```
arn:aws:kms:us-west-2:111122223333:alias/ExampleAlias
```

Nome alias

Il nome alias è una stringa di massimo 256 caratteri. Il nome alias identifica in modo univoco una chiave KMS associata all'interno di un account e di una regione. Nell' AWS KMS API, i nomi degli alias iniziano sempre con. `alias/` Per informazioni sulla ricerca del nome dell'alias di una chiave KMS, consulta [Trova il nome dell'alias e l'alias ARN per una chiave KMS](#).

Il formato di un nome alias è il seguente:

```
alias/<alias-name>
```

Ad esempio:

```
alias/ExampleAlias
```

Il prefisso `aws/` di un nome `alias` è riservato alle [Chiavi gestite da AWS](#). Non è possibile creare un `alias` con questo prefisso. Ad esempio, il nome `alias` di Amazon Chiave gestita da AWS Simple Storage Service (Amazon S3) è il seguente.

```
alias/aws/s3
```

Tasti asimmetrici in AWS KMS

Una chiave KMS asimmetrica rappresenta una coppia di chiavi, una pubblica e una privata, correlate matematicamente. Puoi distribuire la chiave pubblica anche a qualcuno che non consideri attendibile, ma la chiave privata deve essere tenuta segreta.

In una chiave KMS asimmetrica, la chiave privata viene creata e non esce mai non crittografata. AWS KMS Per utilizzare la chiave privata, è necessario chiamare. AWS KMS È possibile utilizzare la chiave pubblica interna AWS KMS chiamando le operazioni AWS KMS API. In alternativa, puoi [scaricare la chiave pubblica](#) e usarla all'esterno di AWS KMS.

Se il tuo caso d'uso richiede la crittografia al AWS di fuori degli utenti che non possono effettuare chiamate AWS KMS, le chiavi KMS asimmetriche sono una buona scelta. Tuttavia, se stai creando una chiave KMS per crittografare i dati archiviati o gestiti in un AWS servizio, utilizza una chiave KMS con crittografia simmetrica. [AWS i servizi integrati con AWS KMS](#) utilizzano solo chiavi KMS di crittografia simmetrica per crittografare i dati. Questi servizi non supportano la crittografia con chiavi KMS asimmetriche.

Quando si firmano messaggi di dimensioni superiori a 4 KB con AWS KMS, è necessario eseguire l'hash del messaggio all'esterno prima di firmare. AWS KMS offre tre `MessageType` opzioni per la gestione dell'input dei messaggi: `RAW` per i messaggi in chiaro (dove AWS KMS esegue l'hashing), `DIGEST` per i messaggi pre-hash (dove AWS KMS salta la fase di hashing) e in `EXTERNAL_MU` particolare per le specifiche chiave ML-DSA KMS in cui l'input è un valore μ rappresentativo di 64 byte. Per messaggi di grandi dimensioni che superano il limite di 4 KB, esegui l'hash del messaggio esternamente e utilizzalo [MessageType:DIGEST](#) (o [MessageType:EXTERNAL_MU](#) per le chiavi KMS ML-DSA) quando richiami le operazioni di AWS KMS [firma](#) e AWS KMS [verifica](#).

AWS KMS supporta diversi tipi di chiavi KMS asimmetriche.

chiavi RSA KMS

Una chiave KMS con una coppia di chiavi RSA per la crittografia e la decrittografia o la firma e la verifica (ma non entrambe). AWS KMS supporta diverse lunghezze di chiave per diversi requisiti di sicurezza.

[Per dettagli tecnici sugli algoritmi di crittografia e firma che AWS KMS supportano le chiavi RSA KMS, consulta le specifiche delle chiavi RSA.](#)

Chiavi KMS Elliptic Curve (ECC)

Una chiave KMS con una coppia di chiavi a curva ellittica per la firma e la verifica o la derivazione di segreti condivisi (ma non entrambi). AWS KMS supporta diverse curve di uso comune.

[Per dettagli tecnici sugli algoritmi di firma che AWS KMS supportano le chiavi ECC KMS, consulta le specifiche chiave della curva ellittica.](#)

Chiavi KMS ML-DSA

Una chiave KMS con una coppia di chiavi ML-DSA per la firma e la verifica. ML-DSA è uno standard di crittografia post-quantistica sviluppato dal National Institute of Standards and Technology (NIST) degli Stati Uniti per proteggere dalle minacce alla sicurezza rappresentate dall'informatica quantistica. ML-DSA è l'algoritmo di firma digitale consigliato per le organizzazioni che passano dagli algoritmi di firma digitale RSA o Elliptic Curve alla crittografia sicura post-quantistica.

AWS KMS supporta diverse lunghezze di chiave per diversi requisiti di sicurezza. [Per dettagli tecnici sugli algoritmi di firma che AWS KMS supportano le chiavi KMS ML-DSA, consulta le specifiche delle chiavi ML-DSA.](#)

SM2 Chiavi KMS (solo regioni della Cina)

Una chiave KMS con una coppia di SM2 chiavi per la crittografia e la decrittografia, la firma e la verifica o la derivazione di segreti condivisi (devi scegliere un tipo). [Key usage](#)

Per dettagli tecnici sugli algoritmi di crittografia e firma che AWS KMS supportano le chiavi SM2 KMS (solo per le regioni della Cina), consulta le specifiche delle [SM2 chiavi](#).

Per maggiori informazioni sulla configurazione della chiave asimmetrica, consulta la sezione [Scelta del tipo di chiave KMS da creare](#).

Regioni

Le chiavi KMS asimmetriche e le coppie di chiavi dati asimmetriche sono supportate in tutti i supporti. Regioni AWS AWS KMS

Ulteriori informazioni

- Per creare chiavi KMS asimmetriche, consultare [Creazione di una chiave KMS asimmetrica](#).
- Per creare chiavi KMS asimmetriche multi-regione, consulta la sezione [Crea chiavi primarie multiregionali](#).
- Per informazioni su come firmare messaggi e verificare le firme con chiavi KMS asimmetriche, consulta [Firma digitale con le nuove chiavi asimmetriche di AWS KMS](#) nel Blog di AWS sulla sicurezza.
- Per ulteriori informazioni sulle considerazioni speciali sull'eliminazione delle chiavi KMS asimmetriche, consulta [Deleting asymmetric KMS keys](#).
- Per identificare e visualizzare le chiavi KMS asimmetriche, vedi [Identifica le chiavi KMS asimmetriche](#).

Chiavi HMAC in ingresso AWS KMS

Le chiavi KMS Hash-Based Message Authentication Code (HMAC) sono chiavi simmetriche che vengono utilizzate per generare e verificare all'interno. HMACs AWS KMS Il materiale della chiave univoco associato a ciascuna chiave KMS HMAC fornisce la chiave privata richiesta dagli algoritmi HMAC. È possibile utilizzare una chiave KMS HMAC con le operazioni [GenerateMac](#) e [VerifyMac](#) per verificare l'integrità e l'autenticità dei dati all'interno di AWS KMS.

Gli algoritmi HMAC combinano una funzione hash crittografica e una chiave segreta condivisa. Prendono un messaggio e una chiave segreta, come il materiale della chiave in una chiave KMS HMAC, e restituiscono un codice univoco di dimensioni fisse o tag. Se un solo carattere del messaggio cambia o se la chiave segreta non è identica, il tag risultante è completamente diverso. Richiedendo una chiave segreta, HMAC fornisce anche l'autenticità; è impossibile generare un tag HMAC identico senza la chiave segreta. HMACs a volte vengono chiamate firme simmetriche, perché funzionano come le firme digitali, ma utilizzano un'unica chiave sia per la firma che per la verifica.

[Le chiavi HMAC KMS e gli algoritmi HMAC utilizzati sono conformi agli standard di settore definiti nella AWS KMS RFC 2104](#). L'operazione genera tag HMAC standard. AWS KMS [GenerateMac](#) Le chiavi KMS HMAC vengono generate in moduli di sicurezza AWS KMS hardware certificati secondo il [programma di convalida dei moduli crittografici FIPS 140-3 \(tranne nelle regioni di Cina \(Pechino\)\)](#)

e Cina (Ningxia)) e non vengono mai lasciate non crittografate. AWS KMS Per utilizzare una chiave KMS HMAC, è necessario richiamare AWS KMS.

Le chiavi KMS HMAC sono utilizzate per determinare l'autenticità di un messaggio, come un token Web JSON (JWT), le informazioni della carta di credito tokenizzate o una password inviata. Possono anche essere utilizzate come funzioni sicure di derivazione delle chiavi (KDFs), specialmente nelle applicazioni che richiedono chiavi deterministiche.

Le chiavi HMAC KMS offrono un vantaggio HMACs rispetto al software applicativo perché il materiale chiave viene generato e utilizzato interamente all'interno AWS KMS, in base ai controlli di accesso impostati sulla chiave.

Tip

Le best practice consigliano di limitare la durata dell'efficacia di qualsiasi meccanismo di firma, incluso un HMAC. Ciò scoraggia un attacco in cui l'attore utilizza un messaggio firmato per stabilire la validità ripetutamente o molto tempo dopo la sostituzione del messaggio. I tag HMAC non includono un timestamp, ma puoi includere un timestamp nel token o nel messaggio per rilevare più facilmente quando è il momento di aggiornare l'HMAC.

Operazioni crittografiche supportate

Le chiavi KMS HMAC supportano solo le operazioni crittografiche [GenerateMac](#) e [VerifyMac](#). Non è possibile utilizzare le chiavi KMS HMAC per crittografare i dati o firmare messaggi o utilizzare qualsiasi altro tipo di chiave KMS nelle operazioni HMAC. Quando utilizzi l'operazione `GenerateMac`, fornisci un messaggio fino a 4.096 byte, una chiave KMS HMAC e l'algoritmo MAC compatibile con la specifica della chiave HMAC, e `GenerateMac` calcola il tag HMAC. Per verificare un tag HMAC, è necessario fornire il tag HMAC e lo stesso messaggio, la chiave KMS HMAC e l'algoritmo MAC utilizzato da `GenerateMac` per calcolare il tag HMAC originale. L'operazione `VerifyMac` calcola il tag HMAC e verifica che sia identico al tag HMAC fornito. Se il tag HMAC inserito non corrisponde a quello calcolato, la verifica non riesce.

Le chiavi KMS HMAC non supportano la [rotazione automatica delle chiavi](#) e non puoi creare una chiave KMS HMAC in un [archivio di chiavi personalizzate](#).

Se stai creando una chiave KMS per crittografare i dati in un AWS servizio, usa una chiave di crittografia simmetrica. Non puoi utilizzare una chiave KMS HMAC.

Regioni

Le chiavi HMAC KMS sono supportate in tutto ciò Regioni AWS che AWS KMS supporta.

Ulteriori informazioni

- Per creare chiavi HMAC KMS, vedi. [Creazione di una chiave KMS HMAC](#)
- Per creare chiavi HMAC KMS multiregionali, vedi. [Chiavi multiregionali in ingresso AWS KMS](#)
- Per esaminare la differenza nella politica delle chiavi predefinita impostata dalla AWS KMS console per le chiavi HMAC KMS, vedi. [the section called “Consente agli utenti della chiave di utilizzare una chiave KMS per le operazioni di crittografia”](#)
- Per identificare e visualizzare le chiavi HMAC KMS, vedere. [Identifica le chiavi HMAC KMS](#)
- Per saperne di più sull'utilizzo HMACs per creare token web JSON, consulta [How to protect HMACs inside AWS KMS](#) nel Security Blog.AWS
- Ascolta un podcast: [Introduzione a The HMACs AWS Key Management ServiceOfficial](#) Podcast. AWS

Chiavi ML-DSA AWS KMS

AWS Key Management Service (AWS KMS) supporta l'algoritmo di firma digitale Module-Lattice (ML-DSA) per firme crittografiche post-quantistiche. Questa implementazione segue lo [standard Federal Information Processing Standards \(FIPS\) 204](#) per aiutare a proteggere dalle future minacce informatiche quantistiche. AWS KMS crea e protegge tutte le chiavi ML-DSA e le operazioni di firma nei moduli di sicurezza hardware convalidati FIPS 140-3 Security Level 3. Per aiutare a bilanciare sicurezza e prestazioni, ML-DSA in AWS KMS offre tre livelli di sicurezza distinti attraverso diverse specifiche chiave, ML_DSA_44, ML_DSA_65 e ML_DSA_87.

AWS KMS supporta firme a chiave RAW asimmetriche per messaggi fino a 4 KB utilizzando il tipo di messaggio. Per messaggi più grandi, è necessario calcolare esternamente la rappresentazione del messaggio a 64 byte μ utilizzata nella firma ML-DSA come definita nella sezione 6.2 del NIST FIPS 204. [Utilizzate il tipo di EXTERNAL_μ messaggio nell'operazione Sign per specificare questo messaggio preelaborato da 64 byte. AWS KMS](#) Le firme prodotte dal comando μ calcolato esternamente sono le stesse di RAW quelle prodotte quando si utilizza lo stesso messaggio e la stessa chiave privata. Nota che questa firma è diversa dalla ML-DSA o HashML-DSA «pre-hash» della sezione 5.4 del NIST FIPS 204.

Per ulteriori informazioni sull'uso di ML-DSA e del tipo di messaggio EXTERNAL_MU, vedere.

[Specifiche principali di ML-DSA](#)

Per un esempio di utilizzo di ML-DSA e del tipo di messaggio EXTERNAL_MU, vedere. [Verifica offline con coppie di chiavi ML-DSA](#)

Chiavi multiregionali in ingresso AWS KMS

AWS KMS supporta chiavi multiregionali, che sono disponibili AWS KMS keys in diverse Regioni AWS regioni e possono essere utilizzate in modo intercambiabile, come se avessi la stessa chiave in più regioni. Ogni set di chiavi multiregionali correlate ha lo stesso materiale chiave e lo stesso [ID](#) di chiave, quindi puoi crittografare i dati in una Regione AWS e decrittografarli in un'altra Regione AWS senza doverli crittografare nuovamente o effettuare una chiamata interregionale a. AWS KMS

Come tutte le chiavi KMS, le chiavi multiregionali non escono mai non crittografate. AWS KMS È possibile creare chiavi multi-regione simmetriche o asimmetriche per la crittografia o la firma, creare chiavi multi-regione HMAC per la generazione e la verifica dei tag HMAC e creare [chiavi multi-regione con materiale della chiave importato](#) o materiale della chiave generato da AWS KMS . È necessario gestire ogni chiave multi-regione in modo indipendente, inclusa la creazione di alias e tag, l'impostazione delle policy chiave e delle concessioni e l'abilitazione e la disabilitazione selettiva. È possibile utilizzare chiavi multi-regione in tutte le operazioni di crittografia che è possibile eseguire con le chiavi di singola regione.

Le chiavi multi-regione sono una soluzione flessibile e potente per molti scenari comuni di sicurezza dei dati.

Ripristino di emergenza

In un'architettura di backup e ripristino, le chiavi multiregionali consentono di elaborare i dati crittografati senza interruzioni anche in caso di interruzione. Regione AWS I dati mantenuti nelle regioni di backup possono essere decrittati nella regione di backup e i dati appena crittografati nella regione di backup possono essere decrittati nella regione principale quando tale regione viene ripristinata.

Gestione globale dei dati

Le aziende che operano a livello globale necessitano di dati distribuiti a livello globale e che siano disponibili in modo coerente in Regioni AWS. È possibile creare chiavi multi-regione in tutte le aree geografiche in cui risiedono i dati, quindi utilizzare le chiavi come se fossero una chiave singola regione senza la latenza di una chiamata tra regioni diverse o il costo di una nuova crittografia dei dati sotto una chiave diversa in ogni regione.

Applicazioni per la firma distribuita

Le applicazioni che richiedono funzionalità di firma tra regioni diverse possono utilizzare chiavi di firma asimmetriche multi-regione per generare firme digitali identiche in modo coerente e ripetuto in diverse Regioni AWS.

Se si utilizza il concatenamento dei certificati con un unico archivio affidabile globale (per una singola autorità di certificazione principale (CA) e un sistema intermedio regionale CAs firmato dalla CA principale, non sono necessarie chiavi multiregionali. Tuttavia, se il sistema non supporta funzionalità intermedie CAs, come la firma delle applicazioni, puoi utilizzare chiavi multiregionali per garantire coerenza alle certificazioni regionali.

Applicazioni in modalità attivo-attivo che si estendono su più regioni

Alcuni carichi di lavoro e applicazioni possono estendersi su più regioni in architetture di modalità attivo-attivo. Per queste applicazioni, le chiavi multi-regione possono ridurre la complessità fornendo lo stesso materiale chiave per le operazioni simultanee di crittografia e decrittografia sui dati che potrebbero essere spostati oltre i confini della regione.

[Puoi utilizzare chiavi multiregionali con librerie di crittografia lato client, come AWS Database Encryption SDK e la crittografia lato client Amazon S3. AWS Encryption SDK](#)

[AWS i servizi che si integrano con AWS KMS](#) la crittografia a riposo o le firme digitali attualmente trattano le chiavi multiregione come se fossero chiavi a regione singola. Potrebbero riavvolgere o crittografare i dati spostati tra le regioni. Ad esempio, la replica tra regioni di Amazon S3 decrittata e crittografia nuovamente i dati sotto una chiave KMS nell'area di destinazione, anche durante la replica di oggetti protetti da una chiave multi-regione.

Le chiavi multi-regione non sono globali. Creare una chiave primaria multi-regione e quindi replicarla in Regioni selezionate all'interno di una [partizione AWS](#). Puoi quindi gestire la chiave multi-regione in ogni regione in modo indipendente. Né crea AWS né AWS KMS replica automaticamente chiavi multiregionali in alcuna regione per conto dell'utente. [Chiavi gestite da AWS](#), le chiavi KMS che AWS i servizi creano per te nel tuo account sono sempre chiavi per regione singola.

Nelle regioni della Cina, puoi utilizzare la funzionalità chiave multiregionale per replicare le chiavi KMS all'interno della partizione China Regions (). `aws-cn` Ad esempio, è possibile replicare una chiave dalla regione Cina (Pechino) alla regione Cina (Ningxia) o viceversa. Replicando una chiave da una regione della Cina a un'altra, l'utente accetta di utilizzare la regione AWS Key Management Service di destinazione e di rispettare tutti i termini del contratto applicabili per la regione di

destinazione. Non è possibile replicare una chiave dalle regioni di Pechino e Ningxia in una AWS regione al di fuori della partizione delle regioni della Cina. Allo stesso modo, non è possibile replicare una chiave da una regione al di fuori della partizione delle regioni della Cina nelle regioni di Pechino e Ningxia.

Non è possibile convertire una chiave di regione singola esistente in una chiave multi-regione. Questo design garantisce che tutti i dati protetti con le chiavi esistenti di regione singola mantengano le stesse proprietà di residenza e sovranità dei dati.

Per la maggior parte delle esigenze di sicurezza dei dati, l'isolamento regionale e la tolleranza agli errori delle risorse regionali rendono le chiavi standard per AWS KMS regione singola la soluzione più adatta. Tuttavia, quando è necessario crittografare o firmare i dati in applicazioni lato client in più regioni, è possibile che le chiavi multi-regione siano la soluzione.

Regioni

Le chiavi multiregionali sono supportate in tutti i Regioni AWS supporti. AWS KMS

Prezzi e quote

Ogni chiave di un set di chiavi multi-regione viene conteggiata come una chiave KMS per i prezzi e le quote. Le [quote di AWS KMS](#) sono calcolate separatamente per ogni regione di un account. L'utilizzo e la gestione delle chiavi multi-regione in ogni regione conteggiano per le quote per quella regione.

Tipi di chiavi KMS supportati

È possibile creare i seguenti tipi di chiavi KMS multiregione:

- Chiavi KMS di crittografia simmetrica
- Chiavi KMS asimmetriche
- Chiavi KMS HMAC
- Chiavi KMS con materiale della chiave importato

Non è possibile creare chiavi multi-regione in un archivio delle chiavi personalizzate.

Ulteriori informazioni

- Per informazioni su come controllare l'accesso alle chiavi KMS multiregionali, consulta. [Controlla l'accesso alle chiavi multiregionali](#)
- Per creare chiavi KMS primarie multiregionali di qualsiasi tipo, consulta. [Crea chiavi primarie multiregionali](#)
- Per creare chiavi KMS di replica multiregionale, vedi. [Crea chiavi di replica multiregionali](#)
- Per aggiornare la regione principale, vedere. [Cambia la chiave primaria in un set di chiavi multiregionali](#)
- Per identificare e visualizzare le chiavi KMS multiregionali, vedi. [Identifica le chiavi HMAC KMS](#)
- Per ulteriori informazioni sulle considerazioni speciali sull'eliminazione delle chiavi KMS multiregionali, consulta. [Deleting multi-Region keys](#)

Concetti e terminologia

I termini e i concetti seguenti sono utilizzati con le chiavi multi-regione.

Chiave multi-regione

Una chiave multi-regione è una di un set di chiavi KMS con lo stesso ID chiave e materiale chiave (e altre [proprietà condivise](#)) in diversi Regioni AWS. Ogni chiave multi-regione è una chiave KMS completamente funzionante che può essere utilizzata indipendentemente dalle relative chiavi multi-regione correlate. Poiché tutte le chiavi multiregione correlate hanno lo stesso ID di chiave e lo stesso materiale di chiave, sono interoperabili, ovvero qualsiasi chiave multiregionale correlata in qualsiasi area Regione AWS può decrittografare il testo cifrato crittografato da qualsiasi altra chiave multiregionale correlata.

Puoi impostare la proprietà multi-regione di una chiave KMS al momento della sua creazione. Non è possibile modificare la proprietà multi-Regione su una chiave esistente. Non è possibile convertire una chiave a Regione singola in chiave multi-Regione o convertire una chiave multi-Regione in una chiave a Regione singola. Per spostare i carichi di lavoro esistenti in scenari multi-Regione, è necessario crittografare nuovamente i dati o creare nuove firme con nuove chiavi multi-Regione.

[Una chiave multiregionale può essere simmetrica o asimmetrica e può utilizzare materiale chiave o materiale chiave importato. AWS KMS](#) Non è possibile creare chiavi multi-regione in un [archivio delle chiavi personalizzate](#).

In una serie di chiavi multi-regione correlate, c'è esattamente una [chiave primaria](#) in qualsiasi momento. È possibile creare [chiavi di replica](#) di quella chiave primaria in altre Regioni AWS. È

possibile anche [aggiornare l'area principale](#), che modifica la chiave primaria in una chiave di replica e modifica una chiave di replica specificata nella chiave primaria. Tuttavia, è possibile mantenere una sola chiave primaria o chiave di replica per ciascuna. Regione AWS Tutte le regioni devono trovarsi nella stessa [partizione AWS](#).

È possibile avere più set di chiavi multi-regione correlate nello stesso o in un diverso Regioni AWS. Sebbene le chiavi multi-regione correlate siano interoperabili, le chiavi multi-regione non correlate non sono interoperabili.

Chiave primaria

Una chiave primaria multiregionale è una chiave KMS che può essere replicata in altre Regioni AWS nella stessa partizione. Ogni set di chiavi multi-regione ha una sola chiave primaria.

Una chiave primaria differisce da una chiave di replica nei seguenti modi:

- Solo una chiave primaria può essere [replicata](#).
- La chiave primaria è la fonte per le [proprietà condivise](#) delle sue [chiavi di replica](#), incluso il materiale della chiave e l'ID chiave.
- Puoi abilitare e disabilitare la [rotazione automatica delle chiavi](#) solo su una chiave primaria.
- È possibile [pianificare l'eliminazione di una chiave primaria](#) in qualsiasi momento. Ma non AWS KMS eliminerà una chiave primaria finché non verranno eliminate tutte le relative chiavi di replica.

Tuttavia, le chiavi primarie e di replica non differiscono in alcuna proprietà crittografica. È possibile utilizzare una chiave primaria e le relative chiavi di replica in modo intercambiabile.

Non è necessario replicare una chiave primaria. È possibile utilizzarlo come qualsiasi chiave KMS e replicarlo se e quando è utile. Tuttavia, poiché le chiavi multi-regione dispongono di proprietà di protezione diverse dalle chiavi di regione singola, si consiglia di creare una chiave multi-regione solo quando si prevede di replicarla.

Chiave di replica

Una chiave di replica multiregionale è una chiave KMS che ha lo stesso [ID e lo stesso materiale chiave](#) della chiave [primaria e delle relative chiavi di replica, ma esiste in una chiave](#) diversa. Regione AWS

Una chiave di replica è una chiave KMS completamente funzionale con policy di chiave, privilegi, alias, tag e altre proprietà. Non è una copia o un puntatore alla chiave primaria o a qualsiasi altra

chiave. È possibile utilizzare una chiave di replica anche se la chiave primaria e tutte le chiavi di replica correlate sono disabilitate. È inoltre possibile convertire una chiave di replica in una chiave primaria e una chiave primaria in una chiave di replica. Una volta creata, una chiave di replica si basa sulla sua chiave primaria solo per [rotazione delle chiavi](#) e [aggiornamento della regione primaria](#).

Le chiavi primarie e di replica non differiscono nelle proprietà crittografiche. È possibile utilizzare una chiave primaria e le relative chiavi di replica in modo intercambiabile. I dati crittografati da una chiave primaria o di replica possono essere decrittati dalla stessa chiave o da qualsiasi chiave primaria o di replica correlata.

Replica

È possibile replicare una [chiave primaria](#) multiregionale in un'altra nella stessa partizione. Regione AWS Quando lo fai, AWS KMS crea una [chiave di replica](#) multiregionale nella regione specificata con lo stesso [ID di chiave](#) e altre [proprietà condivise](#) della chiave primaria. Quindi trasporta in modo sicuro il materiale chiave attraverso il confine della regione e lo associa alla nuova chiave di replica, il tutto all'interno di AWS KMS.

Proprietà condivise

Le proprietà condivise sono proprietà di una chiave primaria multiregionale condivise con le relative chiavi di replica. AWS KMS crea le chiavi di replica con gli stessi valori di proprietà condivisi di quelli della chiave primaria. Quindi, sincronizza periodicamente i valori delle proprietà condivise della chiave primaria con le relative chiavi di replica. Non è possibile impostare queste proprietà su una chiave di replica.

Di seguito sono riportate le proprietà condivise delle chiavi multi-regione.

- [ID chiave](#) — (L'elemento Region del [ARN della chiave](#) differisce.)
- [Materiale della chiave](#)
- [Origine del materiale della chiave](#)
- [Specifica della chiave](#) e algoritmi di crittografia
- [Utilizzo delle chiavi](#)
- [Rotazione automatica delle chiavi](#), è possibile abilitare e disabilitare la rotazione automatica delle chiavi solo sulla chiave primaria. Le nuove chiavi di replica vengono create con tutte le versioni del materiale della chiave condivisa. Per informazioni dettagliate, consultare [Rotating multi-Region keys](#).

- [Rotazione su richiesta](#): è possibile eseguire la rotazione su richiesta solo sulla chiave primaria. Le nuove chiavi di replica vengono create con tutte le versioni del materiale della chiave condivisa. Per informazioni dettagliate, consultare [Rotating multi-Region keys](#).

È inoltre possibile considerare le designazioni primarie e di replica delle chiavi multi-regione correlate come proprietà condivise. Quando si [creano nuove chiavi di replica](#) o si [aggiorna la chiave primaria](#), AWS KMS sincronizza la modifica con tutte le chiavi multiregionali correlate. Una volta completate queste modifiche, tutte le chiavi multi-regione elencano in modo accurato la chiave primaria e le chiavi di replica.

Tutte le altre proprietà delle chiavi multi-regione sono proprietà indipendenti, compresa la descrizione, la [policy delle chiavi](#), le [concessioni](#), gli [stati chiave abilitati e disabilitati](#), gli [alias](#), e i [tag](#). Puoi impostare gli stessi valori per queste proprietà su tutte le chiavi multi-regione correlate, ma se si modifica il valore di una proprietà indipendente, AWS KMS non lo sincronizza.

È possibile tenere traccia della sincronizzazione delle proprietà condivise delle chiavi multi-regione. Nel tuo AWS CloudTrail registro, cerca l'evento. [SynchronizeMultiRegionKey](#)

Considerazioni sulla protezione per le chiavi multi-regione

Usa una chiave AWS KMS multiregionale solo quando ne hai bisogno. Le chiavi multi-regione offrono una soluzione flessibile e scalabile per carichi di lavoro che spostano dati crittografati tra Regioni AWS o hanno bisogno di un accesso tra regioni. Considera una chiave multi-regione se hai bisogno di condividere, spostare o eseguire il backup di dati protetti tra regioni o creare firme digitali identiche di applicazioni che operano in diverse regioni.

Tuttavia, il processo di creazione di una chiave multi-regione sposta il materiale chiave su limiti Regione AWS all'interno di AWS KMS. Il testo cifrato generato da una chiave multi-regione può potenzialmente essere decrittato da più chiavi correlate in più posizioni geografiche. I servizi e le risorse isolate a livello regionale offrono vantaggi significativi. Ogni Regione AWS è isolata e indipendente dalle altre regioni. Le regioni forniscono la tolleranza ai guasti, la stabilità e la resilienza e possono anche ridurre la latenza. Consentono di creare risorse ridondanti che restano disponibili e non influenzate da un'interruzione in un'altra regione. Inoltre AWS KMS, assicurano che ogni testo cifrato possa essere decrittografato con una sola chiave.

Le chiavi multi-regione sollevano anche nuove considerazioni sulla sicurezza:

- Il controllo dell'accesso e l'applicazione della policy di sicurezza dei dati è più complesso con le chiavi multi-regione. È necessario assicurarsi che la policy sia controllata in modo coerente sulla

chiave in più aree isolate. E devi usare la policy per applicare i limiti, invece di fare affidamento su chiavi separate.

Ad esempio, è necessario impostare le condizioni di policy sui dati per impedire ai team del ciclo paghe di una regione di leggere i dati del ciclo paghe per una regione diversa. Inoltre, è necessario utilizzare il controllo di accesso per impedire uno scenario in cui una chiave multi-regione in una regione protegge i dati di un tenant e una chiave multi-regione correlata in un'altra regione protegge i dati di un tenant diverso.

- Anche la verifica delle chiavi in tutte le regioni è più complesso. Con le chiavi multi-regione, è necessario esaminare e riconciliare le attività di verifica in più regioni per ottenere una comprensione completa delle attività chiave sui dati protetti.
- La conformità ai requisiti di residenza dei dati può essere più complessa. Con le regioni isolate, è possibile garantire la residenza dei dati e la conformità alla sovranità dei dati. Le chiavi KMS in una determinata regione possono decrittare i dati sensibili solo in tale regione. I dati crittografati in una regione possono rimanere completamente protetti e inaccessibili in qualsiasi altra regione.

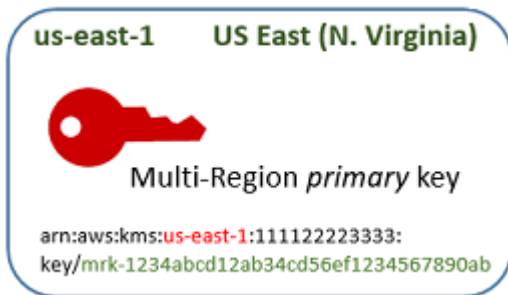
Per verificare la residenza e la sovranità dei dati con chiavi multiregionali, è necessario implementare politiche di accesso e compilare eventi in più regioni. AWS CloudTrail

[Per semplificare la gestione del controllo degli accessi sulle chiavi multiregionali, l'autorizzazione a replicare una chiave multiregionale \(kms:ReplicateKey\) è separata dall'autorizzazione standard per la creazione di chiavi \(kms:\). CreateKey](#) Inoltre, AWS KMS supporta diverse condizioni politiche per le chiavi multiregionali `kms:MultiRegion`, tra cui la possibilità di consentire o negare l'autorizzazione a creare, utilizzare o gestire chiavi multiregionali e la limitazione delle regioni in cui è possibile `kms:ReplicaRegion` replicare una chiave multiregionale. Per informazioni dettagliate, consultare [Controlla l'accesso alle chiavi multiregionali](#).

Come funzionano le chiavi multi-regione

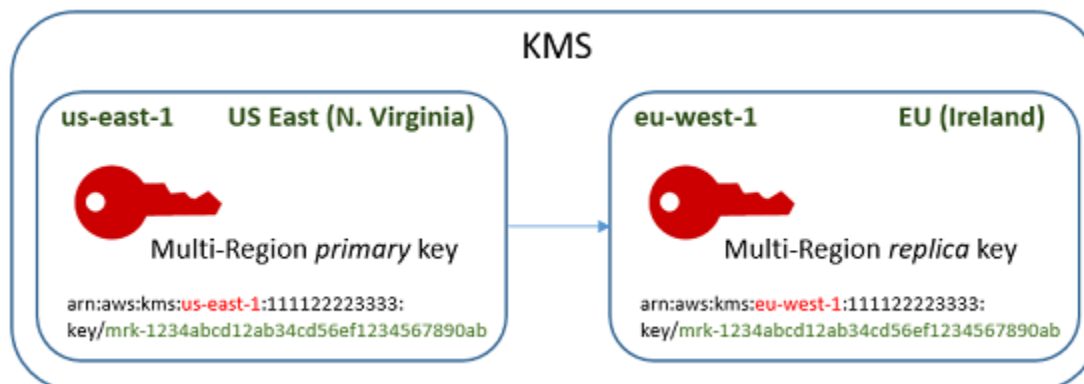
Si inizia creando una [chiave primaria multiregionale](#) simmetrica o asimmetrica in un sistema che supporti, ad esempio Stati Uniti orientali (Virginia settentrionale). Regione AWS AWS KMS È possibile decidere se una chiave è di regione singola o multi-regione solo al momento della creazione. Non è possibile modificare questa proprietà in un secondo momento. Come per qualsiasi chiave KMS, è possibile impostare una policy delle chiavi per la chiave multi-regione ed è possibile creare concessioni e aggiungere alias e tag per la categorizzazione e l'autorizzazione. (Queste sono [proprietà indipendenti](#) che non sono condivise o sincronizzate con altre chiavi.) È possibile utilizzare la chiave primaria multi-regione nelle operazioni di crittografia per la crittografia o la firma.

È possibile [creare una chiave primaria multiregionale nella](#) AWS KMS console o utilizzando l'API con il parametro impostato su `CreateKeyMultiRegiontrue`. Si noti che le chiavi multi-regione hanno un ID chiave distintivo che inizia con `mrk-`. È possibile utilizzare il `mrk-` prefisso per l'identificazione MRKs a livello di codice.



Se lo desideri, puoi [replicare](#) la chiave primaria multiregionale in una o più diverse Regioni AWS nella stessa [AWS partizione](#), ad esempio Europa (Irlanda). Quando lo fai, AWS KMS crea una [chiave di replica](#) nella regione specificata con lo stesso ID di chiave e altre [proprietà condivise](#) della chiave primaria. Trasporta in modo sicuro il materiale chiave attraverso il confine della regione e lo associa alla nuova chiave KMS nella regione di destinazione, il tutto all'interno di AWS KMS. Il risultato è due chiavi multi-regione correlate, una chiave primaria e una chiave di replica, che possono essere utilizzate in modo intercambiabile.

È possibile [creare una chiave di replica multiregionale](#) nella AWS KMS console o utilizzando l'API. [ReplicateKey](#)



La risultante [chiave di replica multi-regione](#) è una chiave KMS completamente funzionale con le stesse [proprietà condivise](#) come chiave primaria. Per tutti gli altri aspetti, si tratta di una chiave KMS indipendente con descrizione, policy delle chiavi, concessioni, alias e tag propri. L'attivazione o la disattivazione di una chiave multi-regione non ha alcun effetto sulle chiavi multi-regione. È possibile utilizzare le chiavi primarie e di replica in modo indipendente nelle operazioni di crittografia o coordinarne l'utilizzo. Ad esempio, è possibile crittografare i dati con la chiave primaria nella regione

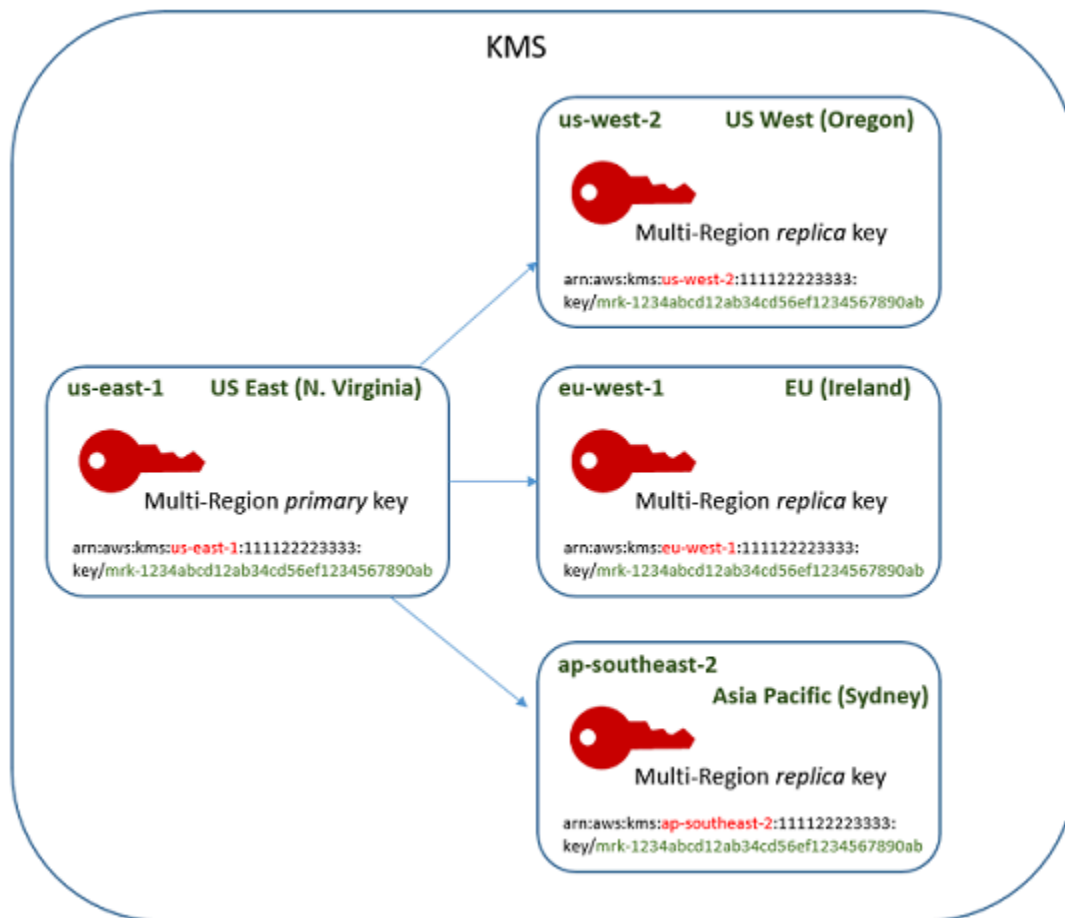
Stati Uniti orientali (Virginia settentrionale), spostare i dati nella regione Europa (Irlanda) e utilizzare la chiave di replica per decrittare i dati.

Le chiavi multi-regione correlate hanno lo stesso ID chiave. La loro chiave ARNs (Amazon Resource Names) differisce solo nel campo Regione. Ad esempio, la chiave primaria multiregionale e le chiavi di replica potrebbero avere la seguente chiave di esempio. ARNs L'ID chiave, l'ultimo elemento nell'ARN della chiave, è identico. Entrambe le chiavi hanno l'ID chiave distintivo delle chiavi multi-regione, che inizia con `mrk-`.

```
Primary key: arn:aws:kms:us-east-1:111122223333:key/mrk-1234abcd12ab34cd56ef12345678990ab
Replica key: arn:aws:kms:eu-west-1:111122223333:key/mrk-1234abcd12ab34cd56ef12345678990ab
```

Per l'interoperabilità è necessario disporre dello stesso ID chiave. Durante la crittografia, AWS KMS associa l'ID della chiave KMS al testo cifrato in modo che il testo cifrato possa essere decrittografato solo con quella chiave KMS o con una chiave KMS con lo stesso ID di chiave. Questa caratteristica rende anche facili da riconoscere le chiavi multi-regione correlate e rende più facile utilizzarle in modo intercambiabile. Ad esempio, quando le si utilizzano in un'applicazione, è possibile fare riferimento alle chiavi multi-regione correlate tramite il relativo ID chiave condivisa. Quindi, se necessario, specificare la regione o l'ARN per distinguerli.

Man mano che le esigenze relative ai dati cambiano, puoi replicare la chiave primaria su altre Regioni AWS chiavi della stessa partizione, ad esempio Stati Uniti occidentali (Oregon) e Asia Pacifico (Sydney). Il risultato sono quattro chiavi multiregionali correlate con lo stesso materiale chiave e la stessa chiave IDs, come illustrato nel diagramma seguente. Gestisci le chiavi in modo indipendente. Puoi usarle indipendentemente o in modo coordinato. Ad esempio, è possibile crittografare i dati con la chiave di replica in Asia Pacifico (Sydney), spostare i dati in Stati Uniti occidentali (Oregon) e decrittarli con la chiave di replica in Stati Uniti occidentali (Oregon).



Altre considerazioni per le chiavi multi-regione includono le seguenti.

Sincronizzazione delle proprietà condivise: [se una proprietà condivisa delle chiavi multiregionali cambia, sincronizza AWS KMS automaticamente la modifica dalla chiave primaria a tutte le relative chiavi di replica](#). Non è possibile richiedere o forzare una sincronizzazione delle proprietà condivise. AWS KMS rileva e sincronizza tutte le modifiche per te. Tuttavia, è possibile controllare la sincronizzazione utilizzando l'[SynchronizeMultiRegionKey](#) evento nei registri. CloudTrail

Ad esempio, se abiliti la rotazione automatica delle chiavi su una chiave primaria simmetrica multiregionale, AWS KMS copia tale impostazione su tutte le relative chiavi di replica. Quando il materiale chiave viene ruotato, la rotazione viene sincronizzata tra tutte le chiavi multi-regione correlate, in modo che continuino ad avere lo stesso materiale chiave corrente e ad accedere a tutte le versioni precedenti del materiale chiave. Se si crea una nuova chiave di replica, la chiave contiene lo stesso materiale corrente di tutte le chiavi multi-regione correlate e l'accesso a tutte le versioni precedenti del materiale chiave. Per dettagli, consulta [Rotating multi-Region keys](#).

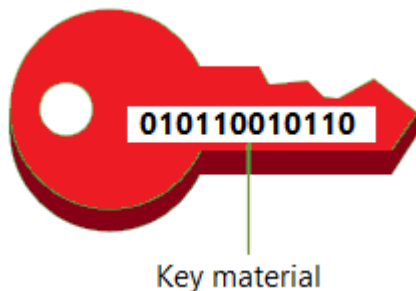
Modifica della chiave primaria — Ogni set di chiavi multi-regione deve avere esattamente una chiave primaria. La [chiave primaria](#) è l'unica chiave che può essere replicata. È anche la fonte delle proprietà condivise delle chiavi di replica. Tuttavia, è possibile modificare la chiave primaria in una replica e promuovere una delle chiavi di replica in primaria. È possibile eseguire questa operazione in modo da poter eliminare una chiave primaria per più aree da una determinata regione o individuare la chiave primaria in una regione più vicina agli amministratori di progetto. Per informazioni dettagliate, consultare [Cambia la chiave primaria in un set di chiavi multiregionali](#).

Eliminazione delle chiavi multiregionali: come tutte le chiavi KMS, è necessario pianificare l'eliminazione delle chiavi multiregionali prima di eliminarle. AWS KMS Mentre la chiave è in attesa di eliminazione, non è possibile utilizzarla in nessuna operazione di crittografia. Tuttavia, non AWS KMS eliminerà una chiave primaria multiregionale finché non verranno eliminate tutte le relative chiavi di replica. Per informazioni dettagliate, consulta [Deleting multi-Region keys](#).

Importazione di materiale chiave per le AWS KMS chiavi

Puoi creare una AWS KMS keys (chiave KMS) con il materiale chiave che fornisci.

Una chiave KMS è una rappresentazione logica di una chiave dati. I metadati di una chiave KMS includono l'ID del materiale chiave utilizzato per eseguire operazioni crittografiche. Quando [crei una chiave KMS](#), per impostazione predefinita, AWS KMS genera il materiale chiave per quella chiave KMS. Tuttavia è possibile creare una chiave KMS senza materiale della chiave e importare il proprio materiale in quella chiave KMS. Questa caratteristica è definita "bring your own key" (BYOK).



Note

AWS KMS non supporta la decrittografia di alcun AWS KMS testo cifrato crittografato da una chiave KMS di crittografia simmetrica al di fuori di AWS KMS, anche se il testo cifrato è stato crittografato con una chiave KMS con materiale chiave importato. AWS KMS non pubblica il formato di testo cifrato richiesto da questa attività e il formato potrebbe cambiare senza preavviso.

Quando si utilizza materiale chiave importato, l'utente rimane responsabile del materiale chiave, consentendone AWS KMS al contempo l'utilizzo di una copia. Potresti scegliere di farlo per uno o più dei seguenti motivi:

- Per dimostrare che il materiale della chiave è stato generato utilizzando una sorgente di entropia che soddisfa i tuoi requisiti.
- Utilizzare materiale chiave proveniente dalla propria infrastruttura con AWS servizi e utilizzarlo per AWS KMS gestire il ciclo di vita di tale materiale chiave all'interno. AWS
- Utilizzare chiavi esistenti e consolidate, ad esempio le chiavi per la firma del codice AWS KMS, la firma dei certificati PKI e le applicazioni con certificato bloccato
- Per impostare una scadenza per il materiale chiave AWS e per [eliminarlo manualmente](#), ma anche per renderlo nuovamente disponibile in futuro. La pianificazione, [dell'eliminazione di una chiave](#), invece, richiede un periodo di attesa compreso tra 7 e 30 giorni, trascorso il quale non puoi più recuperare la chiave KMS eliminata.
- Possedere la copia originale del materiale chiave e conservarla all'esterno AWS per una maggiore durabilità e ripristino di emergenza durante l'intero ciclo di vita del materiale chiave.
- Per le chiavi asimmetriche e le chiavi HMAC, l'importazione crea chiavi compatibili e interoperabili che funzionano all'interno e all'esterno di. AWS

Tipi di chiavi KMS supportati

AWS KMS supporta materiale chiave importato per i seguenti tipi di chiavi KMS. Non puoi importare il materiale della chiave in chiavi KMS in [archivi di chiavi personalizzate](#).

- [Chiavi KMS di crittografia simmetrica](#)
- [Chiavi KMS asimmetriche \(eccetto le chiavi ML-DSA\)](#)
- [Chiavi KMS HMAC](#)
- [Chiavi multi-regione](#) di tutti i tipi supportati.

Regioni

Il materiale chiave importato è supportato in tutti i supporti. Regioni AWS AWS KMS

Nelle regioni della Cina, i requisiti materiali chiave per le chiavi KMS con crittografia simmetrica sono diversi da quelli delle altre regioni. Per informazioni dettagliate, vedi [Fase 3: crittografare il materiale delle chiavi](#).

Ulteriori informazioni

- Per creare chiavi KMS con materiale chiave importato, vedi. [Crea una chiave KMS con materiale chiave importato](#)
- Per creare un avviso che ti avvisi quando il materiale chiave importato in una chiave KMS sta per scadere, vedi. [Crea un CloudWatch allarme per la scadenza del materiale chiave importato](#)
- Per reimportare il materiale chiave in una chiave KMS, vedi. [Reimporta il materiale chiave](#)
- Per importare nuovo materiale chiave in una chiave KMS per la rotazione su richiesta, vedi e. [Importa nuovo materiale chiave](#) [Esegui la rotazione dei tasti su richiesta](#)
- Per identificare e visualizzare le chiavi KMS con materiale chiave importato, vedi. [Identifica le chiavi KMS con materiale chiave importato](#)
- Per ulteriori informazioni sulle considerazioni speciali sull'eliminazione delle chiavi KMS con materiale chiave importato, consulta. [Deleting KMS keys with imported key material](#)

Considerazioni speciali per il materiale chiave importato

Prima di decidere di importare materiale chiave in AWS KMS, è necessario comprendere le seguenti caratteristiche del materiale chiave importato.

Generare il materiale della chiave

Sei responsabile della generazione del materiale della chiave utilizzando una fonte di casualità che soddisfa i tuoi requisiti di sicurezza.

Sei responsabile della disponibilità e della durata

AWS KMS è progettato per garantire un'elevata disponibilità del materiale chiave importato. Tuttavia, AWS KMS non mantiene la durabilità del materiale chiave importato allo stesso livello del materiale chiave che AWS KMS genera. Per informazioni dettagliate, vedi [Protezione del materiale della chiave importato](#).

Puoi eliminare il materiale chiave

Puoi eliminare il [materiale della chiave importato](#) da una chiave KMS, rendendo immediatamente inutilizzabile la chiave KMS. Quando importi il materiale della chiave in una chiave KMS, puoi determinare se la chiave scade e [impostare la data di scadenza](#). Quando arriva la data di scadenza, AWS KMS [elimina il materiale chiave](#). Senza materiale chiave, la chiave KMS non può essere utilizzata in nessuna operazione di crittografia. Per ripristinare la chiave, è necessario importare nuovamente lo stesso materiale chiave nella chiave.

Non è possibile modificare il materiale delle chiavi per le chiavi asimmetriche e HMAC

Quando importi materiale della chiave in una chiave KMS, la chiave KMS viene associata in modo permanente a quel materiale della chiave. Puoi importare [nuovamente lo stesso materiale della chiave](#), ma non puoi importare materiale della chiave diverso in quella chiave KMS. Inoltre, non puoi [abilitare la rotazione automatica](#) delle chiavi per una chiave KMS con materiale della chiave importato. Tuttavia, puoi [ruotare manualmente una chiave KMS](#) con materiale della chiave importato.

È possibile eseguire la rotazione su richiesta su chiavi di crittografia simmetriche

Le chiavi di crittografia simmetriche con materiale chiave importato supportano la rotazione su richiesta. È possibile [importare più materiali chiave](#) in queste chiavi e utilizzare la [rotazione su richiesta](#) per aggiornare il materiale chiave corrente. Il materiale chiave corrente viene utilizzato sia per la crittografia che per la decrittografia, ma altri materiali chiave (non correnti) possono essere utilizzati solo per la decrittografia.

Non puoi modificare l'origine del materiale della chiave

Le chiavi KMS progettate per il materiale chiave importato ha un valore di [origine](#) di EXTERNAL che non può essere modificato. Non è possibile convertire una chiave KMS per materiale chiave importato in modo da utilizzare materiale chiave proveniente da altre fonti, tra cui. AWS KMS Allo stesso modo, non è possibile convertire una chiave KMS con materiale AWS KMS chiave in una chiave progettata per il materiale chiave importato.

Non puoi esportare il materiale della chiave

Non è possibile esportare alcun materiale chiave importato. AWS KMS non può restituirti il materiale chiave importato in nessuna forma. È necessario conservare una copia del materiale chiave importato all'esterno AWS, preferibilmente in un gestore di chiavi, come un modulo di sicurezza hardware (HSM), in modo da poter reimportare il materiale chiave in caso di eliminazione o scadenza.

Puoi creare chiavi multi-regione con materiale della chiave importato

Le chiavi multi-regione con il materiale della chiave importato includono le funzionalità delle chiavi KMS con il materiale della chiave importato e possono interoperare tra Regioni AWS. Per creare una chiave multi-regione con il materiale della chiave importato, devi importare lo stesso materiale della chiave nella chiave KMS primaria e in ogni chiave di replica. Per ulteriori dettagli sull'importazione di materiali chiave per chiavi multiregionali, consulta. [Multi-Region keys](#)

Le chiavi asimmetriche e le chiavi HMAC sono portatili e interoperabili

È possibile utilizzare il materiale chiave asimmetrico e il materiale chiave HMAC all'esterno di AWS per interagire con AWS KMS chiavi con lo stesso materiale chiave importato.

A differenza del testo cifrato AWS KMS simmetrico, che è indissolubilmente legato alla chiave KMS utilizzata nell'algoritmo, AWS KMS utilizza formati HMAC standard e asimmetrici per la crittografia, la firma e la generazione MAC. Di conseguenza, le chiavi sono portatili e supportano gli scenari di chiavi di deposito tradizionali.

Quando la chiave KMS ha importato materiale chiave, puoi utilizzare il materiale chiave importato all'esterno per eseguire le seguenti operazioni. AWS

- Chiavi HMAC: puoi verificare un tag HMAC generato dalla chiave KMS HMAC con il materiale della chiave importato. Puoi anche utilizzare la chiave KMS HMAC con il materiale chiave importato per verificare un tag HMAC generato dal materiale chiave all'esterno di. AWS
- Chiavi di crittografia asimmetriche: puoi utilizzare la tua chiave di crittografia asimmetrica privata all'esterno di AWS per decrittografare un testo cifrato crittografato dalla chiave KMS con la chiave pubblica corrispondente. Puoi anche utilizzare la tua chiave KMS asimmetrica per decrittografare un testo cifrato asimmetrico generato all'esterno di. AWS
- Chiavi di firma asimmetriche: puoi utilizzare la tua chiave KMS di firma asimmetrica con materiale chiave importato per verificare le firme digitali generate dalla tua chiave di firma privata all'esterno di. AWS Puoi anche utilizzare la tua chiave di firma pubblica asimmetrica all'esterno di per verificare le firme generate dalla tua chiave KMS asimmetrica. AWS
- Chiavi di accordo con chiave asimmetrica: puoi utilizzare la tua chiave KMS con contratto a chiave asimmetrica con materiale chiave importato per ricavare segreti condivisi con un altro utente esterno. AWS

Se importi lo stesso materiale della chiave in chiavi KMS diverse nella stessa Regione AWS, anche queste chiavi sono interoperabili. Per creare chiavi KMS interoperabili in diversi formati, crea una chiave multiregionale con materiale chiave importato. Regioni AWS

Chiavi private RSA

- AWS KMS richiede che le chiavi private RSA importate abbiano fattori primi conformi al test descritto in [FIPS 186-5](#), Sezione A. 1.3. Altri software o dispositivi possono utilizzare algoritmi diversi per convalidare questi fattori primi delle chiavi private RSA. In rari casi, le chiavi convalidate utilizzando altri algoritmi potrebbero non essere accettate da. AWS KMS

Le chiavi di crittografia simmetriche non sono portatili né interoperabili

I testi cifrati simmetrici che produce non sono portatili o interoperabili AWS KMS. AWS KMS non pubblica il formato di testo cifrato simmetrico richiesto dalla portabilità e il formato potrebbe cambiare senza preavviso.

- AWS KMS non è in grado di decrittografare testi cifrati simmetrici crittografati all'esterno, anche se si utilizza materiale chiave importato. AWS
- AWS KMS non supporta la decrittografia di alcun testo cifrato AWS KMS simmetrico al di fuori di, anche se il testo cifrato è stato crittografato con una chiave KMS con AWS KMS materiale chiave importato.
- Le chiavi KMS con lo stesso materiale della chiave importato non sono interoperabili. Il testo cifrato simmetrico che genera testo cifrato specifico per ogni chiave KMS. AWS KMS Questo formato di testo criptato garantisce che solo la chiave KMS che ha crittografato i dati possa decrittografarli.

Inoltre, non è possibile utilizzare AWS strumenti, come la [crittografia lato client di Amazon S3](#) [AWS Encryption SDKo Amazon S3, per decrittografare testi cifrati](#) simmetrici. AWS KMS

Di conseguenza, non è possibile utilizzare chiavi con materiale chiave importato per supportare accordi di deposito di chiavi in cui una terza parte autorizzata con accesso condizionato al materiale chiave possa decrittografare determinati testi cifrati all'esterno. AWS KMS Per supportare il deposito delle chiavi, utilizza il [AWS Encryption SDK](#) per crittografare il messaggio in una chiave indipendente da AWS KMS.

Protezione del materiale della chiave importato

Il materiale della chiave importato è protetto durante il transito e a riposo. Prima di importare il materiale chiave, si crittografa (o «avvolge») il materiale chiave con la chiave pubblica di una coppia di chiavi RSA generata nei moduli di sicurezza AWS KMS hardware (HSMs) convalidati secondo il programma di convalida dei moduli crittografici [FIPS 140-3](#). Puoi crittografare il materiale della chiave direttamente con la chiave pubblica di wrapping oppure crittografare il materiale della chiave con una chiave simmetrica AES e quindi crittografare la chiave simmetrica AES con la chiave pubblica RSA.

Al ricevimento, AWS KMS decripta il materiale chiave con la chiave privata corrispondente in un AWS KMS HSM e lo cripta nuovamente con una chiave simmetrica AES che esiste solo nella memoria volatile dell'HSM. Il materiale della chiave non esce mai in testo normale dal modulo HSM. Viene decrittografato solo mentre è in uso e solo all'interno. AWS KMS HSMs

L'uso della chiave KMS con il materiale della chiave importato è determinato esclusivamente dalle [policy di controllo degli accessi](#) che hai impostato sulla chiave KMS. Inoltre, puoi utilizzare [alias](#) e [tag](#) per identificare e [controllare l'accesso](#) alla chiave KMS. È possibile [abilitare e disabilitare](#) la chiave, [visualizzarla](#) e [monitorarla](#) utilizzando servizi come AWS CloudTrail

Ciononostante, conserva solo la copia sicura del materiale della chiave. In cambio di questa ulteriore misura di controllo, sei responsabile della durabilità e della disponibilità complessiva del materiale chiave importato. AWS KMS è progettato per garantire un'elevata disponibilità del materiale chiave importato. Tuttavia, AWS KMS non mantiene la durabilità del materiale chiave importato allo stesso livello del materiale chiave che AWS KMS genera.

Questa differenza di durabilità è significativa nei seguenti casi:

- Quando [impostate una scadenza](#) per il materiale chiave importato, AWS KMS elimina il materiale chiave dopo la sua scadenza. AWS KMS non elimina la chiave KMS o i relativi metadati. Puoi [creare un CloudWatch allarme Amazon](#) che ti avvisa quando il materiale chiave importato si avvicina alla data di scadenza.

Non puoi eliminare il materiale chiave AWS KMS generato per una chiave KMS e non puoi impostare la scadenza del materiale AWS KMS chiave.

- Quando [elimini manualmente il materiale chiave importato](#), AWS KMS elimina il materiale chiave ma non elimina la chiave KMS o i relativi metadati. Al contrario, [la pianificazione dell'eliminazione delle chiavi](#) richiede un periodo di attesa da 7 a 30 giorni, dopodiché elimina AWS KMS definitivamente la chiave KMS, i relativi metadati e il relativo materiale chiave.
- Nell'improbabile eventualità che si verifichino determinati guasti a livello regionale AWS KMS (ad esempio una perdita totale di alimentazione), AWS KMS non è possibile ripristinare automaticamente il materiale chiave importato. Tuttavia, AWS KMS può ripristinare la chiave KMS e i relativi metadati.

È necessario conservare una copia del materiale chiave importato all'esterno di AWS un sistema controllato dall'utente. Ti consigliamo di archiviare una copia esportabile del materiale della chiave importato in un sistema di gestione delle chiavi, ad esempio un modulo HSM. Come procedura consigliata, è consigliabile memorizzare un riferimento all'ARN della chiave KMS e all'ID del materiale chiave generato AWS KMS da insieme alla copia esportabile del materiale chiave. Se il materiale della chiave importato viene eliminato o scade, la chiave KMS associata diventa inutilizzabile fino a quando non importi nuovamente lo stesso materiale della chiave. Se il materiale della chiave

importato viene perso definitivamente, qualunque testo criptato crittografato con la chiave KMS è irrecuperabile.

Important

Alle chiavi di crittografia simmetriche possono essere associati più materiali chiave. L'intera chiave KMS diventa inutilizzabile non appena si elimina uno di questi materiali chiave o se uno di questi materiali chiave scade (a meno che il materiale chiave eliminato o in scadenza non sia o). `PENDING_ROTATION` `PENDING_MULTI_REGION_IMPORT_AND_ROTATION` È necessario reimportare qualsiasi materiale chiave scaduto o eliminato associato a tale chiave prima che la chiave diventi utilizzabile per operazioni crittografiche.

Chiavi KMS in un archivio di chiavi CloudHSM

È possibile creare, visualizzare, gestire, utilizzare e pianificare l'eliminazione delle chiavi AWS KMS keys in un archivio di AWS CloudHSM chiavi. Le procedure utilizzate sono molto simili a quelle impiegate per le altre chiavi KMS. L'unica differenza è che si specifica un archivio di AWS CloudHSM chiavi quando si crea la chiave KMS. Quindi, AWS KMS crea materiale chiave non estraibile per la chiave KMS nel AWS CloudHSM cluster associato all'archivio chiavi. AWS CloudHSM Quando si utilizza una chiave KMS in un archivio di AWS CloudHSM chiavi, le [operazioni crittografiche](#) vengono eseguite nel cluster. HSMs

Funzionalità supportate

Oltre alle procedure illustrate in questa sezione, è possibile effettuare le seguenti operazioni con le chiavi KMS in un AWS CloudHSM archivio di chiavi:

- Utilizzare le policy delle chiavi, le policy IAM e le concessioni per [autorizzare l'accesso](#) alle chiavi KMS.
- [Abilitare e disabilitare](#) le chiavi KMS.
- Assegnare [tag](#), creare [alias](#) e utilizzare il controllo degli accessi basato su attributi (ABAC) per autorizzare l'accesso alle chiavi KMS.
- Utilizza le chiavi KMS per eseguire le seguenti operazioni crittografiche:
 - [Encrypt](#)
 - [Decrypt](#)
 - [GenerateDataKey](#)

- [GenerateDataKeyWithoutPlaintext](#)
- [ReEncrypt](#)

Le operazioni che generano coppie di chiavi di dati asimmetriche [GenerateDataKeyPair](#) e [GenerateDataKeyPairWithoutPlaintext](#), non sono supportate negli archivi di chiavi personalizzati.

- Utilizzare le chiavi KMS con [servizi AWS che si integrano con AWS KMS](#) e supportano le chiavi gestite dal cliente.
- Tieni traccia dell'uso delle tue chiavi KMS nei [AWS CloudTrail log](#) e negli strumenti di [CloudWatch monitoraggio di Amazon](#).

Caratteristiche non supportate

- AWS CloudHSM gli archivi di chiavi supportano solo chiavi KMS con crittografia simmetrica. Non è possibile creare chiavi KMS HMAC, chiavi KMS asimmetriche o coppie di chiavi dati asimmetriche in un archivio di chiavi. AWS CloudHSM
- Non è possibile [importare materiale chiave in una chiave KMS](#) in un archivio di chiavi. AWS CloudHSM AWS KMS genera il materiale chiave per la chiave KMS nel AWS CloudHSM cluster.
- Non è possibile abilitare o disabilitare la [rotazione automatica](#) del materiale chiave per una chiave KMS in un archivio di AWS CloudHSM chiavi.

Utilizzo delle chiavi KMS in un archivio di AWS CloudHSM chiavi

Quando usi la tua chiave KMS in una richiesta, identifica la chiave KMS tramite il suo ID o alias; non è necessario specificare l'archivio delle AWS CloudHSM chiavi o il cluster. AWS CloudHSM La risposta include gli stessi campi che vengono restituiti per qualsiasi chiave KMS di crittografia simmetrica.

Tuttavia, quando si utilizza una chiave KMS in un archivio di AWS CloudHSM chiavi, l'operazione di crittografia viene eseguita interamente all'interno del AWS CloudHSM cluster associato all'archivio delle chiavi. AWS CloudHSM L'operazione utilizza il materiale della chiave nel cluster associato alla chiave KMS scelta.

Perché ciò avvenga, devono essere soddisfatte le seguenti condizioni.

- Lo [stato di chiave](#) della chiave KMS deve essere Enabled. Per trovare lo stato della chiave, usa il campo Status nella [AWS KMS console](#) o il KeyState campo nella [DescribeKey](#) risposta.

- L'archivio delle AWS CloudHSM chiavi deve essere connesso al relativo AWS CloudHSM cluster. Il relativo stato nella [AWS KMS console](#) o ConnectionState nella [DescribeCustomKeyStores](#) risposta deve essere CONNECTED.
- Il AWS CloudHSM cluster associato all'archivio chiavi personalizzato deve contenere almeno un HSM attivo. Per trovare il numero di persone attive HSMs nel cluster, usa la [AWS KMS console](#), la AWS CloudHSM console o l'[DescribeClusters](#) operazione.
- Il AWS CloudHSM cluster deve contenere il materiale chiave per la chiave KMS. Se questo materiale è stato eliminato dal cluster oppure un HSM è stato creato a partire da un backup che non includeva quel materiale, l'operazione di crittografia non riuscirà.

Se queste condizioni non sono soddisfatte, l'operazione di crittografia ha esito negativo e AWS KMS restituisce un `KMSInvalidStateException` eccezione. In genere, è sufficiente [ricollegare l'archivio delle AWS CloudHSM chiavi](#). Per ulteriori informazioni, consulta [Come correggere una chiave KMS non funzionante](#).

Quando utilizzi le chiavi KMS in un AWS CloudHSM archivio di chiavi, tieni presente che le chiavi KMS di ogni archivio AWS CloudHSM chiavi condividono una quota di [richieste di archiviazione chiavi personalizzata per le operazioni di crittografia](#). Se superi la quota, AWS KMS restituisce un `ThrottlingException`. Se il AWS CloudHSM cluster associato all'archivio AWS CloudHSM chiavi elabora numerosi comandi, inclusi quelli non correlati all'archivio AWS CloudHSM chiavi, potresti riceverne uno `ThrottlingException` a una velocità ancora inferiore. Se viene generata un'eccezione `ThrottlingException` per una qualsiasi richiesta, riduci la frequenza delle richieste e riesegui i comandi. Per informazioni dettagliate sulle quote di richiesta dell'archivio delle chiavi personalizzate, consulta [Quote di richiesta per l'archivio delle chiavi personalizzate](#).

Ulteriori informazioni

- Per ulteriori informazioni sui AWS CloudHSM key store, consulta [AWS CloudHSM negozi chiave](#).
- Per creare chiavi KMS in un archivio di AWS CloudHSM chiavi, vedi [Crea una chiave KMS in un AWS CloudHSM archivio di chiavi](#).
- Per identificare e visualizzare le chiavi KMS in un archivio di AWS CloudHSM chiavi, vedi [Identifica le chiavi KMS negli AWS CloudHSM archivi delle chiavi](#).
- Per trovare le chiavi KMS e il relativo materiale in un archivio di AWS CloudHSM chiavi, consulta [Trova le chiavi KMS e il materiale chiave in un archivio di AWS CloudHSM chiavi](#).

- Per informazioni su considerazioni speciali sull'eliminazione delle chiavi KMS in un archivio di chiavi, vedi [Eliminazione delle AWS CloudHSM chiavi KMS](#) da un archivio chiavi. AWS CloudHSM

Chiavi KMS negli archivi di chiavi esterni

Per creare, visualizzare, gestire, utilizzare e pianificare l'eliminazione delle chiavi KMS in un archivio delle chiavi esterne, puoi utilizzare procedure molto simili a quelle impiegate per altre chiavi KMS. Tuttavia, durante la creazione di una chiave KMS in un archivio delle chiavi esterne, specifica un [archivio delle chiavi esterne](#) e una [chiave esterna](#). Quando utilizzi una chiave KMS in un archivio delle chiavi esterne, le [operazioni di crittografia e decrittografia](#) vengono eseguite dal gestore delle chiavi esterne utilizzando la chiave esterna specificata.

AWS KMS non può creare, visualizzare, aggiornare o eliminare alcuna chiave crittografica nel gestore di chiavi esterno. AWS KMS non accede mai direttamente al gestore di chiavi esterno o a qualsiasi chiave esterna. Tutte le richieste relative alle operazioni di crittografia sono mediate dal [proxy dell'archivio delle chiavi esterne](#). Per utilizzare una chiave KMS in un archivio delle chiavi esterne, l'archivio che ospita la chiave KMS deve essere [connesso](#) al relativo proxy dell'archivio delle chiavi esterne.

Funzionalità supportate

Oltre alle procedure discusse in questa sezione, puoi eseguire le seguenti operazioni con le chiavi KMS in un archivio delle chiavi esterne:

- Utilizzare le [policy delle chiavi](#), le [policy IAM](#) e le [concessioni](#) per autorizzare l'accesso alle chiavi KMS.
- [Abilitare e disabilitare](#) le chiavi KMS. Queste azioni non influiscono sulla chiave esterna nel gestore delle chiavi esterne.
- Assegnare [tag](#), creare [alias](#) e utilizzare il [controllo degli accessi basato su attributi](#) (ABAC) per autorizzare l'accesso alle chiavi KMS.
- Utilizza le chiavi KMS per eseguire le seguenti operazioni crittografiche:
 - [Encrypt](#)
 - [Decrypt](#)
 - [GenerateDataKey](#)
 - [GenerateDataKeyWithoutPlaintext](#)
 - [ReEncrypt](#)

Le operazioni che generano coppie di chiavi di dati asimmetriche [GenerateDataKeyPair](#) [GenerateDataKeyPairWithoutPlaintext](#), non sono supportate negli archivi di chiavi personalizzati.

- Utilizzare le chiavi KMS con [Servizi AWS che si integrano con AWS KMS](#) e supportano le [chiavi gestite dal cliente](#).

Caratteristiche non supportate

- Gli archivi delle chiavi esterne supportano solo [chiavi KMS di crittografia simmetrica](#). Non puoi creare chiavi KMS HMAC o chiavi KMS asimmetriche in un archivio delle chiavi esterne.
- [GenerateDataKeyPair](#) e non [GenerateDataKeyPairWithoutPlaintext](#) sono supportati sulle chiavi KMS in un archivio di chiavi esterno.
- Non è possibile utilizzare un [AWS::KMS::Key CloudFormation modello](#) per creare un archivio di chiavi esterno o una chiave KMS in un archivio di chiavi esterno.
- Le [chiavi multi-regione](#) non sono supportate in un archivio delle chiavi esterne.
- Le chiavi KMS con [materiale della chiave importato](#) non sono supportate in un archivio delle chiavi esterne.
- La [rotazione automatica delle chiavi](#) non è supportata per le chiavi KMS in un archivio delle chiavi esterne.

Utilizzo delle chiavi KMS in un archivio di chiavi esterno

Quando utilizzi la chiave KMS in una richiesta, identifica la chiave KMS in base a [ID chiave, ARN chiave, alias o ARN alias](#). Non è necessario specificare l'archivio delle chiavi esterne. La risposta include gli stessi campi che vengono restituiti per qualsiasi chiave KMS di crittografia simmetrica. Tuttavia, quando utilizzi una chiave KMS in un archivio delle chiavi esterne, le operazioni di crittografia e decrittografia vengono eseguite dal gestore delle chiavi esterne utilizzando la chiave esterna associata alla chiave KMS.

[Per garantire che il testo cifrato crittografato da una chiave KMS in un archivio di chiavi esterno sia sicuro almeno quanto qualsiasi testo cifrato crittografato da una chiave KMS standard, utilizza la doppia crittografia. AWS KMS](#) I dati vengono prima crittografati utilizzando materiale chiave. AWS KMS AWS KMS Quindi, viene crittografato dal gestore delle chiavi esterne utilizzando la chiave esterna per la chiave KMS. Per decrittografare il testo criptato con doppia crittografia, il testo viene innanzitutto decodificato dal gestore delle chiavi esterno utilizzando la chiave esterna per la chiave KMS. Quindi vengono decrittografati AWS KMS utilizzando il materiale AWS KMS chiave per la chiave KMS.

Perché ciò avvenga, devono essere soddisfatte le seguenti condizioni.

- Lo [stato di chiave](#) della chiave KMS deve essere Enabled. Per trovare lo stato della chiave, consulta il campo Stato per le chiavi gestite dal cliente, la [AWS KMS console](#) o il KeyState campo nella risposta. [DescribeKey](#)
- L'archivio delle chiavi esterne che ospita la chiave KMS deve essere connesso al relativo [proxy dell'archivio delle chiavi esterne](#), ovvero [lo stato di connessione](#) dell'archivio delle chiavi esterne deve essere CONNECTED.

È possibile visualizzare lo stato della connessione nella pagina Archivi di chiavi esterni della AWS KMS console o nella [DescribeCustomKeyStores](#) risposta. Lo stato di connessione dell'archivio delle chiavi esterne viene visualizzato anche nella pagina dei dettagli della chiave KMS nella console AWS KMS. Nella pagina dei dettagli, scegli la scheda Cryptographic configuration (Configurazione crittografica) e visualizza il campo Connection state (Stato connessione) nella sezione Custom key store (Archivio delle chiavi personalizzate).

Se lo stato della connessione è DISCONNECTED, devi prima connetterlo. Se lo stato della connessione è FAILED, devi risolvere il problema, disconnettere l'archivio delle chiavi esterne e riconnetterlo. Per istruzioni, consultare [Connect e disconnetti gli archivi di chiavi esterni](#).

- Il proxy dell'archivio delle chiavi personalizzate deve essere in grado di trovare la chiave esterna.
- La chiave esterna deve essere abilitata e deve eseguire le operazioni di crittografia e decrittografia.

Lo stato della chiave esterna è indipendente e non influisce sulle modifiche apportate allo [stato chiave](#) della chiave KMS, inclusa l'attivazione e la disabilitazione della chiave stessa. Allo stesso modo, la disabilitazione o l'eliminazione della chiave esterna non modifica lo stato chiave della chiave KMS, ma le operazioni di crittografia che utilizzano la chiave KMS associata avranno esito negativo.

Se queste condizioni non vengono soddisfatte, l'operazione di crittografia ha esito negativo e AWS KMS restituisce un `KMSInvalidStateException` eccezione. Potrebbe essere necessario [ricollegare l'archivio delle chiavi esterne](#) o utilizzare gli strumenti di gestione delle chiavi esterne per riconfigurare o riparare la chiave esterna. Per ulteriori informazioni, consulta [the section called "Risoluzione dei problemi relativi all'archivio delle chiavi esterne"](#).

Quando utilizzi le chiavi KMS in un archivio delle chiavi esterne, tieni presente che le chiavi KMS in ogni archivio delle chiavi esterne condividono una [quota di richiesta dell'archivio delle chiavi personalizzate](#) per le operazioni di crittografia. Se superi la quota, AWS KMS restituisce

`unThrottlingException`. Per informazioni dettagliate sulle quote di richiesta dell'archivio delle chiavi personalizzate, consulta [Quote di richiesta per l'archivio delle chiavi personalizzate](#).

Ulteriori informazioni

- Per ulteriori informazioni sugli archivi di chiavi esterni, consulta [Archivi delle chiavi esterne](#).
- Per ulteriori informazioni sul materiale chiave negli archivi di chiavi esterni, consulta [Chiave esterna](#).
- Per creare chiavi KMS in un archivio di chiavi esterno, vedi [Crea una chiave KMS in archivi di chiavi esterni](#).
- Per identificare e visualizzare le chiavi KMS in un archivio di chiavi esterno, vedi [Identifica le chiavi KMS negli archivi di chiavi esterni](#).
- Per informazioni su considerazioni speciali sull'eliminazione delle chiavi KMS in un archivio di chiavi esterno, vedi [Eliminazione delle chiavi KMS da un archivio di chiavi esterno](#).

AWS KMS elementi essenziali della crittografia

AWS KMS utilizza algoritmi crittografici configurabili in modo che il sistema possa migrare rapidamente da un algoritmo o una modalità approvati a un altro. Il set predefinito iniziale di algoritmi di crittografia è stato selezionato dagli algoritmi Federal Information Processing Standard (FIPS) approvati per le loro proprietà e prestazioni di sicurezza.

Per ulteriori informazioni sugli algoritmi crittografici preferiti e accettabili, vedere [Algoritmi crittografici supportati](#)

Entropia e generazione di numeri casuali

AWS KMS la generazione di chiavi viene eseguita in. AWS KMS HSMs implementano un generatore ibrido di numeri casuali che utilizza il [NIST SP800-90A Deterministic Random Bit Generator \(DRBG\) CTR_DRBG using AES-256](#). Inizia con un generatore di bit casuale non deterministico con 384 bit di entropia ed è aggiornato con entropia aggiuntiva per fornire resistenza di previsione su ogni chiamata per il materiale crittografico.

Operazioni con chiavi simmetriche (solo crittografia)

Tutti i comandi di crittografia a chiave simmetrica utilizzati all'interno HSMs utilizzano gli [Advanced Encryption Standards \(AES\)](#), in [Galois Counter Mode \(GCM\)](#) utilizzando chiavi a 256 bit. Le chiamate analoghe per decrittografare utilizzano la funzione inversa.

AES-GCM è uno schema di crittografia autenticato. Oltre a crittografare testo in chiaro per produrre testo cifrato, calcola un tag di autenticazione sul testo cifrato e tutti i dati aggiuntivi per i quali è richiesta l'autenticazione (dati autenticati in aggiunta, o AAD). Il tag di autenticazione consente di garantire che i dati provengano dall'origine presunta e che il testo cifrato e l'AAD non siano stati modificati.

Spesso AWS omette l'inclusione dell'AAD nelle nostre descrizioni, specialmente quando si fa riferimento alla crittografia delle chiavi di dati. In questi casi, il testo circostante implica che la struttura da crittografare sia partizionata tra il testo in chiaro da crittografare e l'AAD in chiaro da proteggere.

AWS KMS offre la possibilità di importare materiale chiave in un file AWS KMS key anziché fare affidamento su di esso AWS KMS per generare il materiale chiave. Questo materiale chiave importato può essere crittografato utilizzando [RSAES-OAEP](#) per proteggere la chiave durante il trasporto verso l'HSM. AWS KMS Le coppie di chiavi RSA vengono generate su. AWS KMS HSMs Il materiale chiave importato viene decrittografato su un AWS KMS HSM e ricrittografato in AES-GCM prima di essere archiviato dal servizio.

Operazioni con chiave asimmetrica (crittografia, firma digitale e verifica della firma)

AWS KMS supporta l'uso di operazioni con chiavi asimmetriche per le operazioni di crittografia, firma digitale e accordo chiave. Le operazioni con chiave asimmetrica si basano su una chiave pubblica e una coppia di chiavi private correlate matematicamente che è possibile utilizzare per la crittografia e la decrittografia, la firma e la verifica della firma o la derivazione di segreti condivisi. La chiave privata non esce mai non crittografata. AWS KMS È possibile utilizzare la chiave pubblica interna AWS KMS richiamando le operazioni dell' AWS KMS API oppure scaricare la chiave pubblica e utilizzarla all'esterno di AWS KMS.

AWS KMS supporta i seguenti codici asimmetrici.

- RSA-OAEP (per la crittografia) e RSA-PSS e RSA-PKCS-#1-v1_5 (per la firma e la verifica): supporta le lunghezze delle chiavi RSA (in bit): 2048, 3072 e 4096 per diversi requisiti di sicurezza.
- Curva ellittica (ECC): utilizzata per la firma e la verifica o la derivazione di segreti condivisi, ma non per entrambi. Supporta le curve ECC: NIST P256, P384, P521, SECP 256k1, Ed25519.
- ML-DSA: utilizzato per la firma e la verifica. Le specifiche principali supportate da ML-DSA sono: ML_DSA_44, ML_DSA_65 e ML_DSA_87.

- SM2 (Solo regioni della Cina): utilizzato per la crittografia e la decrittografia, la firma e la verifica o la derivazione di segreti condivisi, ma è necessario scegliere un utilizzo della chiave. Supporta SM2 PKE per la crittografia e DSA per la firma SM2.

Funzioni di derivazione chiave

Una funzione di derivazione delle chiavi viene utilizzata per derivare chiavi aggiuntive da un segreto o una chiave iniziale. AWS KMS utilizza una funzione di derivazione delle chiavi (KDF) per derivare chiavi per chiamata per ogni crittografia in base a. AWS KMS key [Tutte le operazioni KDF utilizzano il KDF in modalità contatore utilizzando HMAC \[\] con \[0\]. FIPS197 SHA256 FIPS18](#) La chiave derivata a 256 bit viene utilizzata con AES-GCM per crittografare o decrittare i dati e le chiavi dei clienti.

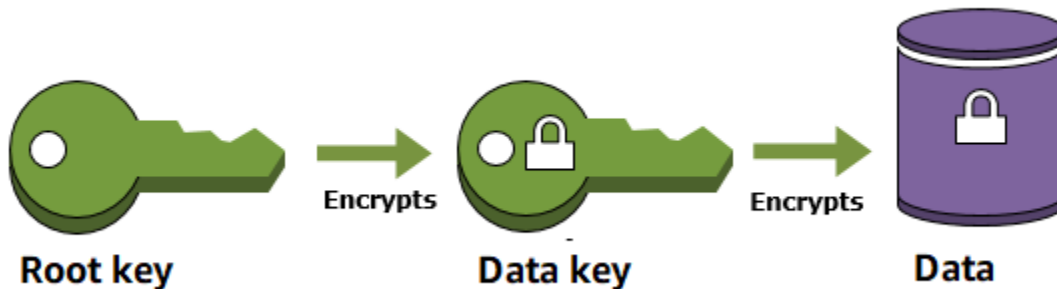
AWS KMS uso interno delle firme digitali

Le firme digitali vengono utilizzate anche per autenticare comandi e comunicazioni tra entità AWS KMS . Tutte le entità del servizio dispongono di una coppia di chiavi ECDSA (Elliptic Curve Digital Signature Algorithm). Eseguono ECDSA come definito in [Utilizzo degli algoritmi ECC \(Elliptic Curve Cryptography\) nella sintassi del messaggio di crittografia \(CMS\)](#) e X9.62-2005: Crittografia a chiave pubblica per il settore dei servizi finanziari: ECDSA (Elliptic Curve Digital Signature Algorithm). Le entità utilizzano l'algoritmo hash sicuro definito nelle [Federal Information Processing Standards Publications, FIPS PUB 180-4](#), noto come. SHA384 Le chiavi vengono generate sulla curva secp384r1 (NIST-P384).

Crittografia envelope

Quando esegui la crittografia dei dati, i dati sono protetti, ma è necessario proteggere la chiave crittografica. Una strategia consiste nel crittografarla. La Crittografia envelope consiste nel crittografare i dati di testo normale con una chiave di dati, quindi crittografare la chiave di dati in un'altra chiave.

È anche possibile crittografare la chiave crittografica dei dati in un'altra chiave crittografica e crittografare tale chiave crittografica con un'altra chiave crittografica. Alla fine, però, una chiave deve rimanere in testo normale in modo da poter decrittografare le chiavi e i dati. Questa chiave crittografica di primo livello della chiave in testo normale è nota come chiave radice.



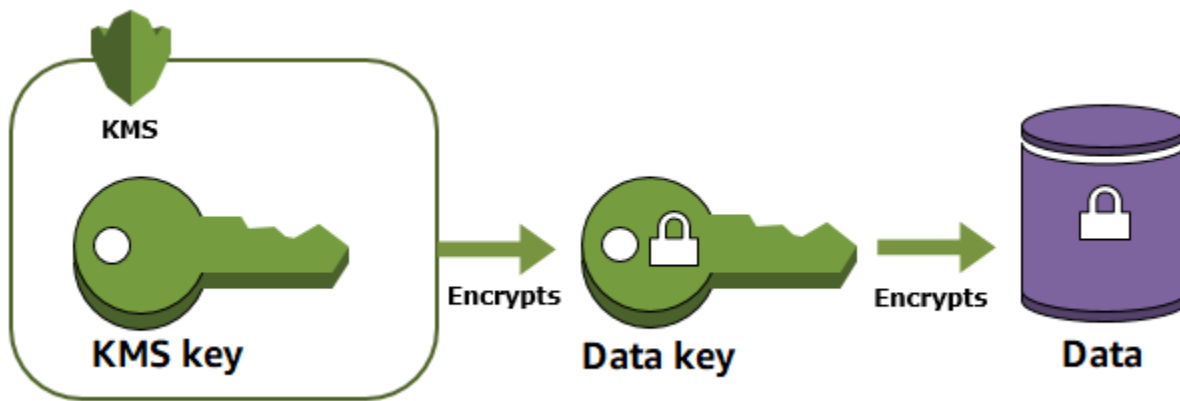
AWS KMS ti aiuta a proteggere le tue chiavi di crittografia archiviandole e gestendole in modo sicuro. La chiave principale memorizzata in AWS KMS, nota come AWS KMS keys, non lascia mai i moduli di sicurezza [hardware convalidati AWS KMS FIPS 140-3 Security Level 3 non crittografati](#). Per utilizzare una chiave KMS, è necessario chiamare AWS KMS.

Una costruzione di base utilizzata all'interno di molti sistemi di crittografia è la crittografia envelope. La crittografia envelope utilizza due o più chiavi di crittografia per proteggere un messaggio. In genere, una chiave è derivata da una chiave statica a lungo termine k e un'altra chiave è una chiave per messaggio, $msgKey$, che viene generata per crittografare il messaggio. L'envelope è formata crittografando il messaggio: $ciphertext = \text{Encrypt}(msgKey, message)$. Quindi la chiave del messaggio viene crittografata con la chiave statica a lungo termine: $encKey = \text{Encrypt}(k, msgKey)$. Infine, i due valori ($encKey$, $ciphertext$) sono assemblati in un'unica struttura, o messaggio crittografato con envelope.

Il destinatario, con accesso a k , può aprire il messaggio con envelope decrittando prima la chiave crittografata e quindi il messaggio.

AWS KMS offre la possibilità di gestire queste chiavi statiche a lungo termine e di automatizzare il processo di crittografia in busta dei dati.

Oltre alle funzionalità di crittografia fornite all'interno del AWS KMS servizio, Encryption [SDK fornisce librerie di AWS crittografia delle buste lato client](#). È possibile utilizzare queste librerie per proteggere i dati e le chiavi di crittografia utilizzate per crittografare i dati.



La crittografia envelope offre diversi vantaggi:

- Protezione delle chiavi dei dati

Quando crittografi una chiave di dati, non è necessario preoccuparsi di dove archivarla, poiché la chiave di dati è intrinsecamente protetta dalla crittografia. Puoi archiviare la chiave di dati crittografata in modo sicuro con i dati crittografati.

- Crittografia degli stessi dati con più chiavi

Le operazioni di crittografia possono essere dispendiose in termini di tempo, soprattutto quando i dati crittografati sono oggetti di grandi dimensioni. Invece di ricrittografare dati grezzi più volte con chiavi diverse, è possibile ricrittografare solo le chiavi di dati che proteggono i dati grezzi.

- Abbinare i punti di forza di più algoritmi

In generale, gli algoritmi di chiavi simmetriche sono più veloci e producono testi cifrati più piccoli rispetto agli algoritmi di chiave pubblica. Tuttavia, gli algoritmi di chiave pubblica forniscono una separazione intrinseca dei ruoli e facilitano la gestione delle chiavi. La crittografia envelope ti consente di abbinare i punti di forza di ciascuna strategia.

Operazioni di crittografia

In AWS KMS, le operazioni crittografiche sono operazioni API che utilizzano chiavi KMS per proteggere i dati. Poiché le chiavi KMS rimangono all'interno AWS KMS, è necessario chiamare AWS KMS per utilizzare una chiave KMS in un'operazione crittografica.

Per eseguire operazioni crittografiche con le chiavi KMS, usa AWS SDKs, AWS Command Line Interface (AWS CLI) o AWS Strumenti per PowerShell. Non è possibile eseguire operazioni di

crittografia nella console AWS KMS . Per esempi di chiamata delle operazioni crittografiche in diversi linguaggi di programmazione, consulta [Esempi di codice per AWS KMS l'utilizzo AWS SDKs](#).

La tabella seguente elenca le operazioni AWS KMS crittografiche. Viene illustrato anche il tipo di chiave e i requisiti di [utilizzo della chiave](#) per le chiavi KMS usate nell'operazione.

Operazioni	Tipo di chiavi	Utilizzo delle chiavi
Decrypt	Simmetrico o asimmetrico	ENCRYPT_DECRYPT
DeriveSharedSecret	Asimmetrica	KEY_AGREEMENT
Encrypt	Simmetrico o asimmetrico	ENCRYPT_DECRYPT
GenerateDataKey	Simmetria	ENCRYPT_DECRYPT
GenerateDataKeyPair	Simmetrica [1] Non è supportato sulle chiavi KMS negli archivi delle chiavi personalizzate.	ENCRYPT_DECRYPT
GenerateDataKeyPairWithoutPlaintext	Simmetrica [1] Non è supportato sulle chiavi KMS negli archivi delle chiavi personalizzate.	ENCRYPT_DECRYPT
GenerateDataKeyWithoutPlaintext	Simmetria	ENCRYPT_DECRYPT
GenerateMac	HMAC	GENERATE_VERIFY_MAC
GenerateRandom	N/D. Questa operazione non	N/D

Operazioni	Tipo di chiavi	Utilizzo delle chiavi
	utilizza una chiave KMS.	
ReEncrypt	Simmetrico o asimmetrico	ENCRYPT_DECRYPT
Sign	Asimmetrica	SIGN_VERIFY
Verify	Asimmetrica	SIGN_VERIFY
VerifyMac	HMAC	GENERATE_VERIFY_MAC

[1] Genera coppie di chiavi di dati asimmetriche protette da una chiave KMS di crittografia simmetrica.

Per informazioni sulle autorizzazioni per le operazioni di crittografia, consulta [the section called “Riferimento per le autorizzazioni”](#).

Per renderlo AWS KMS reattivo e altamente funzionale per tutti gli utenti, AWS KMS stabilisce quote sul numero di operazioni crittografiche richiamate in ogni secondo. Per informazioni dettagliate, vedi [the section called “Quote condivise per le operazioni di crittografia”](#).

Algoritmi crittografici supportati

algoritmi crittografici

Le tabelle seguenti riassumono gli algoritmi crittografici, i codici, le modalità e le dimensioni delle chiavi utilizzati nei AWS suoi servizi per proteggere i dati. Non devono essere considerate un elenco esaustivo di tutte le opzioni di crittografia disponibili in. AWS Gli algoritmi si dividono in due categorie:

- Gli algoritmi preferiti soddisfano gli standard AWS di sicurezza e prestazioni.
- Gli algoritmi accettabili possono essere utilizzati per motivi di compatibilità in alcune applicazioni, ma non sono preferiti.

Crittografia asimmetrica

La tabella seguente elenca gli algoritmi asimmetrici supportati per la crittografia, l'accordo di chiave e le firme digitali.

Tipo	Algoritmo	Stato
Crittografia	RSA-OAEP (modulo a 2048 o 3072 bit)	Accettabile
Crittografia	HPKE (P-256 o P-384, HKDF e AES-GCM)	Accettabile
Accordo chiave	ML-KEM-768 o ML-KEM-1024	Preferito (resistente quantisticamente)
Accordo chiave	ECDH (E) con P-384	Accettabile
Accordo chiave	ECDH (E) con P-256, P-521 o X25519	Accettabile
Accordo chiave	ECDH (E) con BrainPool P256R1, BrainPoolP384R1 o BrainPoolP512R1	Accettabile
Signatures (Firme)	ML-DSA-65 o ML-DSA-87	Preferito (resistente quantisticamente)
Signatures (Firme)	SLH-DSA	Preferito (firma a resistenza quantistica) software/firmware
Signatures (Firme)	ECDSA con P-384	Accettabile
Signatures (Firme)	ECDSA con P-256, P-521 o Ed25519	Accettabile
Signatures (Firme)	RSA-2048 o RSA-3072	Accettabile

Crittografia simmetrica

La tabella seguente elenca gli algoritmi simmetrici supportati per la crittografia, la crittografia autenticata e il key wrapping.

Tipo	Algoritmo	Stato
Crittografia autenticata	AES-GCM-256	Preferito
Crittografia autenticata	AES-GCM-128	Accettabile
Crittografia autenticata	ChaCha20/Poly1305	Accettabile
Modalità di crittografia	AES-XTS-256 (per archiviazione a blocchi)	Preferito
Modalità di crittografia	AES-CBC/CTR (modalità non autenticate)	Accettabile
Confezionamento delle chiavi	AES-GCM-256	Preferito
Confezionamento delle chiavi	AES-KW o AES-KWP con chiavi a 256 bit	Accettabile

Funzioni di crittografia

La tabella seguente elenca gli algoritmi supportati per l'hashing, la derivazione delle chiavi, l'autenticazione dei messaggi e l'hashing delle password.

Tipo	Algoritmo	Stato
Hashing	SHA2-384	Preferito
Hashing	SHA2-256	Accettabile
Hashing	SHA3	Accettabile
Derivazione delle chiavi	HKDF_Expand o HKDF con -256 SHA2	Preferito

Tipo	Algoritmo	Stato
Derivazione chiave	KDF in modalità contatore con HMAC- 256 SHA2	Accettabile
Codice di autenticazione dei messaggi	HMAC- 384 SHA2	Preferito
Codice di autenticazione del messaggio	HMAC- 256 SHA2	Accettabile
Codice di autenticazione del messaggio	KMAC	Accettabile
Hashing delle password	scrypt con SHA384	Preferito
Hashing delle password	PBKDF2	Accettabile

[Per ulteriori dettagli sugli algoritmi crittografici implementati in AWS, vedere Algoritmi di crittografia e Servizi AWS](#)

Accesso e autorizzazioni alle chiavi KMS

Per utilizzarlo AWS KMS, devi disporre di credenziali che AWS possano essere utilizzate per autenticare le tue richieste. [Le credenziali devono includere i permessi di accesso alle AWS risorse e gli alias. AWS KMS keys](#) Nessun AWS principale dispone di alcuna autorizzazione per una chiave KMS a meno che tale autorizzazione non sia fornita esplicitamente e mai negata. Non sono disponibili autorizzazioni implicite o automatiche per l'utilizzo o la gestione di una chiave KMS.

Per controllare l'accesso alle chiavi KMS, è possibile utilizzare i seguenti meccanismi delle policy.

- [Policy delle chiavi](#): ogni chiave KMS ha una policy delle chiavi. Le policy delle chiavi sono il meccanismo principale per controllare l'accesso a una chiave KMS. Puoi utilizzare la sola policy delle chiavi per il controllo dell'accesso, per cui l'accesso alla chiave KMS nella sua interezza è definito in un unico documento (la policy delle chiavi). Per ulteriori informazioni sull'utilizzo delle policy delle chiavi, consulta [Policy delle chiavi](#).
- [Policy IAM](#): è possibile utilizzare le policy IAM insieme alla policy delle chiavi e alle concessioni per controllare l'accesso a una chiave KMS. Il controllo dell'accesso eseguito in questo modo consente di gestire tutte le autorizzazioni delle identità IAM in IAM. Per utilizzare una policy IAM per consentire l'accesso a una chiave KMS, la policy delle chiavi deve consentirla esplicitamente. Per ulteriori informazioni sull'utilizzo di policy IAM consulta [Policy IAM](#).
- [Concessioni](#): è possibile utilizzare le concessioni insieme alla policy delle chiavi e alle policy IAM per consentire l'accesso a una chiave KMS. Il controllo dell'accesso eseguito in questo modo ti consente di accedere alla chiave KMS nella policy delle chiavi e di permettere alle identità di delegare l'accesso ad altri utenti. Per ulteriori informazioni sull'utilizzo di concessioni, consulta [Sovvenzioni in AWS KMS](#).

Politiche chiave KMS

Il modo principale per gestire l'accesso alle AWS KMS risorse è attraverso le politiche. Le policy sono documenti che descrivono quali principali possono accedere a quali risorse. Le politiche associate a un'identità IAM sono chiamate politiche basate sull'identità (o politiche IAM) e le politiche associate ad altri tipi di risorse sono chiamate politiche delle risorse. AWS KMS le politiche relative alle risorse per le chiavi KMS sono chiamate politiche chiave.

Tutte le chiavi KMS dispongono di una policy delle chiavi. Se non ne fornisci una, ne AWS KMS crea una per te. La [politica di chiave predefinita](#) AWS KMS utilizzata varia a seconda che si crei la chiave

nella AWS KMS console o si utilizzi l' AWS KMS API. [Ti consigliamo di modificare la politica delle chiavi predefinita per allinearla ai requisiti della tua organizzazione per le autorizzazioni con privilegi minimi.](#)

Puoi utilizzare la policy chiave da sola per controllare l'accesso se la chiave e il responsabile IAM si trovano nello stesso AWS account, il che significa che l'intero ambito di accesso alla chiave KMS è definito in un unico documento (la politica chiave). Tuttavia, quando un chiamante di un account deve accedere a una chiave in un altro account, non è possibile utilizzare la sola politica delle chiavi per concedere l'accesso. Nello scenario tra più account, è necessario allegare all'utente o al ruolo del chiamante una policy IAM che consenta esplicitamente al chiamante di effettuare la chiamata API.

Puoi anche utilizzare le policy IAM in combinazione con le policy e le concessioni chiave per controllare l'accesso a una chiave KMS. Per utilizzare una policy IAM per controllare l'accesso a una chiave KMS, la policy chiave deve autorizzare l'account a utilizzare le policy IAM. Puoi specificare una [dichiarazione di policy chiave che abiliti le politiche IAM](#) oppure puoi [specificare esplicitamente i principi consentiti](#) nella policy chiave.

Durante la stesura delle policy, assicurati di disporre di controlli rigorosi che limitino chi può eseguire le seguenti azioni:

- Aggiorna, crea ed elimina le politiche chiave IAM e KMS
- Allega e scollega le politiche IAM da utenti, ruoli e gruppi
- Allega e scollega le policy chiave KMS dalle tue chiavi KMS

Concessioni chiave KMS

[Oltre a IAM e alle politiche chiave, AWS KMS supporta le sovvenzioni.](#) Le sovvenzioni forniscono un modo flessibile e potente per delegare le autorizzazioni. Puoi utilizzare le sovvenzioni per concedere l'accesso con chiave KMS a tempo limitato ai responsabili IAM nel tuo AWS account o in altri account. AWS Ti consigliamo di emettere un accesso limitato nel tempo se non conosci i nomi dei responsabili al momento della creazione delle policy o se i principali che richiedono l'accesso cambiano frequentemente. Il [titolare del beneficiario](#) può avere lo stesso account della chiave KMS o un altro account. Se il principale e la chiave KMS si trovano in account diversi, è necessario specificare una policy IAM oltre alla concessione. Le sovvenzioni richiedono una gestione aggiuntiva perché è necessario chiamare un'API per creare la sovvenzione e ritirarla o revocarla quando non è più necessaria.

Politiche chiave in AWS KMS

Una politica chiave è una politica delle risorse per un. AWS KMS key Le policy chiave sono lo strumento principale per controllare l'accesso alle chiavi KMS. Ogni chiave KMS deve avere esattamente una policy chiave. Le istruzioni nella policy delle chiavi determinano chi dispone dell'autorizzazione per utilizzare la chiave KMS e come questa può essere utilizzata. Puoi anche usare le [policy IAM](#) e le [concessioni](#) per controllare l'accesso alla chiave KMS, ma ogni chiave KMS deve avere un documento di policy delle chiavi.

Nessun AWS principale, incluso l'utente root dell'account o il creatore della chiave, dispone delle autorizzazioni relative a una chiave KMS a meno che non siano esplicitamente consentite e mai negate in una policy chiave, in una policy IAM o in una concessione.

A meno che la policy delle chiavi non lo consenta esplicitamente, non è possibile utilizzare le policy IAM per permettere l'accesso a una chiave KMS. Senza l'autorizzazione dalla policy delle chiavi, le policy IAM che consentono le autorizzazioni non hanno alcun effetto. È possibile utilizzare una policy IAM per negare un'autorizzazione a una chiave KMS senza l'autorizzazione da una policy delle chiavi. Per impostazione predefinita, la policy delle chiavi di default abilita le policy IAM. Per abilitare le policy IAM nella policy delle chiavi, aggiungere l'istruzione della policy descritta in [Consente l'accesso a Account AWS e abilita le policy IAM](#).

A differenza delle policy IAM, che sono globali, le policy chiave sono regionali. Una policy delle chiavi controlla l'accesso a una sola chiave KMS nella stessa regione. Non ha alcun effetto sulle chiavi KMS in altre regioni.

Argomenti

- [Creazione di una policy delle chiavi](#)
- [Policy delle chiavi predefinita](#)
- [Visualizza una politica chiave](#)
- [Modificare una policy della chiave](#)
- [Autorizzazioni per i AWS servizi nelle politiche chiave](#)

Creazione di una policy delle chiavi

Puoi creare e gestire le policy chiave nella AWS KMS console o utilizzando operazioni AWS KMS API, come [CreateKey](#), [ReplicateKey](#). [PutKeyPolicy](#)

Quando crei una chiave KMS nella AWS KMS console, la console ti guida attraverso i passaggi per creare una politica chiave basata sulla [politica chiave predefinita per la console](#). Quando utilizzi `CreateKey` o `ReplicateKey` APIs, se non specifichi una politica chiave, questa APIs applica la politica chiave [predefinita per le chiavi create a livello di codice](#). Quando usi l'API `PutKeyPolicy`, devi specificare una policy della chiave.

Ogni documento di policy può avere una o più istruzioni di policy. L'esempio seguente mostra un documento di policy della chiave valido con un'istruzione della policy.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DescribePolicyStatement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:user/Alice"
      },
      "Action": "kms:DescribeKey",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:KeySpec": "SYMMETRIC_DEFAULT"
        }
      }
    }
  ]
}
```

Argomenti

- [Formato della policy della chiave](#)
- [Elementi in una policy della chiave](#)
- [Esempi di policy delle chiavi](#)

Formato della policy della chiave

Un documento di policy della chiave deve rispettare le seguenti regole:

- Fino a 32 kilobyte (32.768 byte)
- L'elemento `Sid` in un'istruzione di policy della chiave non può includere spazi. Gli spazi sono vietati nell'elemento `Sid` di un documento di policy IAM.

Un documento di policy della chiave può includere solo i seguenti caratteri:

- Caratteri ASCII stampabili
- I caratteri stampabili nel set di caratteri Basic Latin e Latin-1 Supplement
- I caratteri speciali di tabulazione (`\u0009`), avanzamento di riga (`\u000A`) e ritorno a capo (`\u000D`)

Elementi in una policy della chiave

Un documento di policy delle chiavi deve avere i seguenti elementi:

Versione

Specifica la versione del documento della policy della chiave. Impostare la versione su `2012-10-17` (la versione più recente).

Dichiarazione

Include le istruzioni della policy. Un documento di policy delle chiavi deve avere almeno un'istruzione.

Ogni istruzione in una policy della chiave può contenere fino a sei elementi. Gli elementi `Effect`, `Principal`, `Action` e `Resource` sono obbligatori.

Sid

(Facoltativo) L'identificatore di istruzione (`Sid`) è una stringa arbitraria che è possibile utilizzare per descrivere l'istruzione. Il `Sid` in una policy della chiave può includere spazi. Non è possibile includere spazi nell'elemento `Sid` di una policy IAM.

Effetto

(Obbligatorio) Specifica se concedere o negare le autorizzazioni nell'istruzione di policy. I valori validi sono Allow e Deny. Se non concedi esplicitamente l'accesso a una chiave KMS, l'accesso viene implicitamente rifiutato. È anche possibile negare esplicitamente l'accesso a una chiave KMS. È possibile eseguire questa operazione per accertarsi che un utente non sia in grado di accedervi, anche quando l'accesso viene concesso da un'altra policy.

Principale

(Obbligatorio) Il [principale](#) è l'identità che ottiene le autorizzazioni specificate nell'istruzione di policy. Puoi specificare utenti IAM Account AWS, ruoli IAM e alcuni AWS servizi come principali in una policy chiave. I [gruppi di utenti](#) IAM non sono un principale valido in alcun tipo di policy.

Un valore asterisco, ad esempio "AWS": "*", rappresenta tutte le identità AWS in tutti gli account.

Important

Non impostare il principale su un asterisco (*) in un'istruzione della policy della chiave che consenta autorizzazioni, a meno che non utilizzi [condizioni](#) per limitare la policy della chiave. Un asterisco indica ogni identità in ogni Account AWS autorizzazione all'uso della chiave KMS, a meno che un'altra dichiarazione politica non lo neghi esplicitamente. Gli utenti di altri paesi Account AWS possono utilizzare la tua chiave KMS ogni volta che dispongono delle autorizzazioni corrispondenti nel proprio account.

Note

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

Quando il principale in una istruzione di policy delle chiavi è un [principale Account AWS](#) espresso come `arn:aws:iam::111122223333:root`, l'istruzione non fornisce l'autorizzazione ad alcun principale IAM. Al contrario, dà il Account AWS permesso di

utilizzare le politiche IAM per delegare le autorizzazioni specificate nella politica chiave. (Un principale in formato `arn:aws:iam::111122223333:root` non rappresenta l'[utente root dell'account AWS](#), nonostante l'uso di "root" nell'identificatore dell'account. Tuttavia, il principale dell'account rappresenta l'account e i relativi amministratori, incluso l'utente root dell'account.)

Quando il principale è un altro Account AWS o i suoi responsabili, le autorizzazioni sono effettive solo quando l'account è abilitato nella regione con la chiave KMS e la policy chiave. Per informazioni sulle regioni non abilitate per impostazione predefinita ("regioni attivate"), consulta [Gestione delle Regioni AWS](#) nella Riferimenti generali di AWS.

Per consentire a un altro Account AWS o ai suoi responsabili di utilizzare una chiave KMS, devi fornire l'autorizzazione in una politica chiave e in una politica IAM nell'altro account. Per informazioni dettagliate, vedi [Autorizzazione per gli utenti in altri account di utilizzare una chiave KMS](#).

Azione

(Obbligatorio) Specificano le operazioni API da permettere o negare. Ad esempio, `kms:Encrypt` corrisponde all'operazione AWS KMS [Encrypt](#). È possibile elencare più di un'operazione in un'istruzione di policy. Per ulteriori informazioni, consulta [Riferimento per le autorizzazioni](#).

Note

Se l'elemento `Action` obbligatorio non è presente nell'istruzione della policy delle chiavi, l'istruzione non avrà alcun effetto. Una dichiarazione politica chiave senza un `Action` elemento non si applica a nessuna chiave KMS.

Quando manca un `Action` elemento in una dichiarazione politica chiave, la AWS KMS console riporta correttamente un errore, ma l'operazione viene [CreateKey](#) eseguita correttamente [PutKeyPolicy](#) APIs, anche se la dichiarazione politica è inefficace.

Risorsa

(Obbligatorio) In una policy della chiave, il valore dell'elemento `Resource` (Risorsa) è `"*"`, che significa "questa chiave KMS". L'asterisco (`"*"`) identifica la chiave KMS a cui è collegata la policy delle chiavi.

 Note

Se l'elemento Resource obbligatorio non è presente nell'istruzione della policy delle chiavi, l'istruzione non avrà alcun effetto. Una istruzione di policy delle chiavi senza un elemento Resource non si applica ad alcuna chiave KMS.

Quando manca un Resource elemento in una dichiarazione politica chiave, la AWS KMS console segnala correttamente un errore, ma l'[CreateKey](#) errore viene restituito [PutKeyPolicy](#) APIs correttamente, anche se la dichiarazione politica è inefficace.

Condizione

(Facoltativo) Le condizioni specificano i requisiti da soddisfare affinché una policy della chiave diventi effettiva. Con le condizioni, AWS può valutare il contesto di una richiesta API per determinare se l'informativa sulla politica è applicabile o meno.

Per specificare le condizioni, si utilizzano chiavi di condizione predefinite. AWS KMS supporta chiavi di [condizione AWS globali e chiavi](#) di [AWS KMS condizione](#). Per supportare il controllo degli accessi basato sugli attributi (ABAC), AWS KMS fornisce chiavi di condizione che controllano l'accesso a una chiave KMS basata su tag e alias. Per informazioni dettagliate, vedi [ABAC per AWS KMS](#).

Il formato per una condizione è:

```
"Condition": {"condition operator": {"condition key": "condition value"}}
```

come per esempio:

```
"Condition": {"StringEquals": {"kms:CallerAccount": "111122223333"}}
```

Per ulteriori informazioni sulla sintassi delle policy, consulta [AWS IAM AWS Policy Reference nella IAM User Guide](#).

Esempi di policy delle chiavi

L'esempio seguente mostra una policy della chiave completa per una chiave KMS di crittografia simmetrica. È utile come riferimento mentre leggi i concetti della policy delle chiavi in questo capitolo.

Questa policy delle chiavi combina le istruzioni di policy di esempio della sezione precedente sulla [policy delle chiavi predefinita](#) in un'unica policy delle chiavi che ottiene quanto segue:

- Consente all'esempio Account AWS, 111122223333, l'accesso completo alla chiave KMS. Concede all'account e ai relativi amministratori, incluso l'utente root dell'account (per le emergenze), l'uso delle policy IAM nell'account per consentire l'accesso alla chiave KMS.
- Consente al ruolo IAM ExampleAdminRole di amministrare la chiave KMS.
- Permette al ruolo IAM ExampleUserRole di utilizzare la chiave KMS.

JSON

```
{
  "Id": "key-consolepolicy",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnableIAMUserPermissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "AllowKeyAdministratorsAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleAdminRole"
      },
      "Action": [
        "kms:Create*",
        "kms:Describe*",
        "kms:Enable*",
        "kms:List*",
        "kms:Put*",
        "kms:Update*",
        "kms:Revoke*",
        "kms:Disable*",
        "kms:Get*",
```

```

        "kms:Delete*",
        "kms:TagResource",
        "kms:UntagResource",
        "kms:ScheduleKeyDeletion",
        "kms:CancelKeyDeletion",
        "kms:RotateKeyOnDemand"
    ],
    "Resource": "*"
},
{
    "Sid": "AllowKeyUse",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleUserRole"
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
},
{
    "Sid": "AllowAttachmentPersistentResources",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleUserRole"
    },
    "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
    ],
    "Resource": "*",
    "Condition": {
        "Bool": {
            "kms:GrantIsForAWSResource": "true"
        }
    }
}
]

```

```
}
```

Policy delle chiavi predefinita

Quando si crea una chiave KMS, è possibile specificare la policy delle chiavi per la nuova chiave KMS. Se non ne fornisci una, ne AWS KMS crea una per te. La politica di chiave predefinita AWS KMS utilizzata varia a seconda che si crei la chiave nella AWS KMS console o si utilizzi l' AWS KMS API.

La policy delle chiavi predefinita al momento della creazione di una chiave KMS a livello di programmazione

Quando crei una chiave KMS a livello di codice con l'[AWS KMS API](#) (anche utilizzando [AWS Command Line Interface](#) o [AWS Strumenti per PowerShell](#)) e non specifichi una politica chiave, AWS KMS applica una policy chiave predefinita molto semplice. [AWS SDKs](#) Questa politica chiave predefinita ha una dichiarazione politica che fornisce al Account AWS proprietario della chiave KMS l'autorizzazione a utilizzare le politiche IAM per consentire l'accesso a tutte le AWS KMS operazioni sulla chiave KMS. Per ulteriori informazioni su questa istruzione di policy, consulta [Consente l'accesso a Account AWS e abilita le policy IAM](#).

Politica delle chiavi predefinita quando si crea una chiave KMS con Console di gestione AWS

Quando [crei una chiave KMS con Console di gestione AWS](#), la policy chiave inizia con l'informativa che [consente l'accesso Account AWS e abilita le politiche IAM](#). [La console aggiunge quindi un'istruzione Key Administrators, un'istruzione key users e \(per la maggior parte dei tipi di chiave\) un'istruzione che consente ai responsabili di utilizzare la chiave KMS con altri servizi.](#) [AWS](#) Puoi utilizzare le funzionalità della AWS KMS console per specificare gli utenti IAM, i ruoli IAM e Account AWS chi sono gli amministratori chiave e gli utenti chiave (o entrambi).

Autorizzazioni

- [Consente l'accesso a Account AWS e abilita le policy IAM](#)
- [Consente agli amministratori delle chiavi di amministrare la chiave KMS](#)
- [Consente agli utenti della chiave di utilizzare la chiave KMS](#)
 - [Consente agli utenti della chiave di utilizzare una chiave KMS per le operazioni di crittografia](#)
 - [Consente agli utenti della chiave di utilizzare la chiave KMS con i servizi AWS](#)

Consente l'accesso a Account AWS e abilita le policy IAM

La seguente istruzione delle policy delle chiavi predefinita è fondamentale.

- Fornisce al Account AWS proprietario della chiave KMS l'accesso completo alla chiave KMS.

A differenza di altre politiche relative alle AWS risorse, una politica AWS KMS chiave non concede automaticamente l'autorizzazione all'account o a nessuna delle sue identità. Per concedere l'autorizzazione agli amministratori di account, la policy delle chiavi deve includere un'istruzione esplicita che fornisce l'autorizzazione, come questa.

- Consente all'account di utilizzare le policy IAM per consentire l'accesso alla chiave KMS, oltre alla policy delle chiavi.

Senza questa autorizzazione, le policy IAM che consentono l'accesso alla chiave sono inefficaci, anche se le policy IAM che negano l'accesso alla chiave sono ancora valide.

- Riduce il rischio che la chiave diventi ingestibile fornendo l'autorizzazione per il controllo degli accessi agli amministratori dell'account, incluso l'utente root dell'account, che non può essere eliminato.

La seguente istruzione della policy delle chiavi è l'unica policy delle chiavi predefinita per le chiavi KMS create a livello di programmazione. È la prima dichiarazione politica nella politica chiave predefinita per le chiavi KMS create nella AWS KMS console.

```
{
  "Sid": "Enable IAM User Permissions",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:root"
  },
  "Action": "kms:*",
  "Resource": "*"
}
```

Consente alle policy IAM di controllare l'accesso alla chiave KMS.

L'informativa chiave mostrata sopra fornisce al proprietario Account AWS l'autorizzazione chiave per utilizzare le politiche IAM, nonché le politiche chiave, per consentire tutte le azioni (kms : *) sulla chiave KMS.

Il principale in questa istruzione della policy delle chiavi è il [principale dell'account](#), rappresentato da un ARN nel formato `arn:aws:iam::account-id:root`. L'account principal rappresenta l'AWS account e i suoi amministratori.

Quando il principale in una istruzione di policy delle chiavi è il principale dell'account, l'istruzione della policy non fornisce al principale IAM l'autorizzazione a utilizzare la chiave KMS.

Consente invece all'account di utilizzare le policy IAM per delegare le autorizzazioni specificate nell'istruzione della policy. Questa istruzione di policy delle chiavi predefinita consente all'account di utilizzare le policy IAM per delegare l'autorizzazione per tutte le operazioni (`kms : *`) sulla chiave KMS.

Riduce il rischio che la chiave KMS diventi ingestibile.

A differenza di altre politiche relative alle AWS risorse, una politica AWS KMS chiave non concede automaticamente l'autorizzazione all'account o ai suoi responsabili. Per fornire l'autorizzazione a qualsiasi principale, incluso il [principale dell'account](#), è necessario utilizzare una istruzione della policy delle chiavi che fornisca esplicitamente l'autorizzazione. Non è richiesto di concedere al principale dell'account, o a qualsiasi principale, l'accesso alla chiave KMS. Tuttavia, l'accesso al principale dell'account aiuta a evitare che la chiave diventi ingestibile.

Ad esempio, si supponga di creare una policy delle chiavi che dia a un solo utente l'accesso alla chiave KMS. Se questo utente viene eliminato, la chiave diventa ingestibile e diventa necessario [contattare AWS Support](#) per ottenere di nuovo l'accesso alla chiave KMS.

La dichiarazione politica chiave mostrata sopra fornisce all'[account l'autorizzazione principale](#) a controllare la chiave. L'account principal rappresenta l'account Account AWS e i suoi amministratori, incluso l'[utente root dell'account](#). L'utente root dell'account è l'unico principale che non può essere eliminato a meno che non si elimini l'Account AWS. Le best practice IAM scoraggiano l'operazione per conto dell'utente root dell'account, tranne in caso di emergenza. Tuttavia, potrebbe essere necessario agire come utente root dell'account se si eliminano tutti gli altri utenti e ruoli con accesso alla chiave KMS.

Consente agli amministratori delle chiavi di amministrare la chiave KMS

La policy delle chiavi predefinita creata dalla console consente di scegliere gli utenti e i ruoli IAM nell'account e renderli amministratori delle chiavi. Questa istruzione si chiama istruzione degli amministratori delle chiavi. Gli amministratori delle chiavi dispongono delle autorizzazioni per gestire la chiave KMS, ma non dispongono delle autorizzazioni per utilizzare la chiave KMS nelle [operazioni](#)

[di crittografia](#). È possibile aggiungere utenti e ruoli IAM all'elenco degli amministratori delle chiavi quando si crea la chiave KMS nella visualizzazione di default o nella visualizzazione della policy.

 Warning

Poiché gli amministratori chiave sono autorizzati a modificare la politica chiave e creare concessioni, possono concedere a se stessi e ad altri AWS KMS autorizzazioni non specificate in questa politica.

I principali che dispongono dell'autorizzazione per gestire tag e alias possono anche controllare l'accesso a una chiave KMS. Per informazioni dettagliate, vedi [ABAC per AWS KMS](#).

 Note

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

L'esempio seguente mostra l'istruzione degli amministratori della chiave nella visualizzazione predefinita della console AWS KMS .

Key policy | Tags

Key policy Switch to policy view

Key administrators
Choose the IAM users and roles who can administer this key through the KMS API. You might need to add additional permissions for the users or roles to administer this key from this console. [Learn more](#)

Add Remove

< 1 >

☐	Name	Path	Type
☐	ExampleAdminRole	/	Role

Key deletion

☒ Allow key administrators to delete this key

L'esempio seguente mostra un esempio di istruzione degli amministratori della chiave nella visualizzazione della policy della console AWS KMS . Questa dichiarazione degli amministratori chiave si riferisce a una chiave KMS di crittografia simmetrica a regione singola.

Note

La AWS KMS console aggiunge gli amministratori chiave alla politica chiave sotto l'identificatore dell'istruzione. "Allow access for Key Administrators" La modifica di questo identificatore di istruzione potrebbe influire sul modo in cui la console visualizza gli aggiornamenti apportati all'istruzione.

```
{  
  "Sid": "Allow access for Key Administrators",  
  "Effect": "Allow",
```

```

"Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleAdminRole"},
"Action": [
  "kms:Create*",
  "kms:Describe*",
  "kms:Enable*",
  "kms:List*",
  "kms:Put*",
  "kms:Update*",
  "kms:Revoke*",
  "kms:Disable*",
  "kms:Get*",
  "kms>Delete*",
  "kms:TagResource",
  "kms:UntagResource",
  "kms:ScheduleKeyDeletion",
  "kms:CancelKeyDeletion",
  "kms:RotateKeyOnDemand"
],
"Resource": "*"
}

```

L'istruzione predefinita degli amministratori della chiave per la chiave KMS più comune, una chiave KMS di crittografia simmetrica mono-regione, concede le seguenti autorizzazioni. Per informazioni dettagliate su ciascuna autorizzazione, consultare [AWS KMS autorizzazioni](#).

Quando si utilizza la AWS KMS console per creare una chiave KMS, la console aggiunge gli utenti e i ruoli specificati all'elemento dell'istruzione Key Administrators.

Molte di queste autorizzazioni contengono il carattere jolly (*), che concede tutte le autorizzazioni che iniziano con il verbo specificato. Di conseguenza, quando AWS KMS aggiunge nuove operazioni API, gli amministratori chiave possono utilizzarle automaticamente. Non è necessario aggiornare le policy della chiave per includere le nuove operazioni. Se si preferisce limitare gli amministratori della chiave a un set fisso di operazioni API, è possibile [modificare la policy della chiave](#).

kms:Create*

Consente [kms:CreateAlias](#) e [kms:CreateGrant](#). (L'autorizzazione `kms:CreateKey` è valida solo in una policy IAM.)

kms:Describe*

Consente [kms:DescribeKey](#). L'autorizzazione `kms:DescribeKey` è richiesta per visualizzare la pagina dei dettagli della chiave per una chiave KMS nella Console di gestione AWS.

kms:Enable*

Permette [kms:EnableKey](#). Per le chiavi KMS di crittografia simmetrica, consente anche [kms:EnableKeyRotation](#).

kms:List*

Permette [kms:ListGrants](#), [kms:ListKeyPolicies](#) e [kms:ListResourceTags](#). (Le autorizzazioni `kms:ListAliases` e `kms:ListKeys`, che sono necessarie per visualizzare le chiavi KMS nella Console di gestione AWS, sono valide solo nelle policy IAM.)

kms:Put*

Consente [kms:PutKeyPolicy](#). Questa autorizzazione consente agli amministratori della chiave di modificare la policy della chiave per questa chiave KMS.

kms:Update*

Consente [kms:UpdateAlias](#) e [kms:UpdateKeyDescription](#). Per le chiavi multi-Regione, consente [kms:UpdatePrimaryRegion](#) su questa chiave KMS.

kms:Revoke*

Concede [kms:RevokeGrant](#), che permette agli amministratori della chiave di [eliminare una concessione](#) anche se non sono un [principale per il ritiro](#) nella concessione.

kms:Disable*

Permette [kms:DisableKey](#). Per le chiavi KMS di crittografia simmetrica, consente anche [kms:DisableKeyRotation](#).

kms:Get*

Permette [kms:GetKeyPolicy](#) e [kms:GetKeyRotationStatus](#). Per le chiavi KMS con materiale chiave importato, consente [kms:GetParametersForImport](#). Per le chiavi KMS asimmetriche, consente [kms:GetPublicKey](#). L'autorizzazione `kms:GetKeyPolicy` è richiesta per visualizzare la policy della chiave per una chiave KMS nella Console di gestione AWS.

kms>Delete*

Consente [kms>DeleteAlias](#). Per le chiavi con materiale chiave importato, consente [kms>DeleteImportedKeyMaterial](#). L'autorizzazione `kms>Delete*` non consente agli amministratori della chiave di eliminare la chiave KMS (`ScheduleKeyDeletion`).

kms:TagResource

Consente [kms:TagResource](#), per cui gli amministratori della chiave possono aggiungere tag alla chiave KMS. Poiché i tag possono essere utilizzati anche per controllare l'accesso alla chiave KMS, questa autorizzazione può consentire agli amministratori di permettere o negare l'accesso alla chiave KMS. Per informazioni dettagliate, vedi [ABAC per AWS KMS](#).

kms:UntagResource

Consente [kms:UntagResource](#), per cui gli amministratori della chiave possono eliminare tag dalla chiave KMS. Poiché i tag possono essere utilizzati per controllare l'accesso alla chiave, questa autorizzazione può consentire agli amministratori di permettere o negare l'accesso alla chiave KMS. Per informazioni dettagliate, vedi [ABAC per AWS KMS](#).

kms:ScheduleKeyDeletion

Consente [kms:ScheduleKeyDeletion](#), per cui gli amministratori della chiave possono [eliminare questa chiave KMS](#). Per eliminare questa autorizzazione, deselecta l'opzione Consenti agli amministratori della chiave di eliminare questa chiave.

kms:CancelKeyDeletion

Consente [kms:CancelKeyDeletion](#), per cui gli amministratori della chiave possono [annullare l'eliminazione di questa chiave KMS](#). Per eliminare questa autorizzazione, deselecta l'opzione Consenti agli amministratori della chiave di eliminare questa chiave.

kms:RotateKeyOnDemand

Consente [kms:RotateKeyOnDemand](#), che consente agli amministratori chiave di [eseguire la rotazione su richiesta del materiale chiave in questa chiave KMS](#).

AWS KMS aggiunge le seguenti autorizzazioni all'istruzione predefinita degli amministratori delle chiavi quando si creano chiavi per scopi speciali.

kms:ImportKeyMaterial

L'autorizzazione [kms:ImportKeyMaterial](#) consente agli amministratori della chiave di importare il materiale chiave nella chiave KMS. Questa autorizzazione è inclusa nella policy delle chiavi solo quando [crei una chiave KMS senza materiale della chiave](#).

kms:ReplicateKey

L'[kms:ReplicateKey](#) autorizzazione consente agli amministratori chiave di [creare una replica di una chiave primaria multiregionale in una regione diversa](#). AWS Questa autorizzazione è inclusa nella policy della chiave solo quando si crea una chiave primaria o di replica multi-Regione.

kms:UpdatePrimaryRegion

L'autorizzazione [kms:UpdatePrimaryRegion](#) consente agli amministratori della chiave di [modificare una replica di una chiave multi-Regione in una chiave primaria multi-Regione](#). Questa autorizzazione è inclusa nella policy della chiave solo quando si crea una chiave primaria o di replica multi-Regione.

Consente agli utenti della chiave di utilizzare la chiave KMS

La policy chiave predefinita creata dalla console per le chiavi KMS consente di scegliere utenti IAM e ruoli IAM nell'account e all'esterno Account AWS e renderli utenti chiave.

La console aggiunge due istruzioni di policy alla policy delle chiavi per gli utenti della chiave.

- [Utilizzare direttamente la chiave KMS](#) — La prima istruzione di policy delle chiavi consente agli utenti della chiave di utilizzare la chiave KMS direttamente per tutte le [operazioni di crittografia](#) per quel tipo di chiave KMS.
- [Usa la chiave KMS con AWS i servizi](#): la seconda dichiarazione politica concede agli utenti chiave il permesso di consentire ai AWS servizi integrati di utilizzare la chiave KMS AWS KMS per loro conto per proteggere risorse, come i bucket Amazon S3 e le tabelle Amazon DynamoDB.

Puoi aggiungere utenti IAM, ruoli IAM e altri Account AWS all'elenco degli utenti chiave quando crei la chiave KMS. È anche possibile modificare l'elenco con la visualizzazione predefinita della console per le policy delle chiavi, come illustrato nella seguente immagine. La visualizzazione predefinita per le policy delle chiavi si trova nella pagina dei dettagli delle chiavi. Per ulteriori informazioni su come consentire agli utenti di altri Account AWS di utilizzare la chiave KMS, consulta. [Autorizzazione per gli utenti in altri account di utilizzare una chiave KMS](#)

Note

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

Key users

The following IAM users and roles can use this key for cryptographic operations. They can also allow AWS services that are integrated with KMS to use the key on their behalf. [Learn more](#)

AddRemove

< 1 >

☐	Name	Path	Type
☐	ExampleRole	/	Role

Other AWS accounts

- arn:aws:iam::444455556666:root

Le istruzioni degli amministratori della chiave predefinite per una chiave KMS simmetrica a Regione singola concedono le seguenti autorizzazioni. Per informazioni dettagliate su ciascuna autorizzazione, consultare [AWS KMS autorizzazioni](#).

Quando usi la AWS KMS console per creare una chiave KMS, la console aggiunge gli utenti e i ruoli specificati all'Principalelemento in ogni istruzione key users.

Note

La AWS KMS console aggiunge gli utenti chiave alla politica chiave sotto gli identificatori "Allow use of the key" di dichiarazione e "Allow attachment of persistent

resources La modifica di questi identificatori delle istruzioni potrebbe influire sul modo in cui la console visualizza gli aggiornamenti apportati all'istruzione.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": [
    "arn:aws:iam::111122223333:role/ExampleRole",
    "arn:aws:iam::444455556666:root"
  ]},
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
},
{
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
  "Principal": {"AWS": [
    "arn:aws:iam::111122223333:role/ExampleRole",
    "arn:aws:iam::444455556666:root"
  ]},
  "Action": [
    "kms:CreateGrant",
    "kms:ListGrants",
    "kms:RevokeGrant"
  ],
  "Resource": "*",
  "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
}
```

Consente agli utenti della chiave di utilizzare una chiave KMS per le operazioni di crittografia

Gli utenti della chiave sono autorizzati a utilizzare direttamente la chiave KMS in tutte le [operazioni di crittografia supportate](#) nella chiave KMS. Possono inoltre utilizzare l'[DescribeKey](#) operazione per

ottenere informazioni dettagliate sulla chiave KMS nella AWS KMS console o utilizzare le AWS KMS operazioni API.

Per impostazione predefinita, la AWS KMS console aggiunge alla politica delle chiavi predefinita le istruzioni chiave degli utenti chiave, come quelle degli esempi seguenti. Dato che supportano diverse operazioni API, le operazioni nelle istruzioni della policy per le chiavi KMS di crittografia simmetrica, le chiavi KMS HMAC, le chiavi asimmetriche per la crittografia a chiave pubblica e le chiavi KMS asimmetriche per la firma e la verifica sono leggermente diverse.

Chiavi KMS di crittografia simmetrica

La console aggiunge l'istruzione seguente alla policy della chiave per le chiavi KMS di crittografia simmetrica.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:Decrypt",
    "kms:DescribeKey",
    "kms:Encrypt",
    "kms:GenerateDataKey*",
    "kms:ReEncrypt*"
  ],
  "Resource": "*"
}
```

Chiavi KMS HMAC

La console aggiunge l'istruzione seguente alla policy della chiave per le chiavi KMS HMAC.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:DescribeKey",
    "kms:GenerateMac",
    "kms:VerifyMac"
  ],
  "Resource": "*"
}
```



```
}
```

Chiavi KMS asimmetriche per la crittografia a chiave pubblica

La console aggiunge l'istruzione seguente alla policy delle chiavi per le chiavi KMS asimmetriche con un utilizzo di chiave Encrypt and decrypt (Crittografia e decrittografia).

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:DescribeKey",
    "kms:GetPublicKey"
  ],
  "Resource": "*"
}
```

Chiavi KMS asimmetriche per la firma e la verifica

La console aggiunge l'istruzione seguente alla policy delle chiavi per le chiavi KMS asimmetriche con un utilizzo di chiave Sign and verify (Firma e verifica).

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:DescribeKey",
    "kms:GetPublicKey",
    "kms:Sign",
    "kms:Verify"
  ],
  "Resource": "*"
}
```

Chiavi KMS asimmetriche per derivare segreti condivisi

La console aggiunge la seguente dichiarazione alla politica chiave per le chiavi KMS asimmetriche con un utilizzo chiave di Key agreement.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:DescribeKey",
    "kms:GetPublicKey",
    "kms:DeriveSharedSecret"
  ],
  "Resource": "*"
}
```

Le operazioni in queste istruzioni forniscono agli utenti della chiave le autorizzazioni seguenti.

[kms:Encrypt](#)

Permette agli utenti della chiave di crittografare i dati con questa chiave KMS.

[kms:Decrypt](#)

Permette agli utenti della chiave di decrittare i dati con questa chiave KMS.

[kms:DeriveSharedSecret](#)

Consente agli utenti chiave di ricavare segreti condivisi con questa chiave KMS.

[kms:DescribeKey](#)

Permette agli utenti della chiave di ottenere informazioni dettagliate su questa chiave KMS, compresi gli identificatori, la data di creazione e lo stato della chiave. Consente inoltre agli utenti principali di visualizzare i dettagli sulla chiave KMS nella console. AWS KMS

kms:GenerateDataKey*

Permette agli utenti della chiave di richiedere una chiave di dati simmetrica o una coppia di chiavi di dati asimmetriche per operazioni di crittografia sul lato client. La console utilizza il carattere jolly * per rappresentare l'autorizzazione per le seguenti operazioni API: [GenerateDataKey](#), [GenerateDataKeyWithoutPlaintext](#), [GenerateDataKeyPair](#), e.

[GenerateDataKeyPairWithoutPlaintext](#) Queste autorizzazioni sono valide solo per le chiavi KMS di crittografia simmetrica che crittografano le chiavi di dati.

[km: GenerateMac](#)

Consente agli utenti della chiave di utilizzare una chiave KMS HMAC per generare un tag HMAC.

[km: GetPublicKey](#)

Permette agli utenti della chiave di scaricare la chiave pubblica della chiave KMS asimmetrica. Le parti con cui condividi questa chiave pubblica possono crittografare i dati all'esterno di AWS KMS. Questi testi cifrati possono essere decrittati solo chiamando l'operazione [Decrypt](#) in AWS KMS.

[km: * ReEncrypt](#)

Permette agli utenti della chiave di crittografare nuovamente i dati originariamente crittografati con questa chiave KMS o di utilizzare questa chiave KMS per ricrittografare dati crittografati in precedenza. L'[ReEncrypt](#) operazione richiede l'accesso alle chiavi KMS di origine e di destinazione. A tal fine, puoi concedere l'autorizzazione `kms:ReEncryptFrom` sulla chiave KMS di origine e l'autorizzazione `kms:ReEncryptTo` sulla chiave KMS di destinazione. Per semplicità, però, la console consente `kms:ReEncrypt*` (con il carattere jolly `*`) su entrambe le chiavi KMS.

[kms:Sign](#)

Permette agli utenti della chiave di firmare messaggi con questa chiave KMS.

[kms:Verify](#)

Permette agli utenti della chiave di verificare le firme con questa chiave KMS.

[km: VerifyMac](#)

Consente agli utenti della chiave di utilizzare una chiave KMS HMAC per verificare un tag HMAC.

Consente agli utenti della chiave di utilizzare la chiave KMS con i servizi AWS

La politica delle chiavi predefinita nella console offre inoltre agli utenti chiave le autorizzazioni di concessione di cui hanno bisogno per proteggere i propri dati nei AWS servizi che utilizzano le sovvenzioni. AWS i servizi spesso utilizzano le sovvenzioni per ottenere autorizzazioni specifiche e limitate all'uso di una chiave KMS.

[Questa dichiarazione politica chiave consente all'utente principale di creare, visualizzare e revocare le concessioni sulla chiave KMS, ma solo quando la richiesta di operazione di concessione proviene](#)

[da un servizio integrato con AWS KMS](#) La condizione [kms: GrantIsFor AWSResource](#) policy non consente all'utente di chiamare direttamente queste operazioni di concessione. Se l'utente chiave lo consente, un AWS servizio può creare una concessione per conto dell'utente che consente al servizio di utilizzare la chiave KMS per proteggere i dati dell'utente.

Gli utenti della chiave hanno bisogno di queste autorizzazioni di concessione per utilizzare la loro chiave KMS con i servizi integrati, ma queste autorizzazioni non sono sufficienti. Gli utenti della chiave hanno bisogno dell'autorizzazione anche per utilizzare i servizi integrati. Per informazioni dettagliate su come concedere agli utenti l'accesso a un AWS servizio che si integra con AWS KMS, consulta la documentazione del servizio integrato.

```
{
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:CreateGrant",
    "kms:ListGrants",
    "kms:RevokeGrant"
  ],
  "Resource": "*",
  "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
}
```

Ad esempio, gli utenti della chiave possono utilizzare queste autorizzazioni sulla chiave KMS nei modi seguenti.

- Usa questa chiave KMS con Amazon Elastic Block Store (Amazon EBS) e Amazon Elastic Compute Cloud (EC2Amazon) per collegare un volume EBS crittografato a un'istanza. EC2 L'utente della chiave concede implicitamente ad Amazon l' EC2 autorizzazione a utilizzare la chiave KMS per collegare il volume crittografato all'istanza. Per ulteriori informazioni, consulta [Come utilizza Amazon Elastic Block Store \(Amazon EBS\) AWS KMS](#).
- Utilizza questa chiave KMS con Amazon Redshift per avviare un cluster crittografato. L'utente della chiave offre implicitamente a Amazon Redshift l'autorizzazione a utilizzare la chiave KMS per avviare il cluster crittografato e creare snapshot crittografate. Per ulteriori informazioni, consultare [Come utilizza Amazon Redshift AWS KMS](#).
- Utilizzare questa chiave KMS con altri [servizi AWS integrati con AWS KMS](#) che utilizzano concessioni per creare, gestire o utilizzare le risorse crittografate con quei servizi.

La policy delle chiavi predefinita consente agli utenti della chiave di delegare l'autorizzazione di concessione a tutti i servizi integrati che utilizzano le concessioni. Tuttavia, puoi creare una politica di chiave personalizzata che limiti l'autorizzazione a servizi specifici. AWS Per ulteriori informazioni, consulta la chiave di condizione [kms:ViaService](#).

Visualizza una politica chiave

Puoi visualizzare la politica chiave per una [chiave gestita dal AWS KMS cliente](#) o [Chiave gestita da AWS](#) nel tuo account utilizzando la AWS KMS console o l'[GetKeyPolicy](#) operazione nell' AWS KMS API. Non è possibile utilizzare queste tecniche per visualizzare la policy delle chiavi di una chiave KMS in un Account AWS diverso.

Per ulteriori informazioni sulle politiche AWS KMS chiave, consulta [Politiche chiave in AWS KMS](#). Per informazioni su come determinare quali utenti e ruoli hanno accesso a una chiave KMS, consulta [the section called "Determinazione dell'accesso"](#).

Utilizzo della AWS KMS console

Gli utenti autorizzati possono visualizzare la policy delle chiavi per una [Chiave gestita da AWS](#) o una [chiave gestita dal cliente](#) nella scheda Key policy (Policy delle chiavi) della Console di gestione AWS.

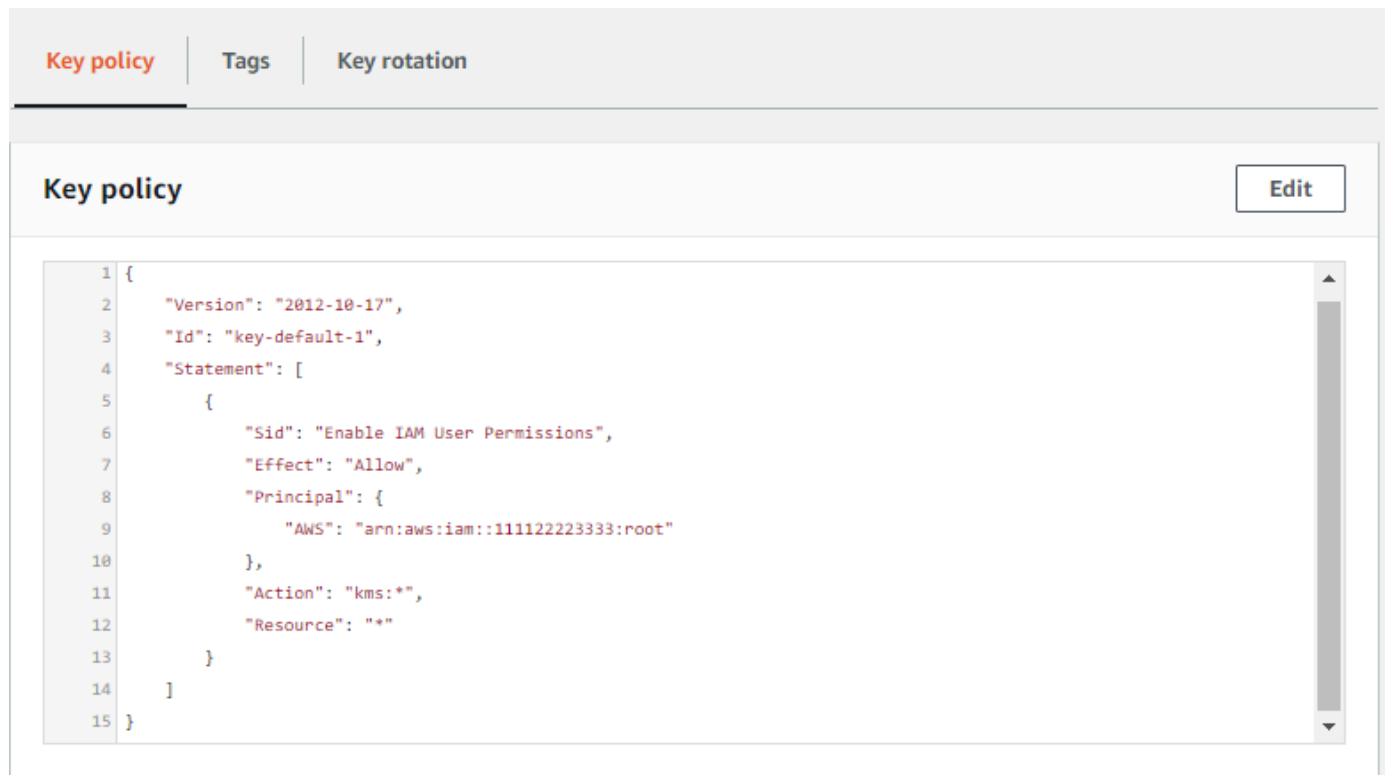
Per visualizzare la politica chiave per una chiave KMS in Console di gestione AWS, devi disporre delle autorizzazioni [kms: ListAliases](#), [kms: DescribeKey](#) e [kms: GetKeyPolicy](#).

1. [Accedi a Console di gestione AWS e apri la console AWS Key Management Service \(AWS KMS in /kms. <https://console.aws.amazon.com>](#)
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Per visualizzare le chiavi dell'account che AWS crea e gestisce per te, nel riquadro di navigazione, scegli chiavi gestite.AWS Per visualizzare le chiavi nell'account creato e gestito dall'utente, nel riquadro di navigazione, seleziona Chiavi gestite dal cliente.
4. Nell'elenco di chiavi KMS, scegliere l'alias o l'ID chiave della chiave KMS che si intende esaminare.
5. Scegli la scheda Policy della chiave.

Nella tab Policy chiave è possibile che venga visualizzato il documento della policy delle chiavi. Si tratta della visualizzazione policy. Nelle istruzioni della policy delle chiavi è possibile

visualizzare i principali a cui la policy ha concesso l'accesso alla chiave KMS e le operazioni che tali entità possono eseguire.

Nell'esempio seguente viene illustrata la visualizzazione policy per la [policy di chiave predefinita](#).



In alternativa, se hai creato la chiave KMS in Console di gestione AWS, vedrai la visualizzazione predefinita con le sezioni Amministratori delle chiavi, Eliminazione delle chiavi e Utenti chiave. Per visualizzare il documento di policy delle chiavi, scegliere Switch to policy view (Passa alla visualizzazione policy).

Nell'esempio seguente viene illustrata la visualizzazione predefinita per la [policy di chiave predefinita](#).

Key policy

Tags

Key rotation

Key policy

Switch to policy view

Key administrators

Choose the IAM users and roles who can administer this key through the KMS API. You might need to add additional permissions for the users or roles to administer this key from this console. [Learn more](#)

Add

Remove

Q

< 1 >

	Name	Path	Type
Empty Resources			
No resources to display			

Key deletion

☐ Allow key administrators to delete this key

Key users

The following IAM users and roles can use this key to encrypt and decrypt data from within applications and when using AWS services integrated with KMS. [Learn more](#)

Add

Remove

Q

< 1 >

	Name	Path	Type
Empty Resources			
No resources to display			

Utilizzo dell' AWS KMS API

Per ottenere la politica chiave per una chiave KMS nella tua Account AWS, utilizza l'[GetKeyPolicy](#) operazione nell' AWS KMS API. Non è possibile utilizzare questa operazione per visualizzare una policy di chiave in un account diverso.

L'esempio seguente utilizza il [get-key-policy](#) comando contenuto in AWS Command Line Interface (AWS CLI), ma puoi utilizzare qualsiasi AWS SDK per effettuare questa richiesta.

Il parametro `PolicyName` è obbligatorio, anche se l'unico valore valido è `default`. Inoltre, questo comando richiede l'output nel testo, anziché in JSON, per rendere la visualizzazione più semplice.

Prima di eseguire questo comando, sostituisci l'ID chiave di esempio con un ID valido dell'account.

```
$ aws kms get-key-policy --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --policy-name default --output text
```

La risposta deve essere simile a quella seguente, che restituisce la [policy di chiave predefinita](#).

JSON

```
{
  "Version": "2012-10-17",
  "Id": "key-consolepolicy-3",
  "Statement": [ {
    "Sid": "EnableIAMUserPermissions",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:root"
    },
    "Action": "kms:*",
    "Resource": "*"
  } ]
}
```

Modificare una policy della chiave

Puoi modificare la politica chiave per una chiave KMS nel tuo Account AWS sistema utilizzando l'[PutKeyPolicy](#) operazione Console di gestione AWS o. Non è possibile utilizzare queste tecniche per modificare la policy delle chiavi di una chiave KMS in un Account AWS diverso.

Quando modifichi una policy delle chiavi, ricorda le seguenti regole:

- Puoi visualizzare la policy delle chiavi per una [Chiave gestita da AWS](#) o per una [chiave gestita dal cliente](#), ma puoi modificare la policy delle chiavi solo per una chiave gestita dal cliente. Le politiche di Chiavi gestite da AWS vengono create e gestite dal AWS servizio che ha creato la chiave KMS nel tuo account. Non puoi visualizzare o modificare la policy delle chiavi per una [Chiave di proprietà di AWS](#).

- Puoi aggiungere o rimuovere utenti IAM, ruoli IAM e Account AWS nella policy chiave e modificare le azioni consentite o negate per tali responsabili. Per ulteriori informazioni sui metodi per specificare principali e autorizzazioni in una policy delle chiavi, consulta [Policy delle chiavi](#).
- Non puoi aggiungere gruppi IAM a una policy delle chiavi, ma puoi aggiungere più utenti IAM e ruoli IAM. Per ulteriori informazioni, consulta [Autorizzazione per più principali IAM di accedere a una chiave KMS](#).
- Se aggiungi elementi esterni Account AWS a una policy chiave, devi utilizzare anche le policy IAM negli account esterni per concedere le autorizzazioni agli utenti, ai gruppi o ai ruoli IAM in tali account. Per ulteriori informazioni, consulta [Autorizzazione per gli utenti in altri account di utilizzare una chiave KMS](#).
- Il documento di policy delle chiavi risultante non può superare i 32 KB (32.768 byte).

Come modificare una policy delle chiavi

Puoi modificare una policy chiave in tre diversi modi, ognuno dei quali è illustrato nelle seguenti sezioni.

Argomenti

- [Utilizzo della visualizzazione predefinita della Console di gestione AWS](#)
- [Utilizzo della visualizzazione delle politiche Console di gestione AWS](#)
- [Utilizzo dell'API AWS KMS](#)

Utilizzo della visualizzazione predefinita della Console di gestione AWS

Puoi utilizzare la console per modificare una policy delle chiavi mediante un'interfaccia grafica denominata visualizzazione predefinita.

Se le procedure seguenti non corrispondono a ciò che viene visualizzato nella console, è possibile che questa policy delle chiavi non sia stata creata con la console oppure che sia stata modificata in un modo non supportato dalla visualizzazione predefinita della console. In questo caso, segui i passaggi descritti in [Utilizzo della visualizzazione delle politiche Console di gestione AWS](#) o [Utilizzo dell'API AWS KMS](#).

1. Visualizza la policy delle chiavi per una chiave gestita dal cliente come descritto in [Utilizzo della AWS KMS console](#). (Non è possibile modificare le politiche chiave di Chiavi gestite da AWS.)
2. Decidere cosa modificare.

- Per aggiungere o rimuovere [amministratori delle chiavi](#) e per consentire o impedire agli amministratori di [eliminare la chiave KMS](#), utilizza i controlli nella sezione Key Administrators della pagina. Gli amministratori delle chiavi gestiscono la chiave KMS, possono abilitare e disabilitare la stessa, impostare la policy delle chiavi e [abilitare la rotazione delle chiavi](#).
- Per aggiungere o rimuovere [utenti chiave](#) e per consentire o impedire l'uso della chiave KMS Account AWS da parte di esterni, utilizza i controlli nella sezione Utenti chiave della pagina. Gli utenti della chiave possono utilizzare la chiave KMS nelle [operazioni di crittografia](#), ad esempio crittografia, decrittografia, ricrittografia e generazione di chiavi di dati.

Utilizzo della visualizzazione delle politiche Console di gestione AWS

Puoi utilizzare la console per modificare un documento di policy delle chiavi mediante la visualizzazione policy della console.

1. Visualizza la policy delle chiavi per una chiave gestita dal cliente come descritto in [Utilizzo della AWS KMS console](#). (Non è possibile modificare le politiche chiave di Chiavi gestite da AWS.)
2. Nella sezione Policy della chiave, scegli Passa alla visualizzazione della policy.
3. Scegli Modifica.
4. Decidere cosa modificare.
 - Per aggiungere una nuova dichiarazione, scegli Aggiungi nuova dichiarazione. Quindi, puoi selezionare le azioni, i principi e le condizioni per la nuova dichiarazione politica chiave dalle opzioni elencate nel pannello di creazione delle politiche o inserire manualmente gli elementi della dichiarazione politica.
 - Per rimuovere una dichiarazione dalla tua politica chiave, seleziona la dichiarazione e scegli Rimuovi. Esamina l'informativa politica selezionata e conferma che desideri rimuoverla. Se decidi di non voler procedere con la rimozione dell'informativa, scegli Annulla.
 - Per modificare una dichiarazione politica chiave esistente, seleziona la dichiarazione. Quindi, puoi utilizzare il pannello di creazione delle dichiarazioni per scegliere elementi specifici che desideri modificare o modificare manualmente l'istruzione.
5. Scegli Save changes (Salva modifiche).

Utilizzo dell'API AWS KMS

Puoi utilizzare l'[PutKeyPolicy](#) operazione per modificare la politica chiave di una chiave KMS nel tuo Account AWS. Non è possibile utilizzare questa API per una chiave KMS in un Account AWS differente.

1. Utilizza l'[GetKeyPolicy](#) operazione per ottenere il documento di politica chiave esistente, quindi salva il documento di politica chiave in un file. Per il codice di esempio in più linguaggi di programmazione, consulta [Utilizzo GetKeyPolicy con un AWS SDK o una CLI](#).
2. Aprire il documento di policy delle chiavi in un editor di testo, modificarlo e salvare il file.
3. Utilizza l'[PutKeyPolicy](#) operazione per applicare il documento di policy chiave aggiornato alla chiave KMS. Per il codice di esempio in più linguaggi di programmazione, consulta [Utilizzo PutKeyPolicy con un AWS SDK o una CLI](#).

Per un esempio di copia di una politica chiave da una chiave KMS a un'altra, vedi l'[GetKeyPolicy esempio](#) nel Command Reference. AWS CLI

Autorizzazioni per i AWS servizi nelle politiche chiave

Molti AWS servizi lo utilizzano AWS KMS keys per proteggere le risorse che gestiscono. Quando un servizio utilizza [Chiavi di proprietà di AWS](#) o [Chiavi gestite da AWS](#), il servizio stabilisce e mantiene le policy della chiave per queste chiavi KMS.

Tuttavia, quando si utilizza una [chiave gestita dal cliente](#) con un servizio AWS, è l'utente che imposta e mantiene la policy della chiave. Tale policy della chiave deve concedere al servizio le autorizzazioni minime necessarie per proteggere la risorsa per conto dell'utente. Si consiglia di seguire il principio del privilegio minimo: concedere al servizio soltanto le autorizzazioni necessarie. È possibile farlo in modo efficace scoprendo di quali autorizzazioni il servizio ha bisogno e utilizzando le [chiavi di condizione globali AWS](#) e le [chiavi di condizione AWS KMS](#) per perfezionare le autorizzazioni.

Per trovare le autorizzazioni richieste dal servizio su una chiave gestita dal cliente, consultare la documentazione di crittografia per il servizio. L'elenco seguente include collegamenti alla documentazione di alcuni servizi:

- AWS CloudTrail autorizzazioni: [configura le politiche AWS KMS chiave per CloudTrail](#)
- Autorizzazioni Amazon Elastic Block Store - [Amazon EC2 User Guide](#) e [Amazon EC2 User Guide](#)
- AWS Lambda permissions - [Crittografia dei dati a riposo per Lambda](#)
- Autorizzazioni Amazon Q - [Crittografia dei dati per Amazon Q](#)

- [AWS KMS Autorizzazioni di Amazon Relational Database Service: gestione delle chiavi](#)
- Gestione dei segreti AWSpermessi: [autorizzazione all'uso della chiave KMS](#)
- Autorizzazioni Amazon Simple Queue Service - Gestione delle chiavi [Amazon SQS](#)

Utilizzo delle politiche IAM con AWS KMS

Puoi utilizzare le policy IAM, insieme alle [policy chiave](#), alle [sovvenzioni](#) e alle policy degli [endpoint VPC](#), per controllare l'accesso al tuo account. AWS KMS keys AWS KMS

Note

Per utilizzare una policy IAM per controllare l'accesso a una chiave KMS, la policy chiave per la chiave KMS deve concedere all'account l'autorizzazione all'utilizzo delle policy IAM. In particolare, la policy chiave deve includere [l'istruzione di policy che abilita le policy IAM](#). Questa sezione spiega come utilizzare le policy IAM per controllare l'accesso alle operazioni. AWS KMS Per informazioni più generiche su IAM, consulta la [Guida per l'utente di IAM](#).

Tutte le chiavi KMS devono disporre di una policy chiave. Le policy IAM sono facoltativi. Per utilizzare una policy IAM per controllare l'accesso a una chiave KMS, la policy chiave per la chiave KMS deve concedere all'account l'autorizzazione all'utilizzo delle policy IAM. In particolare, la policy chiave deve includere [l'istruzione di policy che abilita le policy IAM](#).

Le policy IAM possono controllare l'accesso a qualsiasi AWS KMS operazione. A differenza delle policy chiave, le policy IAM possono controllare l'accesso a più chiavi KMS e fornire le autorizzazioni per le operazioni di diversi servizi correlati AWS . Ma le policy IAM sono particolarmente utili per controllare l'accesso alle operazioni [CreateKey](#), ad esempio quelle operazioni che non possono essere controllate da una policy chiave perché non coinvolgono alcuna chiave KMS particolare.

Se accedi AWS KMS tramite un endpoint Amazon Virtual Private Cloud (Amazon VPC), puoi anche utilizzare una policy sugli endpoint VPC per limitare l'accesso alle tue AWS KMS risorse quando usi l'endpoint. Ad esempio, quando utilizzi l'endpoint VPC, potresti consentire solo ai tuoi responsabili di accedere Account AWS alle tue chiavi gestite dai clienti. Per i dettagli, consulta le [politiche degli endpoint VPC](#).

Per informazioni sulla scrittura e sulla formattazione di un documento delle policy JSON, vedi la sezione [Documentazione di riferimento alla policy JSON IAM](#) nella Guida per l'utente di IAM.

Puoi utilizzare policy IAM nei seguenti modi:

- Associa una policy di autorizzazioni a un ruolo per le autorizzazioni federative o interaccount: puoi allegare una policy IAM a un ruolo IAM per abilitare la federazione delle identità, consentire le autorizzazioni tra account o concedere autorizzazioni alle applicazioni in esecuzione su istanze. EC2 Per ulteriori informazioni sui vari casi d'uso per i ruoli IAM, consulta [Ruoli IAM](#) nella Guida per l'utente di IAM.
- Allegare una policy di autorizzazioni a un utente o a un gruppo – Puoi collegare una policy che consente a un utente o a un gruppo di utenti di richiamare operazioni AWS KMS . Tuttavia, le best practice IAM consigliano di utilizzare identità con credenziali temporanee, come i ruoli IAM, quando possibile.

L'esempio seguente mostra una policy IAM con autorizzazioni. AWS KMS Questa policy consente alle identità IAM a cui è collegata di ottenere tutte le chiavi KMS e gli alias.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:ListKeys",
        "kms:ListAliases"
      ],
      "Resource": "*"
    }
  ]
}
```

Come tutte le policy IAM, questa policy non ha un elemento `Principal`. Quando colleghi una policy IAM a un'identità IAM, tale identità ottiene le autorizzazioni specificate nella policy.

Per una tabella che mostra tutte le azioni AWS KMS API e le risorse a cui si applicano, consulta [la Riferimento per le autorizzazioni](#).

Autorizzazione per più principali IAM di accedere a una chiave KMS

I gruppi IAM non sono principali validi in una policy chiave. Per consentire a più utenti e ruoli di accedere a una chiave KMS, procedi in uno dei seguenti modi:

- Usa un ruolo IAM come principale nella policy delle chiavi. Più utenti autorizzati possono assumere il ruolo secondo necessità. Per i dettagli, consulta [Ruoli IAM](#) nella Guida per l'utente IAM.

Sebbene sia possibile collegare più utenti IAM in una policy delle chiavi, questa procedura non è consigliata perché richiede l'aggiornamento della policy delle chiavi ogni volta che l'elenco di utenti autorizzati viene modificato. Inoltre, le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

- Utilizza una policy IAM per collegare l'autorizzazione a un gruppo IAM. Per fare ciò, assicurati che la policy delle chiavi includa la dichiarazione che [consente alle policy IAM di consentire l'accesso alla chiave KMS](#), [crea una policy IAM](#) che consenta l'accesso alla chiave KMS e quindi [collega tale policy a un gruppo IAM](#) che contenga gli utenti IAM autorizzati. L'utilizzo di questo approccio non richiede l'aggiornamento di policy quando l'elenco degli utenti autorizzati viene modificato. È sufficiente aggiungere o rimuovere tali utenti dal gruppo IAM appropriato. Per i dettagli, consulta la sezione [Gruppi di utenti IAM](#) nella Guida per l'utente IAM

Per ulteriori informazioni su come le policy AWS KMS chiave e le policy IAM interagiscono, consulta [Risoluzione dei problemi relativi alle autorizzazioni AWS KMS](#).

Best practice per le policy IAM

Garantire l'accesso a AWS KMS keys è fondamentale per la sicurezza di tutte le AWS risorse. Le chiavi KMS vengono utilizzate per proteggere molte risorse sensibili nell' Account AWS. Prenditi il tempo necessario per progettare le [policy chiave](#), le [policy IAM](#), le [sovvenzioni](#) e le policy degli endpoint VPC che controllano l'accesso alle tue chiavi KMS.

Nelle istruzioni delle policy IAM che controllano l'accesso alle chiavi KMS, utilizza il [principio del privilegio minimo](#). Assegna ai principali IAM solo le autorizzazioni necessarie solo per quelle chiavi KMS che devono utilizzare o gestire.

Le seguenti best practice si applicano alle policy IAM che controllano l'accesso a AWS KMS chiavi e alias. Per una guida generale alle best practice delle policy IAM, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

Utilizzo delle policy delle chiavi

Quando possibile, concedi le autorizzazioni nelle policy chiave che interessano una chiave KMS anziché in una policy IAM applicabile a molte chiavi KMS, incluse quelle di altri Account AWS. Ciò è particolarmente importante per le autorizzazioni sensibili come [kms: PutKeyPolicy](#) e [kms: ScheduleKeyDeletion](#) ma anche per le operazioni crittografiche che determinano il modo in cui i dati sono protetti.

CreateKey Limita le autorizzazioni

Concedi il permesso di creare chiavi ([kms: CreateKey](#)) solo ai principali che ne hanno bisogno. I principali che creano una chiave KMS impostano anche le policy delle chiavi, in modo che possano concedere a se stessi e agli altri l'autorizzazione per utilizzare e gestire le chiavi KMS che creano. Quando concedi questa autorizzazione, è consigliabile limitarla utilizzando [le condizioni delle policy](#). Ad esempio, puoi utilizzare la KeySpec condizione [kms:](#) per limitare l'autorizzazione alle chiavi KMS di crittografia simmetrica.

Specificare le chiavi KMS in una policy IAM

Come best practice, specifica l'[ARN di chiave](#) di ciascuna chiave KMS a cui si applica l'autorizzazione nell'elemento Resource dell'istruzione della policy. Questa procedura limita l'autorizzazione per le chiavi KMS a quella richiesta dal principale. Ad esempio, questo elemento Resource elenca solo le chiavi KMS che il principale deve utilizzare.

```
"Resource": [  
  "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"  
]
```

Quando non è possibile specificare le chiavi KMS, utilizza un Resource valore che limiti l'accesso alle chiavi KMS in una regione affidabile e affidabile, ad esempio. Account AWS `arn:aws:kms:region:account:key/*` Oppure limita l'accesso alle chiavi KMS in tutte le regioni (*) di una persona affidabile, ad esempio. Account AWS `arn:aws:kms:*:account:key/*`

Non puoi utilizzare un [ID chiave](#), un [nome alias](#) o l'[ARN dell'alias](#) per rappresentare una chiave KMS nel campo Resource di una policy IAM. Se si specifica un alias ARN, la policy si applica all'alias e non alla chiave KMS. Per informazioni generali sulle policy IAM, consulta [Controllo dell'accesso agli alias](#).

Evita "Resource": "*" in una policy IAM

Utilizza i caratteri jolly (*) in modo giudizioso. In una policy delle chiavi, il carattere jolly nell'elemento Resource rappresenta la chiave KMS a cui è collegata la policy delle chiavi. Tuttavia, in una policy IAM, un solo carattere jolly presente nell'elemento Resource ("Resource": "*") applica le autorizzazioni a tutte le chiavi KMS relative a tutte le chiavi KMS Account AWS che l'account del principale è autorizzato a utilizzare. Ciò potrebbe includere [le chiavi KMS in altri account Account AWS](#), oltre alle chiavi KMS nell'account del principale.

Ad esempio, per utilizzare una chiave KMS in un'altra Account AWS, un principale necessita dell'autorizzazione della politica chiave della chiave KMS nell'account esterno e di una politica IAM nel proprio account. Supponiamo che un account arbitrario conceda l'autorizzazione Account AWS [kms:Decrypt](#) sulle sue chiavi KMS. In tal caso, una policy IAM nell'account che concede a un ruolo l'autorizzazione kms:Decrypt per tutte le chiavi KMS ("Resource": "*") soddisferebbe la parte IAM del requisito. Di conseguenza, i principali che possono assumere tale ruolo possono ora decrittare i testi cifrati utilizzando la chiave KMS nell'account non attendibile. Le voci relative alle relative operazioni vengono visualizzate nei CloudTrail registri di entrambi gli account.

In particolare, evita di utilizzare "Resource": "*" in un'istruzione della policy che consente le seguenti operazioni API. Queste operazioni possono essere richiamate sulle chiavi KMS di altri Account AWS

- [DescribeKey](#)
- [GetKeyRotationStatus](#)
- [Operazioni crittografiche \(Encrypt, Decrypt,,,,, GenerateDataKey, GenerateDataKeyPair, GenerateDataKeyWithoutPlaintextSign GenerateDataKeyPairWithoutPlaintextGetPublicKey, ReEncryptVerify\)](#)
- [CreateGrant](#), [ListGrants](#), [ListRetirableGrants](#), [RetireGrant](#), [RevokeGrant](#)

Quando utilizzare "Resource": "*"

In una policy IAM usi un carattere jolly nell'elemento Resource solo per le autorizzazioni che lo richiedono. Solo le seguenti autorizzazioni richiedono l'elemento "Resource": "*".

- [km: CreateKey](#)
- [km: GenerateRandom](#)
- [km: ListAliases](#)
- [km: ListKeys](#)

- [Autorizzazioni per archivi di chiavi personalizzati, come kms: CreateCustomKeyStore e kms: ConnectCustomKeyStore](#)

Note

Le autorizzazioni per le operazioni con alias ([kms:CreateAlias](#), [kms:](#), [kms: UpdateAlias](#), [DeleteAlias](#)) devono essere allegate all'alias e alla chiave KMS. Puoi utilizzare "Resource": "*" in una policy IAM per rappresentare gli alias e le chiavi KMS oppure per specificare gli alias e le chiavi KMS nell'elemento Resource. Per alcuni esempi, consulta [Controllo dell'accesso agli alias](#).

Negli esempi riportati in questo argomento vengono fornite ulteriori informazioni e indicazioni per la progettazione di policy IAM per le chiavi KMS. Per le migliori pratiche IAM per tutte le AWS risorse, consulta le [migliori pratiche di sicurezza in IAM nella IAM User Guide](#).

Specificazione delle chiavi KMS nelle istruzioni della policy IAM

Puoi utilizzare una policy IAM per consentire a un principale di utilizzare o gestire le chiavi KMS. Le chiavi KMS sono specificate nell'elemento Resource dell'istruzione della policy.

- Per specificare una chiave KMS in un'istruzione delle policy IAM, devi utilizzare l'[ARN della chiave](#). Non è possibile utilizzare un [ID chiave](#), un [nome alias](#) o l'[ARN di alias](#) per identificare una chiave KMS in un'istruzione della policy IAM.

Ad esempio: "Resource": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"

Per controllare l'accesso a una chiave KMS in base ai relativi alias, usa le chiavi [kms: RequestAlias](#) o [kms: condition. ResourceAliases](#) Per informazioni dettagliate, vedi [ABAC per AWS KMS](#).

Utilizzate un alias ARN come risorsa solo in una dichiarazione di policy che controlla l'accesso alle operazioni di alias, ad esempio [CreateAlias](#), o. [UpdateAliasDeleteAlias](#) Per informazioni dettagliate, vedi [Controllo dell'accesso agli alias](#).

- Per specificare più chiavi KMS nell'account e nella regione, utilizza i caratteri jolly (*) nelle posizioni dell'ID risorsa e della regione dell'ARN di chiave.

Ad esempio, per specificare tutte le chiavi KMS nella Regione Stati Uniti occidentali (Oregon) di un account, utilizza "Resource": "arn:aws:kms:us-west-2:111122223333:key/*". Per specificare tutte le chiavi KMS in tutte le Regioni dell'account, utilizza "Resource": "arn:aws:kms:*:111122223333:key/*".

- Per rappresentare tutte le chiavi KMS, utilizza solo un carattere jolly ("*"). Utilizza questo formato per operazioni che non utilizzano alcuna chiave KMS particolare, vale a dire [CreateKey](#), [GenerateRandom](#), [ListAliasesListKeys](#)

Quando scrivi le istruzioni delle policy, come [best practice](#) è consigliabile limitare le chiavi KMS a quelle che i principali devono utilizzare, anziché concedere loro l'accesso a tutte le chiavi KMS.

Ad esempio, la seguente dichiarazione di policy IAM consente al principale di richiamare le operazioni [DescribeKey](#), [GenerateDataKey](#), [Decrypt](#) solo sulle chiavi KMS elencate nell'Resourceelemento dell'informativa politica. La specifica delle chiavi KMS in base all'ARN della chiave, che è una best practice, garantisce che le autorizzazioni siano limitate solo alle chiavi KMS indicate.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:DescribeKey",
      "kms:GenerateDataKey",
      "kms:Decrypt"
    ],
    "Resource": [
      "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "arn:aws:kms:us-west-2:111122223333:key/01234abcd-12ab-34cd-56ef-1234567890ab"
    ]
  }
}
```

Per applicare l'autorizzazione a tutte le chiavi KMS di un particolare trusted Account AWS, puoi utilizzare caratteri jolly (*) nelle posizioni Region e Key ID. Ad esempio, la seguente istruzione di policy consente al principale di richiamare le operazioni specificate in tutte le chiavi KMS in due account di esempio attendibili.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:DescribeKey",
      "kms:GenerateDataKey",
      "kms:GenerateDataKeyPair"
    ],
    "Resource": [
      "arn:aws:kms:*:111122223333:key/*",
      "arn:aws:kms:*:444455556666:key/*"
    ]
  }
}
```

Inoltre puoi utilizzare un carattere jolly ("*") da solo nell'elemento Resource. Poiché consente l'accesso a tutte le chiavi KMS che l'account ha l'autorizzazione di utilizzare, è consigliato principalmente per le operazioni senza una particolare chiave KMS e per le istruzioni Deny. Puoi inoltre utilizzarlo nelle istruzioni di policy che consentono esclusivamente operazioni di sola lettura meno sensibili. Per determinare se un' AWS KMS operazione coinvolge una particolare chiave KMS, cerca il valore della chiave KMS nella colonna Risorse della tabella in [the section called “Riferimento per le autorizzazioni”](#)

Ad esempio, l'istruzione di policy riportata di seguito utilizza un effetto Deny per impedire ai principali di utilizzare le operazioni specificate per qualsiasi chiave KMS. Viene utilizzato un carattere jolly nell'elemento Resource per rappresentare tutte le chiavi KMS.

JSON

```
{
```

```
"Version": "2012-10-17",
"Statement": {
  "Effect": "Deny",
  "Action": [
    "kms:CreateKey",
    "kms:PutKeyPolicy",
    "kms:CreateGrant",
    "kms:ScheduleKeyDeletion"
  ],
  "Resource": "*"
}
```

Nell'istruzione di policy seguente viene utilizzato un carattere jolly da solo per rappresentare tutte le chiavi KMS. Tuttavia consente solo operazioni di sola lettura meno sensibili e operazioni che non si applicano a nessuna particolare chiave KMS.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:CreateKey",
      "kms:ListKeys",
      "kms:ListAliases",
      "kms:ListResourceTags"
    ],
    "Resource": "*"
  }
}
```

Esempi di policy IAM

Questa sezione include esempi di policy IAM che concedono autorizzazioni per varie azioni AWS KMS .

⚠ Important

Alcune delle autorizzazioni nelle seguenti policy sono consentite solo quando anche la policy delle chiavi della chiave KMS le consente. Per ulteriori informazioni, consulta [Riferimento per le autorizzazioni](#).

Per informazioni sulla scrittura e sulla formattazione di un documento di policy JSON, consulta la [Documentazione di riferimento sulla policy IAM JSON](#) nella Guida per l'utente di IAM.

Esempi

- [Consenti a un utente di visualizzare le chiavi KMS nella console AWS KMS](#)
- [Consentire a un utente di creare chiavi KMS](#)
- [Consenti a un utente di crittografare e decrittografare con qualsiasi chiave KMS in una determinata chiave Account AWS](#)
- [Consenti a un utente di crittografare e decrittografare con qualsiasi chiave KMS in una regione e in una regione specifiche Account AWS](#)
- [Consentire a un utente di eseguire la crittografia e la decrittazione con chiavi KMS specifiche](#)
- [Impedire a un utente di disabilitare o eliminare le chiavi KMS](#)

Consenti a un utente di visualizzare le chiavi KMS nella console AWS KMS

La seguente policy IAM consente agli utenti l'accesso in sola lettura alla console. AWS KMS Gli utenti con queste autorizzazioni possono visualizzare tutte le chiavi KMS in loro possesso Account AWS, ma non possono creare o modificare alcuna chiave KMS.

Per visualizzare le chiavi KMS nelle pagine Chiavi gestite da AWS e nelle pagine delle chiavi gestite dal cliente, i principali richiedono le GetResources autorizzazioni [kms: ListKeys](#), [kms: e tag: ListAliases](#), anche se le chiavi non hanno tag o alias. Le autorizzazioni rimanenti, in particolare [kms: DescribeKey](#), sono necessarie per visualizzare le colonne e i dati opzionali della tabella delle chiavi KMS nelle pagine di dettaglio delle chiavi KMS. Le ListRoles autorizzazioni [iam: ListUsers](#) e [iam: ListGroups](#) sono necessarie per visualizzare la politica chiave nella visualizzazione predefinita senza errori. Per visualizzare i dati nella pagina Custom key store e i dettagli sulle chiavi KMS negli archivi chiavi personalizzati, i mandanti necessitano anche dell'autorizzazione [kms: DescribeCustomKeyStores](#).

Se limiti l'accesso alla console di un utente a particolari chiavi KMS, la console visualizza un errore per ogni chiave KMS che non è visibile.

Questa policy include due istruzioni di policy. L'elemento Resource nella prima istruzione di policy consente le autorizzazioni specificate per tutte le chiavi KMS in tutte le regioni dell' Account AWS di esempio. I visualizzatori della console non hanno bisogno di un accesso aggiuntivo perché la console AWS KMS visualizza solo le chiavi KMS nell'account del principale. Questo è vero anche se hanno il permesso di visualizzare le chiavi KMS in altri. Account AWS Le autorizzazioni rimanenti AWS KMS e quelle IAM richiedono un "Resource": "*" elemento perché non si applicano a nessuna chiave KMS particolare.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReadOnlyAccessForAllKMSKeysInAccount",
      "Effect": "Allow",
      "Action": [
        "kms:GetPublicKey",
        "kms:GetKeyRotationStatus",
        "kms:GetKeyPolicy",
        "kms:DescribeKey",
        "kms:ListKeyPolicies",
        "kms:ListResourceTags",
        "tag:GetResources"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    },
    {
      "Sid": "ReadOnlyAccessForOperationsWithNoKMSKey",
      "Effect": "Allow",
      "Action": [
        "kms:ListKeys",
        "kms:ListAliases",
        "iam:ListRoles",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

Consentire a un utente di creare chiavi KMS

La seguente policy IAM consente a un utente di creare chiavi KMS di tutti i tipi. Il valore dell'`Resource` è `*` dovuto al fatto che l'`CreateKey` operazione non utilizza AWS KMS risorse particolari (chiavi o alias KMS).

[Per limitare l'utente a particolari tipi di chiavi KMS, usa le chiavi di condizione `kms:KeySpec`, `kms:KeyUsage`, `kms:KeyOrigin`.](#)

JSON

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

I principali che creano le chiavi potrebbero richiedere alcune autorizzazioni correlate.

- `kms:PutKeyPolicy` — I responsabili che dispongono dell'`kms:CreateKey` autorizzazione possono impostare la politica chiave iniziale per la chiave KMS. Tuttavia, il `CreateKey` chiamante deve disporre di [kms:PutKeyPolicy](#) permission, che gli consente di modificare la politica della chiave KMS, oppure deve specificare il `BypassPolicyLockoutSafetyCheck` parametro di `CreateKey`, che non è consigliato. Il chiamante `CreateKey` può ottenere l'autorizzazione `kms:PutKeyPolicy` per la chiave KMS da una policy IAM o può includere questa autorizzazione nella policy chiavi della chiave KMS che sta creando.
- `kms:TagResource` — Per aggiungere tag alla chiave KMS durante l'`CreateKey` operazione, il `CreateKey` chiamante deve disporre dell'autorizzazione [kms:TagResource](#) in una policy IAM. Includere questa autorizzazione nella policy delle chiavi della nuova chiave KMS non è sufficiente. Tuttavia, se il chiamante `CreateKey` include `kms:TagResource` nella policy delle chiavi iniziale, può aggiungere i tag in una chiamata separata dopo la creazione della chiave KMS.
- `kms:CreateAlias` — I principali che creano una chiave KMS nella AWS KMS console devono avere l'`CreateAlias` autorizzazione [kms:sulla chiave KMS](#) e sull'alias. La console effettua due chiamate, una a `CreateKey` e una a `CreateAlias`. È necessario fornire l'autorizzazione per l'alias in una

policy IAM. Puoi fornire l'autorizzazione chiave KMS in una policy delle chiavi o in una policy IAM. Per informazioni dettagliate, vedi [Controllo dell'accesso agli alias](#).

Inoltre `kms:CreateKey`, la seguente policy IAM fornisce l'`kms:TagResource` autorizzazione su tutte le chiavi KMS presenti nell'account Account AWS e l'autorizzazione su tutti gli alias dell'account `kms:CreateAlias`. Include anche alcune utili autorizzazioni di sola lettura che possono essere fornite solo in una policy IAM.

Questa policy IAM non include l'autorizzazione `kms:PutKeyPolicy` o altre autorizzazioni che possono essere impostate in una policy delle chiavi. Come [best practice](#), è consigliabile impostare queste autorizzazioni nella policy delle chiavi che si applica esclusivamente a una chiave KMS.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPermissionsForParticularKMSKeys",
      "Effect": "Allow",
      "Action": "kms:TagResource",
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    },
    {
      "Sid": "IAMPermissionsForParticularAliases",
      "Effect": "Allow",
      "Action": "kms:CreateAlias",
      "Resource": "arn:aws:kms:*:111122223333:alias/*"
    },
    {
      "Sid": "IAMPermissionsForAllKMSKeys",
      "Effect": "Allow",
      "Action": [
        "kms:CreateKey",
        "kms:ListKeys",
        "kms:ListAliases"
      ],
      "Resource": "*"
    }
  ]
}
```



```
}
```

Consenti a un utente di crittografare e decrittografare con qualsiasi chiave KMS in una determinata chiave Account AWS

La seguente politica IAM consente a un utente di crittografare e decrittografare i dati con qualsiasi chiave KMS in 111122223333. Account AWS

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    }
  ]
}
```

Consenti a un utente di crittografare e decrittografare con qualsiasi chiave KMS in una regione e in una regione specifiche Account AWS

La seguente politica IAM consente a un utente di crittografare e decrittografare i dati con qualsiasi chiave KMS Account AWS 111122223333 nella regione Stati Uniti occidentali (Oregon).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt"
      ],

```

```
"Resource": [  
  "arn:aws:kms:us-west-2:111122223333:key/*"  
]  
}
```

Consentire a un utente di eseguire la crittografia e la decrittazione con chiavi KMS specifiche

La policy IAM seguente consente a un utente di crittografare e decrittare i dati con le due chiavi KMS specificate nell'elemento Resource. Per specificare una chiave KMS in un'istruzione della policy IAM, devi utilizzare l'[ARN chiave](#) della chiave KMS.

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "kms:Encrypt",  
        "kms:Decrypt"  
      ],  
      "Resource": [  
        "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
        "arn:aws:kms:us-west-2:111122223333:key/01234abc-d12a-b34c-d56e-f1234567890a"  
      ]  
    }  
  ]  
}
```

Impedire a un utente di disabilitare o eliminare le chiavi KMS

La policy IAM seguente impedisce a un utente di disabilitare o eliminare le chiavi KMS, anche quando un'altra policy IAM o una policy delle chiavi lo consente. Una policy che nega autorizzazioni in modo esplicito sostituisce tutte le altre policy, anche quelle che concedono le stesse autorizzazioni. Per ulteriori informazioni, consulta [Risoluzione dei problemi relativi alle autorizzazioni AWS KMS](#).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": [
      "kms:DisableKey",
      "kms:ScheduleKeyDeletion"
    ],
    "Resource": "*"
  }
}
```

Politiche di controllo delle risorse in AWS KMS

Le politiche di controllo delle risorse (RCPs) sono un tipo di politica organizzativa che puoi utilizzare per applicare controlli preventivi sulle AWS risorse dell'organizzazione. RCPs ti aiutano a limitare centralmente l'accesso esterno alle tue AWS risorse su larga scala. RCPs integrano le politiche di controllo dei servizi (SCP). SCPs Può essere utilizzato per impostare centralmente le autorizzazioni massime per i ruoli e gli utenti IAM dell'organizzazione, ma RCPs può essere utilizzato per impostare centralmente le autorizzazioni massime per AWS le risorse dell'organizzazione.

Puoi utilizzarle RCPs per gestire le autorizzazioni relative alle chiavi KMS gestite dal cliente nella tua organizzazione. RCPs da soli non sono sufficienti a concedere le autorizzazioni alle chiavi gestite dai clienti. Nessun permesso viene concesso da un RCP. Un RCP definisce una barriera di autorizzazioni, o impone dei limiti, alle azioni che le identità possono intraprendere sulle risorse degli account interessati. L'amministratore deve comunque allegare politiche basate sull'identità ai ruoli o agli utenti IAM o politiche chiave per concedere effettivamente le autorizzazioni.

Note

Le politiche di controllo delle risorse dell'organizzazione non si applicano a. [Chiavi gestite da AWS](#)

Chiavi gestite da AWS vengono creati, gestiti e utilizzati per conto dell'utente da un AWS servizio, non è possibile modificare o gestire le relative autorizzazioni.

Ulteriori informazioni

- Per informazioni più generali su RCPs, consulta le [politiche di controllo delle risorse](#) nella Guida per l'AWS Organizations utente.
- Per dettagli su come definire RCPs, inclusi esempi, consultate la [sintassi RCP nella Guida](#) per l'AWS Organizations utente.

L'esempio seguente dimostra come utilizzare un RCP per impedire ai responsabili esterni di accedere alle chiavi gestite dai clienti dell'organizzazione. Questa politica è solo un esempio e dovreste personalizzarla per soddisfare le vostre esigenze aziendali e di sicurezza specifiche. Ad esempio, potresti voler personalizzare la tua politica per consentire l'accesso ai tuoi partner commerciali. Per maggiori dettagli, consulta l'archivio degli [esempi di politiche perimetrali dei dati](#).

Note

L'`kms:RetireGrant` autorizzazione non è valida in un RCP, anche se l'azione elemento specifica un asterisco (*) come jolly.

Per ulteriori informazioni su come `kms:RetireGrant` viene determinata l'autorizzazione a, vedere. [Ritirare e revocare le concessioni](#)

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RCPEnforceIdentityPerimeter",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "kms:*",
      "Resource": "*",
      "Condition": {
        "StringNotEqualsIfExists": {
          "aws:PrincipalOrgID": "my-org-id"
        },
        "Bool": {
          "aws:PrincipalIsAWSService": "false"
        }
      }
    }
  ]
}
```

```

    }
  }
]
}

```

L'esempio seguente RCP richiede che i responsabili del AWS servizio possano accedere alle chiavi KMS gestite dai clienti solo quando la richiesta proviene dall'organizzazione. Questa politica applica il controllo solo alle richieste presenti. `aws:SourceAccount` Ciò garantisce che le integrazioni di servizi che non richiedono l'uso di `aws:SourceAccount` non siano influenzate. Se `aws:SourceAccount` è presente nel contesto della richiesta, la `Null` condizione restituisce un valore pari a `true`, causando l'`aws:SourceOrgID` applicazione della chiave.

Per ulteriori informazioni sul problema del vice confuso, consulta [Il problema del vice confuso](#) nella Guida per l'utente di IAM.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RCPEnforceConfusedDeputyProtection",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "kms:*",
      "Resource": "*",
      "Condition": {
        "StringNotEqualsIfExists": {
          "aws:SourceOrgID": "my-org-id"
        },
        "Bool": {
          "aws:PrincipalIsAWSService": "true"
        },
        "Null": {
          "aws:SourceAccount": "false"
        }
      }
    }
  ]
}

```

}

Sovvenzioni in AWS KMS

Una concessione è uno strumento delle policy che permette ai [principali AWS](#) di usare le chiavi KMS nelle operazioni di crittografia. Può anche consentire loro di visualizzare una chiave KMS (`DescribeKey`) e creare e gestire concessioni. Quando autorizzi l'accesso a una chiave KMS, le concessioni vengono prese in considerazione insieme alle [policy chiave](#) e alle [policy IAM](#). I privilegi vengono spesso utilizzati per le autorizzazioni temporanee perché è possibile crearne una, utilizzarne le autorizzazioni ed eliminarla senza modificare le policy chiave o le policy IAM.

Le sovvenzioni sono comunemente utilizzate dai AWS servizi che si integrano con AWS KMS per crittografare i dati inattivi. Il servizio crea una concessione per conto di un utente nell'account, ne utilizza le autorizzazioni e la revoca non appena l'attività è completata. Per informazioni dettagliate su come i AWS servizi utilizzano le concessioni, consulta l'argomento *Encryption at rest* nella guida per l'utente del servizio o nella guida per sviluppatori.

Le concessioni sono un meccanismo di controllo degli accessi molto flessibile e utile. Quando crei una concessione per una chiave KMS, la concessione consente al principale beneficiario di chiamare le operazioni di concessione specificate nella chiave KMS a condizione che siano soddisfatte tutte le condizioni specificate nella concessione.

- Ogni concessione consente l'accesso a una sola chiave KMS. È possibile creare una concessione per una chiave KMS in un Account AWS diverso.
- Una concessione può consentire l'accesso a una chiave KMS, ma non negare l'accesso.
- Ogni concessione ha un [beneficiario principale](#). Il beneficiario principale può rappresentare una o più identità nella Account AWS stessa chiave KMS o in un account diverso.
- Una concessione può consentire solo [operazioni di concessione](#). Le operazioni di concessione devono essere supportate dalla chiave KMS nella concessione. Se si specifica un'operazione non supportata, la [CreateGrant](#)richiesta ha esito negativo con un'eccezione. `ValidationError`
- Il principale beneficiario può utilizzare le autorizzazioni concesse dalla concessione senza specificarla, proprio come se le autorizzazioni provenissero da una policy chiave o da una policy IAM. Tuttavia, poiché l' AWS KMS API segue un [eventuale modello di coerenza](#), quando si crea, si ritira o si revoca una sovvenzione, potrebbe verificarsi un breve ritardo prima che la modifica sia disponibile per intero. AWS KMS Per utilizzare immediatamente le autorizzazioni in una concessione, [usa un token di concessione](#).

- Un principale autorizzato può eliminare la concessione ([ritirala](#) o [revocala](#)). L'eliminazione di una concessione elimina tutte le autorizzazioni consentite dalla concessione. Non è necessario individuare le policy da aggiungere o rimuovere per annullare la concessione.
- AWS KMS limita il numero di concessioni per ogni chiave KMS. Per informazioni dettagliate, consultare [Concessioni per chiave KMS: 50.000](#).

Prestare attenzione quando si creano concessione e quando si concede ad altri l'autorizzazione di creare concessione. L'autorizzazione a creare sovvenzioni ha implicazioni sulla sicurezza, proprio come concedere il permesso di [kms: PutKeyPolicy](#) autorizzazione a impostare politiche.

- Gli utenti autorizzati a creare concessioni per una chiave KMS (`kms:CreateGrant`) possono utilizzare una concessione per consentire a utenti e ruoli, inclusi i AWS servizi, di utilizzare la chiave KMS. I principali possono essere identità proprie Account AWS o identità appartenenti a un account o a un'organizzazione diversi.
- Le sovvenzioni possono consentire solo un sottoinsieme di operazioni. AWS KMS È possibile utilizzare le concessioni per consentire ai principali di visualizzare la chiave KMS, utilizzarla nelle operazioni di crittografia e creare e ritirare i concessioni. Per informazioni dettagliate, consulta [Operazioni di concessione](#). È possibile utilizzare anche i [vincoli di concessione](#) per limitare le autorizzazioni nell'ambito di una concessione per una chiave crittografica simmetrica.
- Le entità principali possono ottenere l'autorizzazione per creare concessioni da una policy chiave o da una policy IAM. I principali che ottengono l'autorizzazione `kms:CreateGrant` da una policy possono creare concessioni per qualunque [operazione di concessione](#) sulla chiave KMS. Questi principali non sono tenuti ad avere l'autorizzazione che stanno concedendo sulla chiave. Quando si consente un'autorizzazione `kms:CreateGrant` in una policy, è possibile utilizzare le [condizioni della policy](#) per limitare questa autorizzazione.
- I principali possono inoltre ottenere l'autorizzazione per creare concessioni da una concessione. Questi responsabili possono delegare solo le autorizzazioni che sono state loro concesse, anche se dispongono di altre autorizzazioni previste da una politica. Per informazioni dettagliate, consultare [Concessione dell'autorizzazione CreateGrant](#).

Concetti delle concessioni

Per utilizzare le concessioni in modo efficace, è necessario comprendere i termini e i concetti che AWS KMS usa.

Vincoli di concessione

Condizione che limita le autorizzazioni nella concessione. Attualmente, AWS KMS supporta vincoli di concessione basati sul [contesto di crittografia](#) nella richiesta di un'operazione crittografica. Per informazioni dettagliate, consultare [Utilizzo dei vincoli di concessione](#).

ID concessione

L'identificatore univoco di una concessione per una chiave KMS. È possibile utilizzare un ID di concessione, insieme a un [identificatore chiave](#), per identificare una concessione in una richiesta or. [RetireGrantRevokeGrant](#)

Operazioni di concessione

Le AWS KMS operazioni che puoi consentire in una sovvenzione. Se si specificano altre operazioni, la [CreateGrant](#) richiesta ha esito negativo con un `ValidationError` eccezione. Queste sono anche le operazioni che accettano un [token di concessione](#). Per informazioni dettagliate sulla modifica di queste autorizzazioni, consulta [AWS KMS autorizzazioni](#).

Queste operazioni di concessione rappresentano effettivamente l'autorizzazione per l'utilizzo dell'operazione. Pertanto, per l'operazione `ReEncrypt`, puoi specificare `ReEncryptFrom`, `ReEncryptTo` o entrambi `ReEncrypt*`.

Le operazioni di concessione sono:

- Operazioni di crittografia
 - [Decrypt](#)
 - [DeriveSharedSecret](#)
 - [Encrypt](#)
 - [GenerateDataKey](#)
 - [GenerateDataKeyPair](#)
 - [GenerateDataKeyPairWithoutPlaintext](#)
 - [GenerateDataKeyWithoutPlaintext](#)
 - [GenerateMac](#)
 - [ReEncryptFrom](#)
 - [ReEncryptTo](#)
 - [Sign](#)
 - [Verify](#)
 - [VerifyMac](#)

- Altre operazioni
 - [CreateGrant](#)
 - [DescribeKey](#)
 - [GetPublicKey](#)
 - [RetireGrant](#)

Le operazioni di concessione che autorizzi devono essere supportate dalla chiave KMS nella concessione. Se si specifica un'operazione non supportata, la [CreateGrant](#) richiesta ha esito negativo con un `ValidationError` eccezione. Ad esempio, le concessioni per le chiavi KMS di crittografia simmetrica non possono permettere le operazioni [Sign](#), [Verify](#), [GenerateMac](#) o [VerifyMac](#). Le concessioni per le chiavi KMS asimmetriche non possono permettere operazioni che generano chiavi di dati o coppie di chiavi di dati.

Concessione di token

L' AWS KMS API segue un [eventuale modello di coerenza](#). Quando crei una concessione, potrebbe verificarsi un breve ritardo prima che la modifica sia disponibile in AWS KMS. Generalmente, occorrono pochissimi secondi per la propagazione della modifica in tutto il sistema, ma in alcuni casi possono essere necessari diversi minuti. Se provi a utilizzare una concessione prima della propagazione in tutto il sistema, potrebbe verificarsi un errore di accesso negato. Un token di concessione consente di fare riferimento alla concessione e di utilizzare immediatamente le autorizzazioni di concessione.

Un grant token (token di concessione) è una stringa univoca, non segreta, di lunghezza variabile, codificata in base64 che rappresenta una concessione. È possibile utilizzare il token di concessione per identificare la concessione in qualsiasi [operazione di concessione](#). Tuttavia, poiché il valore del token è un hash digest, non rivela alcun dettaglio sulla concessione.

Un token di concessione è progettato in modo da poter essere utilizzato solo quando la concessione si è propagata in AWS KMS. Dopo di che, l'[assegnatario principale](#) può utilizzare l'autorizzazione nella concessione senza fornire un token di concessione o qualsiasi altra prova della concessione. Puoi utilizzare un token di concessione in qualsiasi momento, ma una volta che la concessione è alla fine coerente, AWS KMS utilizza la concessione per determinare le autorizzazioni, non il token di concessione.

Ad esempio, il comando seguente richiama l'[GenerateDataKey](#) operazione. Utilizza un token di concessione per rappresentare la concessione che dà al chiamante (l'assegnatario principale) il permesso di chiamare `GenerateDataKey` sulla chiave KMS specificata.

```
$ aws kms generate-data-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --key-spec AES_256 \  
  --grant-token $token
```

Puoi anche utilizzare un token di concessione per identificare una concessione in qualsiasi operazione che gestisce le concessioni. Ad esempio, il [preside uscente](#) può utilizzare un token di concessione in una chiamata all'[RetireGrant](#) operazione.

```
$ aws kms retire-grant \  
  --grant-token $token
```

CreateGrant è l'unica operazione che restituisce un token di concessione. Non è possibile ottenere un token di concessione da nessun'altra AWS KMS operazione o dall'[evento di CloudTrail registro](#) relativo all' CreateGrant operazione. Le [ListRetirableGrants](#) operazioni [ListGrants](#) and restituiscono l'[ID della concessione](#), ma non un token di concessione.

Per informazioni dettagliate, consultare [Utilizzo di un token di concessione](#).

Principale assegnatario

Le identità che ottengono le autorizzazioni specificate nella concessione. Ogni concessione ha un principale beneficiario che può rappresentare più identità.

Il principale beneficiario può essere qualsiasi AWS principale, incluso un Account AWS (root), un [utente IAM](#), un [ruolo IAM](#), un [ruolo](#) o utente [federato o un utente con ruolo](#) presunto. L'assegnatario principale può essere nello stesso account della chiave KMS o in un account diverso. Tuttavia, l'assegnatario principale della concessione non può essere un [principale del servizio](#), un [gruppo IAM](#), o un'[organizzazione AWS](#).

Note

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

Ritiro (di una concessione)

Termina una concessione. Ritiri una concessione al termine dell'utilizzo delle autorizzazioni.

Sia la revoca che il ritiro di una concessione eliminano la concessione. Ma il ritiro può essere fatto da un principale specificato nella concessione. La revoca viene in genere eseguita da un amministratore della chiave. Per informazioni dettagliate, consultare [Ritirare e revocare le concessioni](#).

Principale per il ritiro

Un principale che può [ritirare una concessione](#). È possibile specificare un principale per il ritiro in una concessione, ma non è obbligatorio. Il principale uscente può essere qualsiasi AWS principale, inclusi utenti IAM Account AWS, ruoli IAM, utenti federati e utenti assunti. Il principale per il ritiro può essere nello stesso account della chiave KMS o in un account diverso.

Note

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

Oltre al capitale in pensione specificato nella sovvenzione, una sovvenzione può essere ritirata dalla società Account AWS in cui è stata creata. Se la concessione consente l'operazione `RetireGrant`, l'[assegnatario principale](#) può ritirare la concessione. Inoltre, l'Account AWS ente Account AWS preposto al pensionamento può delegare l'autorizzazione a ritirare una sovvenzione a un responsabile IAM dello stesso Account AWS. Per informazioni dettagliate, consultare [Ritirare e revocare le concessioni](#).

Revoca (di una concessione)

Termina una concessione. Revochi una concessione per negare attivamente le autorizzazioni consentite dalla concessione.

Sia la revoca che il ritiro di una concessione eliminano la concessione. Ma il ritiro può essere fatto da un principale specificato nella concessione. La revoca viene in genere eseguita da un amministratore della chiave. Per informazioni dettagliate, consultare [Ritirare e revocare le concessioni](#).

Consistenza finale (per le concessioni)

L'AWS KMS API segue un [eventuale](#) modello di coerenza. Quando crei, ritiri o revochi una concessione, potrebbe verificarsi un breve ritardo prima che la modifica sia disponibile in AWS.

KMS. Generalmente, occorrono pochissimi secondi per la propagazione della modifica in tutto il sistema, ma in alcuni casi possono essere necessari diversi minuti.

Potresti apprendere di questo breve ritardo se ricevi errori imprevisti. Ad esempio, se si tenta di gestire una nuova concessione o di utilizzare le autorizzazioni di una nuova concessione prima che la concessione sia nota per intero AWS KMS, è possibile che venga visualizzato un errore di accesso negato. Se ritiri o revochi una concessione, l'assegnatario principale potrebbe ancora essere in grado di utilizzare le autorizzazioni per un breve periodo fino a quando la concessione non viene completamente eliminata. La strategia tipica consiste nel riprovare la richiesta e alcune AWS SDKs includono il backoff automatico e la logica dei tentativi.

AWS KMS dispone di funzionalità per mitigare questo breve ritardo.

- Per utilizzare immediatamente le autorizzazioni in una nuova concessione, usa un [token di concessione](#). È possibile utilizzare un token di concessione per fare riferimento a una concessione in qualsiasi [operazione di concessione](#). Per istruzioni, consultare [Utilizzo di un token di concessione](#).
- L'[CreateGrant](#) operazione ha un Name parametro che impedisce che le operazioni di nuovo tentativo creino concessioni duplicate.

Note

I token di concessione sostituiscono la validità della concessione fino a quando tutti gli endpoint del servizio non sono stati aggiornati con il nuovo stato della concessione. Nella maggior parte dei casi, la consistenza finale sarà raggiunta entro cinque minuti.

Per ulteriori informazioni, consulta [Consistenza finale di AWS KMS](#).

Le migliori pratiche per le sovvenzioni AWS KMS

AWS KMS consiglia le seguenti best practice per la creazione, l'utilizzo e la gestione delle sovvenzioni.

- Limita le autorizzazioni nella concessione a quelle richieste dall'assegnatario principale. Utilizzare il principio di [accesso meno privilegiato](#).
- Utilizza un assegnatario principale specifico, ad esempio un ruolo IAM, e concedigli l'autorizzazione di utilizzare solo le operazioni API richieste.

- Usa il contesto di crittografia [Vincoli di concessione](#) per garantire che i chiamanti utilizzino la chiave KMS per lo scopo previsto. Per informazioni dettagliate su come utilizzare il contesto di crittografia in una richiesta di protezione dei dati, consulta [Come proteggere l'integrità dei dati crittografati utilizzando AWS Key Management Service e EncryptionContext](#) nel blog sulla AWS sicurezza.

Tip

Utilizza il vincolo di [EncryptionContextEqual](#) concessione ogni volta che è possibile. Il vincolo di [EncryptionContextSubset](#) concessione è più difficile da usare correttamente. Se devi utilizzarlo, leggi attentamente la documentazione e testa il vincolo di concessione per assicurarti che funzioni come previsto.

- Eliminare concessioni duplicate. Le concessioni duplicate hanno lo stesso ARN chiave, le stesse azioni API, lo stesso assegnatario principale, lo stesso contesto di crittografia e lo stesso nome. Se ritiri o revochi la concessione originale ma lasci i duplicati, le concessioni duplicate rimanenti rappresentano escalation involontarie di privilegi. Per evitare di duplicare le concessioni quando riprovi una richiesta `CreateGrant`, utilizza il [parametro Name](#). Per rilevare sovvenzioni duplicate, utilizza l'operazione [ListGrants](#). Se crei accidentalmente una concessione duplicata, ritirla o revoca la prima possibile.

Note

Le concessioni per [chiavi gestite da AWS](#) potrebbero sembrare duplicate ma avere diversi assegnatari principali.

Il campo `GranteePrincipal` nella risposta `ListGrants` contiene solitamente il principal dell'assegnatario della concessione. Tuttavia, quando il beneficiario principale della sovvenzione è un AWS servizio, il `GranteePrincipal` campo contiene il [principale](#) del servizio, che potrebbe rappresentare diversi committenti del beneficiario.

- Ricorda che le concessioni non scadono automaticamente. [Ritira o revoca la concessione](#) non appena l'autorizzazione non sarà più necessaria. Le concessioni che non vengono eliminate potrebbero creare un rischio per la sicurezza delle risorse crittografate.

Controllo dell'accesso alle concessioni

È possibile controllare l'accesso alle operazioni che creano e gestiscono concessioni nelle policy chiave, nelle policy IAM e nelle concessioni. I principali che ottengono l'autorizzazione CreateGrant da una concessione hanno [autorizzazioni di concessione più limitate](#).

Operazione API	Policy chiave o policy IAM	Grant
CreateGrant	✓	✓
ListGrants	✓	-
ListRetirableGrants	✓	-
Ritiro di concessioni	(Limitato. Consulta Ritirare e revocare le concessioni)	✓
RevokeGrant	✓	-

Quando utilizzi una policy chiave o una policy IAM per controllare l'accesso alle operazioni di creazione e gestione delle sovvenzioni, puoi utilizzare una o più delle seguenti condizioni politiche per limitare l'autorizzazione. AWS KMS supporta tutte le seguenti chiavi di condizione relative alle sovvenzioni. Per informazioni dettagliate ed esempi, consulta [AWS KMS tasti di condizione](#).

[km: GrantConstraintType](#)

Consente ai principali di creare una concessione solo quando la concessione include il [vincolo di concessione](#) specificato.

[km: GrantsFor AWSResource](#)

Consente ai principali di chiamare CreateGrant o RevokeGrant solo quando [un AWS servizio integrato con AWS KMS](#) invia la richiesta per conto del principale. ListGrants

[km: GrantOperations](#)

Consente ai principali di creare una concessione, ma limita la concessione alle operazioni specificate.

[km: GranteePrincipal](#)

Consente ai principali di creare una concessione solo per l'[assegnatario principale](#) specificato.

[km: RetiringPrincipal](#)

Consente ai principali di creare una concessione solo quando la concessione specifica un particolare [principale per il ritiro](#).

Creazione di concessioni

Prima di creare una concessione, scopri le opzioni per la personalizzazione della concessione. È possibile utilizzare vincoli di concessione per limitare le autorizzazioni nella concessione. Scopri di più sulla concessione dell'autorizzazione `CreateGrant`. Le entità principali che ricevono l'autorizzazione per creare concessioni da una concessione sono limitate nelle concessioni che possono creare.

Argomenti

- [Creazione di una concessione](#)
- [Concessione dell'autorizzazione `CreateGrant`](#)

Creazione di una concessione

Per creare una concessione, chiama l'[CreateGrant](#) operazione. Specifica una chiave KMS, un [assegnatario principale](#) e un elenco di [operazioni di concessione](#) consentite. È inoltre possibile designare un [principale per il ritiro](#) opzionale. Per personalizzare la concessione, puoi usare i parametri `Constraints` opzionali per definire i [vincoli di concessione](#).

Quando si crea, si ritira o si revoca una concessione, potrebbe verificarsi un breve ritardo, in genere meno di cinque minuti, prima che la modifica sia disponibile in AWS KMS. Per ulteriori informazioni, consulta [Consistenza finale \(per concessioni\)](#).

Ad esempio, il comando `CreateGrant` seguente crea una concessione che consente agli utenti autorizzati ad assumere il ruolo `keyUserRole` di chiamare l'operazione [Decrypt](#) sulla [chiave KMS simmetrica](#) specificata. La concessione utilizza il parametro `RetiringPrincipal` per designare un'entità principale che può ritirare la concessione. Include anche un vincolo di concessione che consente l'autorizzazione solo quando il [contesto di crittografia](#) nella richiesta include `"Department": "IT"`.

```
$ aws kms create-grant \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \  
  --operations Decrypt \  
  --constraints '{"Department": "IT"}'
```

```
--retiring-principal arn:aws:iam::111122223333:role/adminRole \  
--constraints EncryptionContextSubset={Department=IT}
```

Se il tuo codice tenta di nuovo l'operazione `CreateGrant` o utilizza un [SDK AWS che rinvia automaticamente le richieste](#), utilizza il parametro opzionale [Nome](#) per impedire la creazione di concessioni duplicate. Se AWS KMS riceve una `CreateGrant` richiesta di concessione con le stesse proprietà di una concessione esistente, incluso il nome, riconosce la richiesta come un nuovo tentativo e non crea una nuova concessione. Non puoi utilizzare il valore `Name` per identificare la concessione in qualsiasi operazione AWS KMS.

Important

Non includere informazioni riservate o sensibili nel nome della concessione. Può apparire in testo semplice nei CloudTrail log e in altri output.

```
$ aws kms create-grant \  
  --name IT-1234abcd-keyUserRole-decrypt \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \  
  --operations Decrypt \  
  --retiring-principal arn:aws:iam::111122223333:role/adminRole \  
  --constraints EncryptionContextSubset={Department=IT}
```

Per esempi di codice che dimostrano come creare sovvenzioni in diversi linguaggi di programmazione, vedere. [Utilizzo `CreateGrant` con un AWS SDK o una CLI](#)

Utilizzo dei vincoli di concessione

I [vincoli di concessione](#) stabiliscono condizioni relative alle autorizzazioni che la concessione dà all'assegnatario principale. I vincoli di concessione sostituiscono le [chiavi di condizione](#) in una [policy chiave](#) o in una [policy IAM](#). Ogni valore del vincolo di concessione può includere fino a 8 coppie del contesto di crittografia. Il valore del contesto di crittografia in ogni vincolo di concessione non può superare i 384 caratteri.

Important

Non includere informazioni riservate o sensibili in questo campo. Questo campo può essere visualizzato in testo semplice nei CloudTrail log e in altri output.

AWS KMS supporta due vincoli di concessione `EncryptionContextEquals` ed entrambi stabiliscono `EncryptionContextSubset` i requisiti per il [contesto di crittografia](#) in una richiesta di operazione crittografica.

I vincoli di concessione del contesto di crittografia sono progettati per essere utilizzati con [operazioni di concessione](#) che dispongono di un parametro contesto di crittografia.

- I vincoli di contesto di crittografia sono validi solo in una concessione per una chiave KMS di crittografia simmetrica. Le operazioni di crittografia con altre chiavi KMS non supportano un contesto di crittografia.
- Il vincolo di contesto di crittografia viene ignorato per le operazioni `DescribeKey` e `RetireGrant`. `DescribeKey` e `RetireGrant` non dispongono di un parametro di contesto di crittografia, ma è possibile includere queste operazioni in una concessione con un vincolo di contesto di crittografia.
- È possibile utilizzare un vincolo di contesto di crittografia in una concessione per l'operazione `CreateGrant`. Il vincolo del contesto di crittografia richiede che tutte le concessioni create con l'autorizzazione `CreateGrant` hanno un vincolo di contesto di crittografia altrettanto rigoroso o più rigoroso.

AWS KMS supporta i seguenti vincoli di concessione del contesto di crittografia.

`EncryptionContextEquals`

Utilizza `EncryptionContextEquals` per specificare il contesto di crittografia esatto per le richieste consentite.

`EncryptionContextEquals` richiede che le coppie del contesto di crittografia nella richiesta corrispondano in modo esatto a livello di maiuscole e minuscole alle coppie del contesto di crittografia nel vincolo di concessione. Le coppie possono essere visualizzate in qualsiasi ordine, ma le chiavi e i valori in ciascuna coppia non possono variare.

Ad esempio, se il vincolo di concessione `EncryptionContextEquals` richiede la coppia del contesto di crittografia `"Department": "IT"`, la concessione consente le richieste del tipo specificato solo quando il contesto di crittografia nella richiesta è esattamente `"Department": "IT"`.

EncryptionContextSubset

Utilizza `EncryptionContextSubset` per richiedere che le richieste includano particolari coppie di contesto di crittografia.

`EncryptionContextSubset` richiede che le richieste includano tutte le coppie del contesto di crittografia nel vincolo di concessione (corrispondenti in modo esatto a livello di maiuscole e minuscole), ma la richiesta può avere anche altre coppie di contesto di crittografia. Le coppie possono essere visualizzate in qualsiasi ordine, ma le chiavi e i valori in ciascuna coppia non possono variare.

Ad esempio, se il vincolo di concessione `EncryptionContextSubset` richiede la coppia del contesto di crittografia `Department=IT`, la concessione consente le richieste del tipo specificato quando il contesto di crittografia nella richiesta è `"Department": "IT"`, o include `"Department": "IT"` insieme ad altre coppie di contesto di crittografia, come `"Department": "IT", "Purpose": "Test"`.

Per specificare un vincolo di contesto di crittografia in una concessione per una chiave KMS di crittografia simmetrica, utilizzate il parametro nell'operazione. Constraints [CreateGrant](#) La concessione creata da questo comando concede agli utenti autorizzati ad assumere il ruolo `keyUserRole` l'autorizzazione a chiamare l'operazione API [Decrypt](#). Tuttavia, tale autorizzazione è valida solo quando il contesto di crittografia nella richiesta `Decrypt` è una coppia di contesto di crittografia `"Department": "IT"`.

```
$ aws kms create-grant \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \  
  --operations Decrypt \  
  --retiring-principal arn:aws:iam::111122223333:role/adminRole \  
  --constraints EncryptionContextEquals={Department=IT}
```

La concessione risultante è simile alla seguente. Tieni presente che l'autorizzazione concessa al ruolo `keyUserRole` è valida solo quando la richiesta `Decrypt` usa la stessa coppia del contesto di crittografia specificata nel vincolo di concessione. Per trovare le concessioni su una chiave KMS, usa l'operazione. [ListGrants](#)

```
$ aws kms list-grants --key-id 1234abcd-12ab-34cd-56ef-1234567890ab  
{
```

```

    "Grants": [
      {
        "Name": "",
        "IssuingAccount": "arn:aws:iam::111122223333:root",
        "GrantId":
"abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
        "Operations": [
          "Decrypt"
        ],
        "GranteePrincipal": "arn:aws:iam::111122223333:role/keyUserRole",
        "Constraints": {
          "EncryptionContextEquals": {
            "Department": "IT"
          }
        },
        "CreationDate": 1568565290.0,
        "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "RetiringPrincipal": "arn:aws:iam::111122223333:role/adminRole"
      }
    ]
  }

```

Per soddisfare il vincolo di concessione `EncryptionContextEquals`, il contesto di crittografia nella richiesta per l'operazione `Decrypt` deve essere una coppia `"Department": "IT"`. Una richiesta dall'assegnatario principale come la seguente soddisferebbe il vincolo di concessione `EncryptionContextEquals`.

```

$ aws kms decrypt \
  --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab\
  --ciphertext-blob fileb://encrypted_msg \
  --encryption-context Department=IT

```

Quando il vincolo di concessione è `EncryptionContextSubset`, le coppie del contesto di crittografia nella richiesta devono includere le coppie del contesto di crittografia nel vincolo di concessione, ma la richiesta può includere anche altre coppie di contesto di crittografia. Il seguente vincolo di concessione richiede che una delle coppie di contesto di crittografia nella richiesta sia `"Department": "IT"`.

```

"Constraints": {

```

```
"EncryptionContextSubset": {  
  "Department": "IT"  
}  
}
```

La seguente richiesta dall'assegnatario principale soddisferebbe entrambi i vincoli di concessione `EncryptionContextEqual` e `EncryptionContextSubset` di questo esempio.

```
$ aws kms decrypt \  
  --key-id arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \  
  --ciphertext-blob fileb://encrypted_msg \  
  --encryption-context Department=IT
```

Tuttavia, una richiesta come la seguente da parte dell'assegnatario principale soddisferebbe il vincolo di concessione `EncryptionContextSubset`, ma fallirebbe il vincolo di concessione `EncryptionContextEquals`.

```
$ aws kms decrypt \  
  --key-id arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \  
  --ciphertext-blob fileb://encrypted_msg \  
  --encryption-context Department=IT,Purpose=Test
```

AWS i servizi spesso utilizzano vincoli di contesto di crittografia nelle concessioni che consentono loro di utilizzare le chiavi KMS nel tuo Account AWS. Ad esempio, Amazon DynamoDB utilizza una concessione come la seguente per ottenere l'autorizzazione a utilizzare la [Chiave gestita da AWS](#) per DynamoDB nel tuo account. Il vincolo di concessione `EncryptionContextSubset` in questa concessione rende le autorizzazioni nella concessione valide solo quando il contesto di crittografia nella richiesta include coppie `"tableName": "Services"` e `"subscriberID": "111122223333"`. Questo vincolo di concessione significa che la concessione consente a DynamoDB di utilizzare la chiave KMS specificata solo per una determinata tabella nel tuo Account AWS.

Per ottenere questo risultato, esegui l'[ListGrants](#) operazione su Chiave gestita da AWS per DynamoDB nel tuo account.

```
$ aws kms list-grants --key-id 0987dcba-09fe-87dc-65ba-ab0987654321
```

```
{
  "Grants": [
    {
      "Operations": [
        "Decrypt",
        "Encrypt",
        "GenerateDataKey",
        "ReEncryptFrom",
        "ReEncryptTo",
        "RetireGrant",
        "DescribeKey"
      ],
      "IssuingAccount": "arn:aws:iam::111122223333:root",
      "Constraints": {
        "EncryptionContextSubset": {
          "aws:dynamodb:tableName": "Services",
          "aws:dynamodb:subscriberId": "111122223333"
        }
      },
      "CreationDate": 1518567315.0,
      "KeyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
      "GranteePrincipal": "dynamodb.us-west-2.amazonaws.com",
      "RetiringPrincipal": "dynamodb.us-west-2.amazonaws.com",
      "Name": "8276b9a6-6cf0-46f1-b2f0-7993a7f8c89a",
      "GrantId":
        "1667b97d27cf748cf05b487217dd4179526c949d14fb3903858e25193253fe59"
    }
  ]
}
```

Concessione dell'autorizzazione CreateGrant

Una concessione può includere l'autorizzazione per chiamare l'operazione CreateGrant. Ma quando un [assegnatario principale](#) ottiene l'autorizzazione di chiamare CreateGrant da una concessione, piuttosto che da una policy, tale autorizzazione è limitata.

- L'assegnatario principale può solo creare concessioni che consentono alcune o tutte le operazioni nella concessione primaria.
- I [vincoli di concessione](#) nelle concessioni che creano devono essere almeno altrettanto rigorosi di quelli della concessione primaria.

Queste limitazioni non si applicano ai principali che ottengono l'autorizzazione CreateGrant da una policy, anche se le loro autorizzazioni possono essere limitate dalle [condizioni della policy](#).

Ad esempio, considera una concessione che consente al principale della concessione di chiamare le operazioni GenerateDataKey, Decrypt e CreateGrant. Chiamiamo una concessione che consente all'autorizzazione CreateGrant a una concessione primaria.

```
# The original grant in a ListGrants response.
{
  "Grants": [
    {
      "KeyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "CreationDate": 1572216195.0,
      "GrantId": "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
      "Operations": [
        "GenerateDataKey",
        "Decrypt",
        "CreateGrant"
      ],
      "RetiringPrincipal": "arn:aws:iam::111122223333:role/adminRole",
      "Name": "",
      "IssuingAccount": "arn:aws:iam::111122223333:root",
      "GranteePrincipal": "arn:aws:iam::111122223333:role/keyUserRole",
      "Constraints": {
        "EncryptionContextSubset": {
          "Department": "IT"
        }
      },
    }
  ]
}
```

L'assegnatario principale, `exampleUser`, può utilizzare questa autorizzazione per creare una concessione che include un sottoinsieme delle operazioni specificate nella concessione originale, ad esempio CreateGrant e Decrypt. La concessione secondaria non può includere altre operazioni, come ScheduleKeyDeletion o ReEncrypt.

Inoltre, i [vincoli nelle concessioni](#) secondarie devono essere altrettanto o più restrittivi di quelli della concessione primaria. Ad esempio, la concessione figlio può aggiungere coppie a un vincolo EncryptionContextSubset nella concessione padre, ma non può rimuoverle. La

concessione figlio può modificare un vincolo `EncryptionContextSubset` in un vincolo `EncryptionContextEquals`, ma non viceversa.

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

Ad esempio, l'assegnatario principale della concessione può utilizzare l'autorizzazione `CreateGrant` che ha ottenuto dalla concessione primaria per creare la seguente concessione secondaria. Le operazioni nella concessione secondaria sono un sottoinsieme di operazioni della concessione primaria e i vincoli di concessione sono più restrittivi.

```
# The child grant in a ListGrants response.
{
  "Grants": [
    {
      "KeyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "CreationDate": 1572249600.0,
      "GrantId": "fedcba9999c1e2e9876abcde6e9d6c9b6a1987650000abcee009abcdef40183f",
      "Operations": [
        "CreateGrant"
        "Decrypt"
      ]
      "RetiringPrincipal": "arn:aws:iam::111122223333:user/exampleUser",
      "Name": "",
      "IssuingAccount": "arn:aws:iam::111122223333:root",
      "GranteePrincipal": "arn:aws:iam::111122223333:user/anotherUser",
      "Constraints": {
        "EncryptionContextEquals": {
          "Department": "IT"
        }
      },
    },
  ]
}
```

L'assegnatario principale nella concessione secondaria, `anotherUser`, può utilizzare la sua autorizzazione `CreateGrant` per creare concessioni. Tuttavia, le concessioni che `anotherUser` crea devono includere le operazioni nella sua concessione primaria o in un sottoinsieme e i vincoli di concessione devono essere uguali o più severi.

Visualizzazione di concessioni

Per visualizzare la concessione, usa l'[ListGrants](#) operazione. È necessario specificare la chiave KMS a cui si applicano le concessioni. È inoltre possibile filtrare l'elenco delle concessioni in base all'ID concessione o all'assegnatario principale. Per ulteriori esempi, consulta [Utilizzo ListGrants con un AWS SDK o una CLI](#).

Per visualizzare tutte le sovvenzioni nella regione Account AWS e con un particolare [capitale uscente, usa](#). [ListRetirableGrants](#) Le risposte includono dettagli su ogni concessione.

Note

Il campo `GranteePrincipal` nella risposta `ListGrants` contiene solitamente il principal dell'assegnatario della concessione. Tuttavia, quando il beneficiario principale della sovvenzione è un AWS servizio, il `GranteePrincipal` campo contiene il [principale](#) del servizio, che potrebbe rappresentare diversi committenti del beneficiario.

Ad esempio, il comando seguente elenca tutte le concessioni per una chiave KMS.

```
$ aws kms list-grants --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
  "Grants": [
    {
      "KeyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "CreationDate": 1572216195.0,
      "GrantId": "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
      "Constraints": {
        "EncryptionContextSubset": {
          "Department": "IT"
        }
      },
      "RetiringPrincipal": "arn:aws:iam::111122223333:role/adminRole",
      "Name": "",
      "IssuingAccount": "arn:aws:iam::111122223333:root",
      "GranteePrincipal": "arn:aws:iam::111122223333:user/exampleUser",
      "Operations": [
        "Decrypt"
      ]
    }
  ]
}
```



```
}  
]  
}
```

Utilizzo di un token di concessione

[L' AWS KMS API segue un eventuale modello di coerenza.](#) Quando si crea una concessione, la concessione potrebbe non essere effettiva immediatamente. Potrebbe verificarsi un breve ritardo prima che la modifica sia disponibile in AWS KMS. Generalmente, occorrono pochissimi secondi per la propagazione della modifica in tutto il sistema, ma in alcuni casi possono essere necessari diversi minuti. Una volta che la concessione è stata propagata in tutto il sistema, il principale assegnatario può utilizzare le autorizzazioni nella concessione senza specificare il token di concessione o una prova della concessione. Tuttavia, se una concessione è così nuova da non essere ancora nota a tutti AWS KMS, la richiesta potrebbe fallire con un `AccessDeniedException` errore.

Per utilizzare immediatamente le autorizzazioni in una nuova concessione, utilizza il [token di concessione](#) per la concessione. Salva il token di concessione restituito dall'[CreateGrant](#) operazione. Quindi invia il token di concessione nella richiesta per l' AWS KMS operazione. Puoi inviare un token di concessione a qualsiasi [operazione di AWS KMS concessione](#) e puoi inviare più token di concessione nella stessa richiesta.

L'esempio seguente utilizza l'`CreateGrant` operazione per creare una concessione che consenta le operazioni [GenerateDataKey](#) e [Decrypt](#). Salva il token di concessione che `CreateGrant` restituisce nella variabile `token`. Quindi, in una chiamata all'operazione `GenerateDataKey`, utilizza il token di concessione nella variabile `token`.

```
# Create a grant; save the grant token  
$ token=$(aws kms create-grant \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --grantee-principal arn:aws:iam::111122223333:user/appUser \  
  --retiring-principal arn:aws:iam::111122223333:user/acctAdmin \  
  --operations GenerateDataKey Decrypt \  
  --query GrantToken \  
  --output text)  
  
# Use the grant token in a request  
$ aws kms generate-data-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --key-spec AES_256 \  
  --grant-token $token
```

```
--grant-tokens $token
```

I responsabili con autorizzazione possono anche utilizzare un token di concessione per ritirare una nuova sovvenzione anche prima che la sovvenzione sia disponibile per tutta la durata. AWS KMS(L'operazione `RevokeGrant` non accetta un token di concessione.) Per informazioni dettagliate, consultare [Ritirare e revocare le concessioni](#).

```
# Retire the grant
$ aws kms retire-grant --grant-token $token
```

Ritirare e revocare le concessioni

Per eliminare una concessione, ritirarla o revocarla.

Le [RevokeGrant](#) operazioni [RetireGrant](#) sono molto simili tra loro. Entrambe le operazioni eliminano una concessione, eliminando le autorizzazioni consentite dalla concessione. La differenza principale tra queste operazioni è il modo in cui sono autorizzate.

RevokeGrant

Come la maggior parte AWS KMS delle operazioni, l'accesso all'`RevokeGrant` operazione è controllato tramite [policy chiave](#) e [policy IAM](#). L'`RevokeGrant` API può essere richiamata da qualsiasi preside con `kms:RevokeGrant` autorizzazione. Questa autorizzazione è inclusa nelle autorizzazioni standard concesse agli amministratori delle chiavi. In genere, gli amministratori revocano una concessione per negare le autorizzazioni consentite dalla concessione.

RetireGrant

La concessione determina chi può ritirarla. Questa struttura consente di controllare il ciclo di vita di una concessione senza modificare le policy chiave o le policy IAM. In genere, si ritira una concessione quando si è terminato di utilizzare le relative autorizzazioni.

Una concessione può essere ritirata da un [principale per il ritiro](#) opzionale specificato nella concessione. L'[assegnatario principale](#) può anche ritirare la concessione, ma solo se è anche un principale per il ritiro o se la concessione include l'operazione `RetireGrant`. Come backup, la sovvenzione Account AWS in cui è stata creata la sovvenzione può ritirarla.

C'è un'autorizzazione `kms:RetireGrant` che può essere utilizzata nelle policy IAM, ma dispone di un'utilità limitata. I principali specificati nella concessione possono ritirare una concessione

senza l'autorizzazione `kms:RetireGrant`. L'autorizzazione `kms:RetireGrant` da sola non consente ai principali di ritirare una concessione. L'autorizzazione `kms:RetireGrant` non è valida in una [politica chiave o in una politica](#) di [controllo delle risorse](#).

- Per negare l'autorizzazione a ritirare una sovvenzione, puoi utilizzare un'azione di negazione con l'autorizzazione `kms:RetireGrant` nelle tue politiche IAM.
- Il Account AWS proprietario della chiave KMS può delegare l'autorizzazione `kms:RetireGrant` a un responsabile IAM dell'account.
- Se l'ente preposto al ritiro è diverso Account AWS, gli amministratori dell'altro account possono delegare l'autorizzazione `kms:RetireGrant` a ritirare la concessione a un responsabile IAM appartenente a quell'account.

[L' AWS KMS API segue un eventuale modello di coerenza](#). Quando crei, ritiri o revochi una concessione, potrebbe verificarsi un breve ritardo prima che la modifica sia disponibile in AWS KMS. Generalmente, occorrono pochissimi secondi per la propagazione della modifica in tutto il sistema, ma in alcuni casi possono essere necessari diversi minuti. Se devi eliminare immediatamente una nuova sovvenzione, prima che sia disponibile per tutta la durata AWS KMS, [utilizza un token di concessione](#) per ritirare la sovvenzione. Non è possibile utilizzare un token di concessione per revocare una concessione.

Chiavi di condizione per AWS KMS

È possibile specificare le condizioni nelle [politiche chiave e nelle politiche IAM](#) che controllano l'accesso alle AWS KMS risorse. L'istruzione di policy diventa effettiva solo quando le condizioni sono true. Ad esempio, potresti decidere che un'istruzione di una policy diventi effettiva solo dopo una data specifica. In alternativa, è possibile impostare un'istruzione di policy per controllare l'accesso solo quando viene visualizzato un valore specifico in una richiesta API.

Per specificare le condizioni, puoi utilizzare le chiavi di condizione nell'[elemento Condition](#) di un'istruzione di policy con gli [operatori della condizione IAM](#). Alcune chiavi di condizione si applicano in generale a AWS, altre sono specifiche a AWS KMS.

I valori delle chiavi condizionali devono rispettare i caratteri e le regole di codifica delle policy AWS KMS chiave e delle policy IAM. Per maggiori informazioni sulle regole delineate nel documento delle policy delle chiavi, consulta la sezione [Formato della policy della chiave](#). Per informazioni dettagliate sulle regole delineate nel documento delle policy IAM, consulta la sezione [Requisiti del nome IAM](#) nella Guida per l'utente di IAM.

Argomenti

- [AWS chiavi di condizione globali](#)
- [AWS KMS tasti di condizione](#)
- [AWS KMS chiavi di condizione per piattaforme attestate](#)

AWS chiavi di condizione globali

AWS definisce [le chiavi di condizione globali](#), un insieme di chiavi di condizioni politiche per tutti i AWS servizi che utilizzano IAM per il controllo degli accessi. AWS KMS supporta tutte le chiavi di condizione globali. È possibile utilizzarli nelle politiche AWS KMS chiave e nelle politiche IAM.

Ad esempio, puoi utilizzare la chiave [aws: PrincipalArn](#) global condition per consentire l'accesso a una AWS KMS key (chiave KMS) solo quando il principale nella richiesta è rappresentato dall'Amazon Resource Name (ARN) nel valore della chiave di condizione. Per supportare il [controllo degli accessi basato sugli attributi](#) (ABAC) in AWS KMS, puoi utilizzare la chiave di condizione globale [aws:ResourceTag/tag-key](#) in una policy IAM per consentire l'accesso alle chiavi KMS con un tag particolare.

Per evitare che un AWS servizio venga utilizzato come sostituto confuso in una policy in cui il principale è un responsabile del [AWS servizio, puoi utilizzare le chiavi di condizione](#) o global condition. [aws:SourceArnaws:SourceAccount](#) Per informazioni dettagliate, vedi [Utilizzo delle chiavi di condizione aws:SourceArn o aws:SourceAccount](#).

Per informazioni sulle chiavi di condizione AWS globali, inclusi i tipi di richieste in cui sono disponibili, consulta [AWS Global Condition Context Keys](#) nella IAM User Guide. Per esempi di utilizzo delle chiavi di condizione globali nelle policy IAM, consulta [Controllo dell'accesso alle richieste](#) e [Controllo delle chiavi di tag](#) nella Guida per l'utente di IAM.

Negli argomenti seguenti vengono fornite linee guida speciali per l'utilizzo delle chiavi di condizione basate su indirizzi IP e endpoint VPC.

Argomenti

- [Utilizzo della condizione con indirizzo IP nelle policy con autorizzazioni AWS KMS](#)
- [Utilizzo delle condizioni di endpoint VPC nelle policy con autorizzazioni AWS KMS](#)
- [Utilizzo IPv6 degli indirizzi in IAM e delle politiche chiave AWS KMS](#)

Utilizzo della condizione con indirizzo IP nelle policy con autorizzazioni AWS KMS

Puoi utilizzarli AWS KMS per proteggere i tuoi dati in un [AWS servizio integrato](#). Tuttavia, fai attenzione quando specifichi [gli operatori di condizione dell'indirizzo IP](#) o la chiave di `aws:SourceIp` condizione nella stessa dichiarazione di politica che consente o nega l'accesso. AWS KMS Ad esempio, la policy in [AWS: Denies Access to AWS Based on the Source IP](#) limita AWS le azioni alle richieste provenienti dall'intervallo IP specificato.

Considera questo scenario:

1. Alleghi una policy come quella mostrata in [AWS: Denies Access to AWS Based on the Source IP](#) a un'identità IAM. Imposti il valore della chiave di condizione `aws:SourceIp` sull'intervallo di indirizzi IP per l'azienda dell'utente. A questa identità IAM sono associate altre policy che le consentono di utilizzare Amazon EBS EC2, Amazon e AWS KMS.
2. L'identità tenta di collegare un volume EBS crittografato a un' EC2 istanza. Questa operazione ha esito negativo con un errore di autorizzazione anche se l'utente ha l'autorizzazione a utilizzare tutti i servizi rilevanti.

La fase 2 non riesce perché la richiesta AWS KMS di decrittografia della chiave dati crittografata del volume proviene da un indirizzo IP associato all'infrastruttura Amazon EC2 . Per avere successo, la richiesta deve provenire dall'indirizzo IP dell'utente di origine. Poiché la politica della fase 1 nega esplicitamente tutte le richieste provenienti da indirizzi IP diversi da quelli specificati, ad Amazon EC2 viene negata l'autorizzazione a decrittografare la chiave dati crittografata del volume EBS.

Inoltre, la chiave di condizione `aws:SourceIP` non è efficace quando la richiesta proviene da un [endpoint Amazon VPC](#). Per limitare le richieste a un endpoint VPC, incluso un [endpoint VPC AWS KMS](#), utilizza `aws:SourceVpce` o le chiavi di condizione `aws:SourceVpc`. Per maggiori informazioni, consulta [Endpoint VPC - Controllo dell'uso degli endpoint](#) nella Guida per l'utente di Amazon VPC.

Utilizzo delle condizioni di endpoint VPC nelle policy con autorizzazioni AWS KMS

[AWS KMS supporta gli endpoint Amazon Virtual Private Cloud \(Amazon VPC\) alimentati da. AWS PrivateLink](#) Puoi utilizzare le seguenti [chiavi di condizione globale](#) nelle policy chiave e nelle policy IAM per controllare l'accesso alle AWS KMS risorse quando la richiesta proviene da un VPC o utilizza un endpoint VPC. Per informazioni dettagliate, vedi [Usa gli endpoint VPC per controllare l'accesso alle risorse AWS KMS](#).

- `aws:SourceVpc` limita l'accesso alle richieste dal VPC specificato.

- `aws:SourceVpce` limita l'accesso alle richieste dall'endpoint VPC specificato.

Se utilizzi queste chiavi condizionali per controllare l'accesso alle chiavi KMS, potresti inavvertitamente negare l'accesso ai servizi che utilizzi per AWS tuo conto. AWS KMS

Fai attenzione a evitare una situazione come quella illustrata nell'esempio delle [chiavi di condizione con indirizzo IP](#). Se limiti le richieste di una chiave KMS a un endpoint VPC o VPC, le chiamate AWS KMS da un servizio integrato, come Amazon S3 o Amazon EBS, potrebbero non riuscire. Questo può accadere anche se la richiesta dell'origine sostanzialmente proviene dal VPC o dall'endpoint VPC.

Utilizzo IPv6 degli indirizzi in IAM e delle politiche chiave AWS KMS

Prima di provare ad accedere IPv6, assicurati che tutte le chiavi e le politiche IAM contenenti restrizioni AWS KMS sugli indirizzi IP siano aggiornate per includere gli intervalli di IPv6 indirizzi. Le policy basate su IP che non vengono aggiornate per gestire IPv6 gli indirizzi possono far sì che i client perdano o ottengano erroneamente l'accesso quando iniziano a IPv6 utilizzarli. Per indicazioni generali sui controlli di accesso KMS, consulta [Accesso e autorizzazioni alle chiavi KMS](#). Per maggiori informazioni su KMS e il supporto dual stack, consulta [Supporto per endpoint dual-stack](#).

Important

Queste dichiarazioni non consentono alcuna azione. Utilizza queste istruzioni in combinazione con altre istruzioni che consentono azioni specifiche.

La seguente dichiarazione nega esplicitamente l'accesso a tutte le autorizzazioni KMS per le richieste provenienti dall'intervallo di indirizzi. `192.0.2.* IPv4` A tutti gli indirizzi IP al di fuori di questo intervallo non vengono negate esplicitamente le autorizzazioni KMS. Poiché tutti IPv6 gli indirizzi non rientrano nell'intervallo negato, questa dichiarazione non nega esplicitamente le autorizzazioni KMS per alcun indirizzo. IPv6

```
{
  "Sid": "DenyKMSPermissions",
  "Effect": "Deny",
  "Action": [
    "kms:*"
  ],
```

```

    "Resource": "*",
    "Condition": {
      "NotIpAddress": {
        "aws:SourceIp": [
          "192.0.2.0/24"
        ]
      }
    }
  }
}

```

È possibile modificare l'Conditionelemento per negare sia gli intervalli di indirizzi IPv4 (192.0.2.0/24) che IPv6 (2001:db8:1234::/32), come mostrato nell'esempio seguente.

```

{
  "Sid": "DenyKMSPermissions",
  "Effect": "Deny",
  "Action": [
    "kms:*"
  ],
  "Resource": "*",
  "Condition": {
    "NotIpAddress": {
      "aws:SourceIp": [
        "192.0.2.0/24",
        "2001:db8:1234::/32"
      ]
    }
  }
}

```

AWS KMS tasti di condizione

AWS KMS fornisce un set di chiavi di condizione che è possibile utilizzare nelle politiche chiave e nelle politiche IAM. Queste chiavi di condizione sono specifiche per AWS KMS. Ad esempio, puoi utilizzare la chiave di condizione `kms:EncryptionContext:context-key` per richiedere un [contesto di crittografia](#) specifico per controllare l'accesso a una chiave KMS di crittografia simmetrica.

Condizioni per una richiesta di operazione API

Molte chiavi AWS KMS condizionali controllano l'accesso a una chiave KMS in base al valore di un parametro nella richiesta di un' AWS KMS operazione. Ad esempio, puoi utilizzare la chiave [kms:](#)

[KeySpec](#) condition in una policy IAM per consentire l'uso dell'[CreateKey](#) operazione solo quando il valore del KeySpec parametro nella CreateKey richiesta è. RSA_4096

Questo tipo di condizione funziona anche quando il parametro non è presente nella richiesta, ad esempio quando si usa il valore predefinito del parametro. Ad esempio, puoi utilizzare la chiave di condizione [kms:KeySpec](#) per consentire agli utenti di usare l'operazione CreateKey solo quando il valore del parametro KeySpec è SYMMETRIC_DEFAULT, che è il valore predefinito. Questa condizione consente le richieste che hanno il parametro KeySpec con il valore SYMMETRIC_DEFAULT e le richieste che non hanno alcun parametro KeySpec.

Condizioni per le chiavi KMS utilizzate nelle operazioni API

Alcune chiavi AWS KMS condizionali possono controllare l'accesso alle operazioni in base a una proprietà della chiave KMS utilizzata nell'operazione. Ad esempio, puoi utilizzare la KeyOrigin condizione [kms:](#) per consentire ai responsabili di [GenerateDataKey](#) richiamare una chiave KMS solo quando la Origin chiave KMS è. AWS_KMS Per scoprire se una chiave di condizione può essere utilizzata in questo modo, osserva la descrizione della chiave di condizione.

L'operazione deve essere un'operazione delle risorse delle chiavi KMS, ossia un'operazione autorizzata per una determinata chiave KMS. Per identificare le operazioni delle risorse delle chiavi KMS, nella [tabella Azioni e risorse](#), cerca un valore della KMS key nella colonna Resources per l'operazione. Se si utilizza questo tipo di chiave condizionale con un'operazione non autorizzata per una particolare risorsa chiave KMS, ad esempio [ListKeys](#), l'autorizzazione non è efficace perché la condizione non può mai essere soddisfatta. Non c'è nessuna risorsa chiave KMS coinvolta nell'autorizzazione dell'operazione ListKeys e nessuna proprietà KeySpec.

Gli argomenti seguenti descrivono ogni chiave di AWS KMS condizione e includono esempi di istruzioni sulle politiche che illustrano la sintassi delle politiche.

Utilizzo di operatori con chiavi di condizione

Quando una condizione di policy confronta due set di valori, ad esempio il set di tag in una richiesta e il set di tag in una policy, è necessario spiegare AWS come confrontare i set. Per farlo, IAM definisce due operatori, ForAnyValue e ForAllValues. Utilizza gli operatori solo con le chiavi di condizione multivalore che li richiedono. Non utilizzare operatori con chiavi di condizione a valore singolo. Testa sempre in modo approfondito le istruzioni di policy prima di avvalertene in un ambiente di produzione.

Le chiavi di condizione sono a valore singolo o multivalore. Per determinare se una chiave di AWS KMS condizione è a valore singolo o multivalore, consultate la colonna Tipo di valore nella descrizione della chiave di condizione.

- Le chiavi di condizione a valore singolo hanno al massimo un valore nel contesto di autorizzazione (la richiesta o la risorsa). Ad esempio, poiché ogni chiamata API può provenire da una sola Account AWS, [kms: CallerAccount](#) è una chiave di condizione a valore singolo. Non utilizzare operatori con una chiave di condizione a valore singolo.
- Le chiavi di condizione multivalore hanno più valori nel contesto di autorizzazione (la richiesta o la risorsa). Ad esempio, poiché ogni chiave KMS può avere più alias, [kms: ResourceAliases](#) può avere più valori. Le chiavi di condizione multivalore richiedono un operatore.

Si noti che la differenza tra chiavi di condizione a valore singolo e multivalore dipende dal numero di valori nel contesto di autorizzazione e non dal numero di valori nella condizione di policy.

Warning

L'utilizzo di un operatore con una chiave di condizione a valore singolo può creare un'istruzione di policy eccessivamente permissiva (o eccessivamente restrittiva). Utilizza gli operatori solo con le chiavi di condizione multivalore.

Se si crea o si aggiorna una politica che include un operatore di `ForAllValues` set con le chiavi `kms:EncryptionContext: context key` o `aws:RequestTag/tag-key` condition, AWS KMS restituisce il seguente messaggio di errore:

```
OverlyPermissiveCondition: Using the ForAllValues set operator with a single-valued condition key matches requests without the specified [encryption context or tag] or with an unspecified [encryption context or tag]. To fix, remove ForAllValues.
```

Per informazioni dettagliate su `ForAnyValue` e sugli operatori `ForAllValues`, consulta [Utilizzo di più chiavi e valori](#) nella Guida per l'utente di IAM. Per informazioni sul rischio di utilizzare l'operatore `ForAllValues` set con una condizione a valore singolo, consulta [Security Warning — ForAllValues with single valued key](#) nella IAM User Guide.

Argomenti

- [kms: BypassPolicyLockoutSafetyCheck](#)
- [kms: CallerAccount](#)
- [kms: \(obsoleto\) CustomerMasterKeySpec](#)
- [kms: CustomerMasterKeyUsage \(obsoleto\)](#)
- [kms: DataKeyPairSpec](#)

- [kms:EncryptionAlgorithm](#)
- [kms:: chiave contestuale EncryptionContext](#)
- [kms:EncryptionContextKeys](#)
- [kms:ExpirationModel](#)
- [kms:GrantConstraintType](#)
- [kms:GrantIsForAWSResource](#)
- [kms:GrantOperations](#)
- [kms:GranteePrincipal](#)
- [kms:KeyAgreementAlgorithm](#)
- [kms:KeyOrigin](#)
- [kms:KeySpec](#)
- [kms:KeyUsage](#)
- [kms:MacAlgorithm](#)
- [kms:MessageType](#)
- [kms:MultiRegion](#)
- [kms:MultiRegionKeyType](#)
- [kms:PrimaryRegion](#)
- [kms:ReEncryptOnSameKey](#)
- [kms:RequestAlias](#)
- [kms:ResourceAliases](#)
- [kms:ReplicaRegion](#)
- [kms:RetiringPrincipal](#)
- [kms:RotationPeriodInDays](#)
- [kms:ScheduleKeyDeletionPendingWindowInDays](#)
- [kms:SigningAlgorithm](#)
- [kms:ValidTo](#)
- [kms:ViaService](#)
- [kms:WrappingAlgorithm](#)
- [kms:WrappingKeySpec](#)

kms:BypassPolicyLockoutSafetyCheck

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:BypassPolicyLockoutSafetyCheck	Booleano	A valore singolo	CreateKey PutKeyPolicy	Solo policy IAM Policy delle chiavi e policy IAM

Il tasto `kms:BypassPolicyLockoutSafetyCheck` condition controlla l'accesso alle [PutKeyPolicy](#) operazioni [CreateKey](#) and in base al valore del `BypassPolicyLockoutSafetyCheck` parametro nella richiesta.

La seguente istruzione di policy IAM di esempio impedisce agli utenti di aggirare il controllo di sicurezza di blocco della policy rifiutando l'autorizzazione a creare le chiavi KMS quando il valore del parametro `BypassPolicyLockoutSafetyCheck` nella richiesta `CreateKey` è `true`.

```
{
  "Effect": "Deny",
  "Action": [
    "kms:CreateKey",
    "kms:PutKeyPolicy"
  ],
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:BypassPolicyLockoutSafetyCheck": true
    }
  }
}
```

È inoltre possibile utilizzare la chiave di condizione `kms:BypassPolicyLockoutSafetyCheck` in una policy IAM o della chiave per controllare l'accesso all'operazione `PutKeyPolicy`. La seguente istruzione di policy di esempio da una policy delle chiavi impedisce agli utenti di aggirare il controllo di sicurezza di blocco della policy durante la modifica della policy di una chiave KMS.

Invece di utilizzare un Deny esplicito, questa istruzione di policy utilizza Allow con l'[operatore di condizione Null](#) per consentire l'accesso solo quando la richiesta non include il parametro BypassPolicyLockoutSafetyCheck. Quando il parametro non viene utilizzato, il valore predefinito è false. Questa istruzione di policy leggermente più debole può essere sostituita nel raro caso in cui un bypass sia necessario.

```
{
  "Effect": "Allow",
  "Action": "kms:PutKeyPolicy",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:BypassPolicyLockoutSafetyCheck": true
    }
  }
}
```

Consulta anche

- [kms:KeySpec](#)
- [kms:KeyOrigin](#)
- [kms:KeyUsage](#)

kms:CallerAccount

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:CallerAccount	Stringa	A valore singolo	Operazioni delle risorse delle chiavi KMS Operazioni dell'archivio delle chiavi personalizzate	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per consentire o negare l'accesso a tutte le identità (utenti e ruoli) in un Account AWS. Nelle policy delle chiavi, è possibile utilizzare l'elemento `Principal` per specificare le identità per le quali vale l'istruzione di policy. La sintassi per l'elemento `Principal` non fornisce un modo per specificare tutte le identità in un Account AWS. Tuttavia, è possibile ottenere questo effetto combinando questa chiave di condizione con un `Principal` elemento che specifica tutte le AWS identità.

È possibile utilizzarlo per controllare l'accesso a qualsiasi operazione di risorsa chiave KMS, ovvero a qualsiasi AWS KMS operazione che utilizza una particolare chiave KMS. Per identificare le operazioni delle risorse delle chiavi KMS, nella [tabella Azioni e risorse](#), cerca un valore della KMS key nella colonna `Resources` per l'operazione. È valido anche per le operazioni che gestiscono gli [archivi delle chiavi personalizzate](#).

Ad esempio, la seguente istruzione di policy chiave dimostra come utilizzare la chiave di condizione `kms:CallerAccount`. Questa dichiarazione rientra nella politica chiave Chiave gestita da AWS per Amazon EBS. Combina un `Principal` elemento che specifica tutte le AWS identità con la chiave di `kms:CallerAccount` condizione per consentire in modo efficace l'accesso a tutte le identità in 111122223333. Account AWS Contiene una chiave di AWS KMS condizione aggiuntiva (`kms:ViaService`) per limitare ulteriormente le autorizzazioni permettendo solo le richieste provenienti da Amazon EBS. Per ulteriori informazioni, consulta [kms:ViaService](#).

```
{
  "Sid": "Allow access through EBS for all principals in the account that are
authorized to use EBS",
  "Effect": "Allow",
  "Principal": {"AWS": "*"},
  "Condition": {
    "StringEquals": {
      "kms:CallerAccount": "111122223333",
      "kms:ViaService": "ec2.us-west-2.amazonaws.com"
    }
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:CreateGrant",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

}

kms: (obsoletoCustomerMasterKeySpec)

La chiave di condizione `kms:CustomerMasterKeySpec` è obsoleta. Usa invece la chiave [kms:condition](#). `KeySpec`

Le chiavi di condizione `kms:CustomerMasterKeySpec` e `kms:KeySpec` funzionano allo stesso modo. Solo i nomi differiscono. Ti consigliamo di utilizzare `kms:KeySpec`. Tuttavia, per evitare di interrompere le modifiche, AWS KMS supporta entrambe le chiavi di condizione.

kms: CustomerMasterKeyUsage (obsoleto)

La chiave di condizione `kms:CustomerMasterKeyUsage` è obsoleta. Usa invece la chiave [kms:condition](#). `KeyUsage`

Le chiavi di condizione `kms:CustomerMasterKeyUsage` e `kms:KeyUsage` funzionano allo stesso modo. Solo i nomi differiscono. Ti consigliamo di utilizzare `kms:KeyUsage`. Tuttavia, per evitare di interrompere le modifiche, AWS KMS supporta entrambe le chiavi di condizione.

kms:DataKeyPairSpec

AWS KMS tasti di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
<code>kms:DataKeyPairSpec</code>	Stringa	A valore singolo	GeneratedDataKeyPair GeneratedDataKeyPairWithoutPlaintext	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per controllare l'accesso alle [GenerateDataKeyPairWithoutPlaintext](#) operazioni [GenerateDataKeyPair](#) and in base al valore del `KeyPairSpec` parametro nella richiesta. Ad esempio, è possibile consentire agli utenti di generare solo determinati tipi di coppie di chiavi di dati.

La seguente istruzione di policy delle chiavi di esempio utilizza la chiave di condizione `kms:DataKeyPairSpec` per consentire agli utenti di usare la chiave KMS per generare solo coppie di chiavi di dati RSA.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:GenerateDataKeyPair",
    "kms:GenerateDataKeyPairWithoutPlaintext"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:DataKeyPairSpec": "RSA*"
    }
  }
}
```

Consulta anche

- [kms:KeySpec](#)
- [the section called “kms:EncryptionAlgorithm”](#)
- [the section called “kms:: chiave contestuale EncryptionContext”](#)
- [the section called “kms:EncryptionContextKeys”](#)

kms:EncryptionAlgorithm

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:EncryptionAlgorithm	Stringa	A valore singolo	Decrypt Encrypt GenerateDataKey	Policy delle chiavi e policy IAM

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
			GeneratedDataKeyPair	
			GeneratedDataKeyPairWithoutPlaintext	
			GeneratedDataKeyWithoutPlaintext	
			ReEncrypt	

È possibile utilizzare la chiave di condizione `kms:EncryptionAlgorithm` per controllare l'accesso alle operazioni di crittografia in base all'algoritmo di crittografia usato nell'operazione. Per quanto riguarda [Encrypt](#), [Decrypt](#) e [ReEncrypt](#) Operations, controlla l'accesso in base al valore del [EncryptionAlgorithm](#) parametro nella richiesta. Per le operazioni che generano chiavi di dati e coppie di chiavi di dati, controlla l'accesso in base all'algoritmo di crittografia utilizzato per crittografare la chiave di dati.

Questa chiave condizionale non ha alcun effetto sulle operazioni eseguite all'esterno AWS KMS, come la crittografia con la chiave pubblica in una coppia di chiavi KMS asimmetrica all'esterno di AWS KMS.

EncryptionAlgorithm parametro in una richiesta

Per consentire agli utenti di utilizzare solo un particolare algoritmo di crittografia con una chiave KMS, usa un'istruzione di policy con un effetto Deny e un operatore di condizione `StringNotEquals`. Ad esempio, la seguente istruzione di policy delle chiavi di esempio impedisce ai principali che possono assumere il ruolo `ExampleRole` di utilizzare questa chiave KMS nelle operazioni di crittografia specificate a meno che l'algoritmo di crittografia nella richiesta non sia `RSAES_OAEP_SHA_256`, un algoritmo di crittografia asimmetrica utilizzato con le chiavi KMS RSA.

A differenza di un'istruzione di policy che consente a un utente di utilizzare un particolare algoritmo di crittografia, un'istruzione di policy con un doppio negativo come questa impedisce ad altre policy e concessioni per questa chiave KMS di autorizzare il ruolo a utilizzare altri algoritmi di crittografia. Il Deny in questa istruzione di policy chiave ha la precedenza sulle policy delle chiavi o policy IAM con effetto Allow, nonché su tutte le concessioni per questa chiave KMS e i relativi principali.

```
{
  "Sid": "Allow only one encryption algorithm with this asymmetric KMS key",
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*"
  ],
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "kms:EncryptionAlgorithm": "RSAES_OAEP_SHA_256"
    }
  }
}
```

Algoritmo di crittografia utilizzato per l'operazione

È possibile utilizzare anche la chiave di condizione `kms:EncryptionAlgorithm` per controllare l'accesso alle operazioni basate sull'algoritmo di crittografia utilizzato nell'operazione, anche quando l'algoritmo non è specificato nella richiesta. Ciò consente di richiedere o vietare l'algoritmo `SYMMETRIC_DEFAULT`, che potrebbe non essere specificato in una richiesta perché è il valore predefinito.

Questa funzione ti consente di usare la chiave di condizione `kms:EncryptionAlgorithm` per controllare l'accesso alle operazioni che generano chiavi di dati e coppie di chiavi di dati. Queste operazioni utilizzano soltanto chiavi KMS di crittografia simmetrica e l'algoritmo `SYMMETRIC_DEFAULT`.

Ad esempio, questa policy IAM limita le sue entità principali alla crittografia simmetrica. Nega l'accesso a qualsiasi chiave KMS nell'account di esempio per le operazioni di crittografia a meno che l'algoritmo di crittografia specificato nella richiesta o utilizzato

nell'operazione non sia SYMMETRIC_DEFAULT. GenerateDataKey*Include le aggiunte [GenerateDataKeyGenerateDataKeyWithoutPlaintextGenerateDataKeyPair](#),, e [GenerateDataKeyPairWithoutPlaintext](#)alle autorizzazioni. La condizione non ha alcun effetto su queste operazioni perché utilizzano sempre un algoritmo di crittografia simmetrica.

```
{
  "Sid": "AllowOnlySymmetricAlgorithm",
  "Effect": "Deny",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringNotEquals": {
      "kms:EncryptionAlgorithm": "SYMMETRIC_DEFAULT"
    }
  }
}
```

Consulta anche

- [the section called “kms:MacAlgorithm”](#)
- [kms:SigningAlgorithm](#)

kms:: chiave contestuale EncryptionContext

AWS KMS chiavi di condizione	Tipo di condizion i	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:Encry ptionCont ext: <i>context-key</i>	Stringa	A valore singolo	CreateGrant Encrypt Decrypt Generated ataKey	Policy delle chiavi e policy IAM

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
			GeneratedDataKeyPair	
			GeneratedDataKeyPairWithoutPlainText	
			GeneratedDataKeyWithoutPlainText	
			ReEncrypt	
			RetireGrant	

È possibile utilizzare la chiave di condizione `kms:EncryptionContext:context-key` per controllare l'accesso a una [chiave KMS di crittografia simmetrica](#) in base al [contesto di crittografia](#) in una richiesta per un'[operazione di crittografia](#). Utilizza questa chiave di condizione per valutare sia la chiave sia il valore nella coppia del contesto di crittografia. Per valutare solo le chiavi del contesto di crittografia o richiedere un contesto di crittografia indipendentemente dalle chiavi o dai valori, usa la chiave [kms: EncryptionContextKeys](#) condition.

Note

I valori delle chiavi di condizione devono rispettare le regole sul numero di caratteri delle policy delle chiavi e delle policy IAM. Alcuni caratteri che sono validi in un contesto di crittografia non sono validi nelle policy. Potrebbe non essere possibile utilizzare questa chiave di condizione per esprimere tutti i valori validi del contesto di crittografia. Per maggiori informazioni sulle regole delineate nel documento delle policy delle chiavi, consulta la sezione [Formato della policy della chiave](#). Per informazioni dettagliate sulle regole delineate nel documento delle policy IAM, consulta la sezione [Requisiti del nome IAM](#) nella Guida per l'utente di IAM.

Non è possibile specificare un contesto di crittografia in un'operazione di crittografia con una [chiave KMS asimmetrica](#) o una [chiave KMS HMAC](#). Gli algoritmi asimmetrici e gli algoritmi MAC non supportano un contesto di crittografia.

Per utilizzare la chiave di condizione `kms:EncryptionContext: context-key`, sostituisci il *context-key* segnaposto con la chiave di contesto di crittografia. Sostituisci il segnaposto *context-value* con il valore del contesto di crittografia.

```
"kms:EncryptionContext:context-key": "context-value"
```

Ad esempio, la seguente chiave di condizione specifica un contesto di crittografia in cui la chiave è `AppName` e il valore è `ExampleApp` (`AppName = ExampleApp`).

```
"kms:EncryptionContext:AppName": "ExampleApp"
```

Si tratta di una [chiave di condizione a valore singolo](#). La chiave nella chiave di condizione specifica una particolare chiave di contesto di crittografia (`context-key`). Sebbene sia possibile includere più coppie di contesto di crittografia in ogni richiesta API, la coppia di contesto di crittografia con la `context-key` specificata può avere un solo valore. Ad esempio, la chiave di condizione `kms:EncryptionContext:Department` si applica solo alle coppie di contesto di crittografia con una chiave `Department` e qualsiasi coppia di contesto di crittografia data con la chiave `Department` può avere solo un valore.

Non utilizzare un operatore con la chiave di condizione `kms:EncryptionContext:context-key`. Se crei un'istruzione di policy con un'azione `Allow`, la chiave di condizione `kms:EncryptionContext:context-key` e l'operatore `ForAllValues`, la condizione consente le richieste senza contesto di crittografia e senza le richieste con coppie di contesto di crittografia che non sono specificate nella condizione di policy.

Warning

Non utilizzare un operatore `ForAnyValue` o `ForAllValues` con una chiave di condizione a valore singolo. Questi operatori possono creare una condizione di policy che non richiede valori che intendi richiedere e consente valori che intendi vietare.

Se si crea o si aggiorna una politica che include un operatore `ForAllValues` set con la chiave `kms:EncryptionContext: context-key`, AWS KMS restituisce il seguente messaggio di errore:

OverlyPermissiveCondition:EncryptionContext: Using the `ForAllValues` set operator with a single-valued condition key matches requests without the specified encryption context or with an unspecified encryption context. To fix, remove `ForAllValues`.

Per richiedere una particolare coppia di contesto di crittografia, utilizzare la chiave di condizione `kms:EncryptionContext:context-key` con l'operatore `StringEquals`.

La seguente istruzione di policy chiave di esempio consente alle entità principali che possono assumere il ruolo per utilizzare la chiave KMS in una richiesta `GenerateDataKey` solo quando il contesto di crittografia nella richiesta include la coppia `AppName:ExampleApp`. Altre coppie di contesto di crittografia sono consentite.

Il nome della chiave non fa distinzione tra maiuscole e minuscole. La distinzione tra maiuscole e minuscole che fa il valore è determinata dall'operatore della condizione, ad esempio `StringEquals`. Per informazioni dettagliate, vedi [Distinzione tra maiuscole e minuscole della condizione del contesto](#).

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    }
  }
}
```

Per richiedere una coppia di contesti di crittografia e vietare tutte le altre coppie di contesti di crittografia, utilizza entrambi `kms:EncryptionContext:context-key` e [kms:EncryptionContextKeys](#) nell'informativa sulla politica. La seguente istruzione di policy chiave utilizza la chiave di condizione `kms:EncryptionContext:AppName` per richiedere la coppia di contesto di crittografia `AppName=ExampleApp` nella richiesta. Utilizza inoltre una chiave di condizione `kms:EncryptionContextKeys` con l'operatore `ForAllValues` per consentire solo la chiave di contesto di crittografia `AppName`.

L'operatore `ForAllValues` limita le chiavi di contesto di crittografia nella richiesta a `AppName`. Se la condizione `kms:EncryptionContextKeys` con l'operatore `ForAllValues` è stata utilizzata da sola in un'istruzione di policy, questo operatore consentirebbe le richieste senza contesto di crittografia. Tuttavia, se la richiesta non avesse un contesto di crittografia, la condizione `kms:EncryptionContext:AppName` avrebbe esito negativo. Per dettagli sull'operatore `ForAllValues`, consulta [Utilizzo di più chiavi e valori](#) nella Guida per l'utente di IAM.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/KeyUsers"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    },
    "ForAllValues:StringEquals": {
      "kms:EncryptionContextKeys": [
        "AppName"
      ]
    }
  }
}
```

È inoltre possibile utilizzare questa chiave di condizione per negare l'accesso a una chiave KMS per una determinata operazione. La seguente istruzione di policy della chiave di esempio utilizza un effetto `Deny` per impedire all'entità principale di utilizzare la chiave KMS se il contesto di crittografia nella richiesta include una coppia di contesto di crittografia `Stage=Restricted`. Questa condizione consente una richiesta con altre coppie di contesto di crittografia, incluse le coppie di contesto di crittografia con la chiave `Stage` e altri valori, come `Stage=Test`.

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
```

```
"StringEquals": {  
  "kms:EncryptionContext:Stage": "Restricted"  
}  
}
```

Utilizzo di più coppie di contesto di crittografia

È possibile richiedere o vietare più coppie di contesto di crittografia. È inoltre possibile richiedere una delle diverse coppie di contesto di crittografia. Per informazioni dettagliate sulla logica utilizzata per interpretare queste condizioni, consulta [Creazione di una condizione con più chiavi o valori](#) nella Guida per l'utente di IAM.

Note

Le versioni precedenti di questo argomento mostravano dichiarazioni di policy che utilizzavano `ForAnyValue` e `ForAllValues` impostavano gli operatori con la chiave di condizione `kms:EncryptionContext: context-key`. L'utilizzo di un operatore con una [chiave di condizione a valore singolo](#) può comportare policy che consentono richieste senza contesto di crittografia e coppie di contesto di crittografia non specificate.

Ad esempio, una condizione di policy con l'effetto `Allow`, l'operatore `ForAllValues` e la chiave di condizione `"kms:EncryptionContext:Department": "IT"` non limita il contesto di crittografia alla coppia `"Department=IT"`. Permette richieste senza contesto di crittografia e richieste con coppie di contesto di crittografia non specificate, come `Stage=Restricted`.

Rivedi le tue politiche ed elimina l'operatore `set` da qualsiasi condizione con `kms:EncryptionContext: context-key`. I tentativi di creare o aggiornare una policy con questo formato hanno esito negativo con un'eccezione `OverlyPermissiveCondition`. Per risolvere il problema, è necessario eliminare l'operatore.

Per richiedere più coppie di contesto di crittografia, elenca le coppie nella stessa condizione. La seguente istruzione di policy della chiave di esempio richiede due coppie di contesto di crittografia, `Department=IT` e `Project=Alpha`. Poiché le condizioni hanno chiavi diverse (`kms:EncryptionContext:Department` and `kms:EncryptionContext:Project`), sono implicitamente collegate da un operatore `AND`. Altre coppie di contesto di crittografia sono consentite, ma non obbligatorie.

```
{
```

```
"Effect": "Allow",
"Principal": {
  "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
},
"Action": "kms:Decrypt",
"Resource": "*",
"Condition": {
  "StringEquals": {
    "kms:EncryptionContext:Department": "IT",
    "kms:EncryptionContext:Project": "Alpha"
  }
}
```

Per richiedere una coppia di contesto di crittografia OPPURE un'altra coppia, inserire ciascuna chiave di condizione in un'istruzione di policy separata. La seguente policy chiave di esempio richiede coppie Department=IT o Project=Alpha, o entrambe. Altre coppie di contesto di crittografia sono consentite, ma non obbligatorie.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Department": "IT"
    }
  }
},
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Project": "Alpha"
    }
  }
}
```



```
}
}
```

Per richiedere coppie di crittografia particolari ed escludere tutte le altre coppie di contesto di crittografia, utilizza entrambi `kms:EncryptionContext: contesto-chiave` e [kms:EncryptionContextKeys](#) nella dichiarazione politica. La seguente dichiarazione politica chiave utilizza la condizione `kms:EncryptionContext: chiave contestuale` per richiedere un contesto di crittografia con entrambe le coppie. `Department=IT Project=Alpha` Utilizza una chiave di condizione `kms:EncryptionContextKeys` con l'operatore `ForAllValues` per consentire solo le chiavi di contesto di crittografia `Department` e `Project`.

L'operatore `ForAllValues` limita le chiavi di contesto di crittografia nella richiesta a `Department` e `Project`. Se fosse usato da solo in una condizione, questo operatore di set consentirebbe richieste senza contesto di crittografia, ma in questa configurazione, la chiave di contesto `kms:EncryptionContext`: in questa condizione avrebbe esito negativo.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Department": "IT",
      "kms:EncryptionContext:Project": "Alpha"
    },
    "ForAllValues:StringEquals": {
      "kms:EncryptionContextKeys": [
        "Department",
        "Project"
      ]
    }
  }
}
```

È inoltre possibile vietare più coppie di contesto di crittografia. La seguente istruzione di policy della chiave di esempio utilizza un effetto `Deny` per impedire all'entità principale di utilizzare le chiavi KMS se il contesto di crittografia nella richiesta include una coppia `Stage=Restricted` o `Stage=Production`.

Valori multipli (Restricted e Production) per la stessa chiave (kms:EncryptionContext:Stage) sono implicitamente collegati da un'OR. Per informazioni dettagliate, consulta [Logica di valutazione per condizioni con più chiavi o valori](#) nella Guida per l'utente di IAM.

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Stage": [
        "Restricted",
        "Production"
      ]
    }
  }
}
```

Distinzione tra maiuscole e minuscole della condizione del contesto

Il contesto di crittografia specificato in un'operazione di decrittografia deve essere una corrispondenza esatta, che fa distinzione tra maiuscole e minuscole per il contesto di crittografia specificato nell'operazione di crittografia. Solo l'ordine delle coppie in un contesto di crittografia con più coppie può variare.

Tuttavia, nelle condizioni della policy, la chiave di condizione non fa distinzione tra maiuscole e minuscole. Se il valore della condizione fa distinzione tra minuscole e maiuscole viene determinato dall'[operatore della condizione della policy](#) che utilizzi, ad esempio StringEquals o StringEqualsIgnoreCase.

Per questo motivo, la chiave di condizione, che include il prefisso kms:EncryptionContext: e la sostituzione *context-key*, non fa distinzione tra minuscole e maiuscole. Una policy che utilizza questa condizione non controlla se i caratteri degli elementi della chiave di condizione sono in maiuscolo o minuscolo. Se il valore della condizione fa distinzione tra minuscole e maiuscole, ovvero la sostituzione di *context-value*, viene determinato dall'operatore della condizione della policy.

Ad esempio, la seguente istruzione di policy consente l'operazione quando il contesto di crittografia include una chiave Appname, senza considerare se i caratteri sono in minuscolo o maiuscolo. La condizione `StringEquals` richiede che `ExampleApp` sia scritto con caratteri maiuscoli come specificato.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Appname": "ExampleApp"
    }
  }
}
```

Per richiedere una chiave contestuale di crittografia con distinzione tra maiuscole e minuscole, utilizza la condizione di `EncryptionContextKeys` policy [kms:](#) con un operatore di condizione con distinzione tra maiuscole e minuscole, ad esempio. `StringEquals` In questa condizione della policy, poiché la chiave di contesto di crittografia è il valore della condizione della policy; la distinzione tra minuscole e maiuscole è determinata dall'operatore della condizione.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:EncryptionContextKeys": "AppName"
    }
  }
}
```

Per richiedere una valutazione con distinzione tra maiuscole e minuscole sia della chiave che del valore del contesto di crittografia, utilizzate insieme le condizioni della politica

`kms:EncryptionContextKeys` and `kms:EncryptionContext`: chiave contestuale nella stessa dichiarazione di policy. L'operatore condizione con distinzione tra maiuscole e minuscole (come `StringEquals`) si applica sempre al valore della condizione. La chiave di contesto di crittografia (ad esempio `AppName`) è il valore della condizione `kms:EncryptionContextKeys`. Il valore del contesto di crittografia (ad esempio `ExampleApp`) è il valore della condizione: `context-key`. `kms:EncryptionContext`

Ad esempio, nella seguente istruzione della policy della chiave di esempio, in quanto l'operatore `StringEquals` fa distinzione tra minuscole e maiuscole, sia la chiave del contesto di crittografia sia il valore del contesto di crittografia fanno distinzione tra minuscole e maiuscole.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:EncryptionContextKeys": "AppName"
    },
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    }
  }
}
```

Utilizzo di variabili in una condizione del contesto di crittografia

La chiave e il valore in una coppia del contesto di crittografia devono essere stringhe letterali semplici. Non possono essere integer o oggetti o un qualsiasi tipo non completamente risolto. Se si utilizza un tipo diverso, ad esempio un numero intero o un float, lo AWS KMS interpreta come una stringa letterale.

```
"encryptionContext": {
  "department": "10103.0"
}
```

Tuttavia, il valore della chiave di condizione `kms:EncryptionContext:context-key` può essere una [variabile della policy IAM](#). Queste variabili di policy vengono risolte in fase di esecuzione

(runtime) in base ai valori nella richiesta. Ad esempio, `aws:CurrentTime` restituisce l'ora della richiesta e `aws:username` restituisce il nome descrittivo dell'intermediario.

Puoi utilizzare queste variabili di policy per creare un'istruzione di policy con una condizione che richiede informazioni molto specifiche in un contesto di crittografia, ad esempio il nome utente dell'intermediario. Dal momento che contiene una variabile, puoi utilizzare la stessa istruzione di policy per tutti gli utenti che possono assumere il ruolo. Non è necessario scrivere un'istruzione di policy separata per ogni utente.

Considera una situazione in cui desideri che tutti gli utenti che possono assumere un ruolo utilizzino la stessa chiave KMS per crittografare e decrittare i dati. Tuttavia, vuoi consentire loro di decrittografare solo i dati che hanno crittografato. Inizia richiedendo che ogni richiesta AWS KMS includa un contesto di crittografia in cui la chiave è `user` e il valore è il nome AWS utente del chiamante, come il seguente.

```
"encryptionContext": {  
  "user": "bob"  
}
```

Quindi, per applicare questo requisito, puoi utilizzare un'istruzione di policy come quella nell'esempio seguente. Questa istruzione di policy concede al ruolo `TestTeam` l'autorizzazione per crittografare e decrittare i dati con la chiave KMS. Tuttavia, l'autorizzazione è valida solo quando il contesto di crittografia nella richiesta include una coppia `"user": "<username>"`. Per rappresentare il nome utente, la condizione utilizza la variabile della policy [aws:username](#).

Quando la richiesta viene valutata, il nome utente dell'intermediario sostituisce la variabile nella condizione. Pertanto, la condizione richiede un contesto di crittografia `"user": "bob"` per "bob" e `"user": "alice"` per "alice".

```
{  
  "Effect": "Allow",  
  "Principal": {  
    "AWS": "arn:aws:iam::111122223333:role/TestTeam"  
  },  
  "Action": [  
    "kms:Decrypt",  
    "kms:Encrypt"  
  ],  
  "Resource": "*",  
  "Condition": {
```

```
"StringEquals": {
  "kms:EncryptionContext:user": "${aws:username}"
}
}
```

Puoi utilizzare una variabile di policy IAM solo nel valore della coppia di chiavi di condizione `kms:EncryptionContext:context-key`. Non è possibile utilizzare una variabile nella chiave.

Puoi anche utilizzare le [chiavi di contesto specifiche del provider](#) nelle variabili. Queste chiavi di contesto identificano in modo univoco gli utenti che hanno effettuato l'accesso AWS utilizzando la federazione delle identità Web.

Come tutte le variabili, queste possono essere utilizzate solo nella condizione `kms:EncryptionContext:context-key` della policy, non nel contesto di crittografia effettivo. Inoltre, possono essere utilizzate solo nel valore della condizione, non nella chiave.

Ad esempio, la seguente istruzione della policy delle chiavi è simile a quella precedente. Tuttavia, la condizione richiede un contesto di crittografia in cui la chiave è `sub` e il valore identifica in modo univoco un utente connesso a un pool di utenti Amazon Cognito. Per dettagli sull'identificazione di utenti e ruoli in Amazon Cognito, consulta [Ruoli IAM](#) nella [Guida per sviluppatori di Amazon Cognito](#).

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/TestTeam"
  },
  "Action": [
    "kms:Decrypt",
    "kms:Encrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:sub": "${cognito-identity.amazonaws.com:sub}"
    }
  }
}
```

Consulta anche

- [the section called “kms:EncryptionContextKeys”](#)

- [the section called “kms:GrantConstraintType”](#)

kms:EncryptionContextKeys

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:EncryptionContextKeys	Stringa (elenco)	Multivalore	CreateGrant Decrypt Encrypt GenerateDataKey GenerateDataKeyPair GenerateDataKeyPairWithoutPlaintext GenerateDataKeyWithoutPlaintext ReEncrypt RetireGrant	Policy delle chiavi e policy IAM

È possibile utilizzare la chiave di condizione `kms:EncryptionContextKeys` per controllare l'accesso a una [chiave KMS di crittografia simmetrica](#) in base al [contesto di crittografia](#) in una richiesta per un'operazione di crittografia. Utilizza questa chiave di condizione per valutare solo la chiave in ciascuna coppia del contesto di crittografia. Per valutare sia la chiave sia il valore nel contesto di crittografia, usa la chiave di condizione `kms:EncryptionContext:context-key`.

Non è possibile specificare un contesto di crittografia in un'operazione di crittografia con una [chiave KMS asimmetrica](#) o una [chiave KMS HMAC](#). Gli algoritmi asimmetrici e gli algoritmi MAC non supportano un contesto di crittografia.

Note

I valori delle chiavi condizionali, inclusa una chiave di contesto di crittografia, devono essere conformi ai caratteri e alle regole di codifica delle politiche AWS KMS chiave. Potrebbe non essere possibile utilizzare questa chiave di condizione per esprimere tutte le chiavi valide nel contesto di crittografia. Per maggiori informazioni sulle regole delineate nel documento delle policy delle chiavi, consulta la sezione [Formato della policy della chiave](#). Per informazioni dettagliate sulle regole delineate nel documento delle policy IAM, consulta la sezione [Requisiti del nome IAM](#) nella Guida per l'utente di IAM.

Si tratta di una [chiave di condizione multivalore](#). È possibile specificare più coppie di contesto di crittografia in ogni richiesta API. `kms:EncryptionContextKeys` confronta le chiavi del contesto di crittografia nella richiesta con il set di chiavi del contesto di crittografia nella policy. Per determinare il modo in cui questi set vengono confrontati, devi fornire un operatore `ForAnyValue` o `ForAllValues` nella condizione di policy. Per dettagli sugli operatori, consulta [Utilizzo di più chiavi e valori](#) nella Guida per l'utente di IAM.

- `ForAnyValue`: almeno una chiave di contesto di crittografia nella richiesta deve corrispondere a una chiave di contesto di crittografia nella condizione della policy. Sono consentite altre chiavi di contesto di crittografia. Se la richiesta non dispone di contesto di crittografia, la condizione non viene soddisfatta.
- `ForAllValues`: ogni chiave di contesto di crittografia nella richiesta deve corrispondere a una chiave di contesto di crittografia nella condizione della policy. Questo operatore limita le chiavi di contesto di crittografia a quelle nella condizione della policy. Non richiede alcuna chiave di contesto di crittografia, ma vieta le chiavi di contesto di crittografia non specificate.

La seguente istruzione di policy della chiave di esempio utilizza la chiave di condizione `kms:EncryptionContextKeys` con l'operatore `ForAnyValue`. Questa istruzione di policy utilizza una chiave KMS per le operazioni specificate, ma solo quando almeno una delle coppie del contesto di crittografia nella richiesta include la chiave `AppName`, indipendentemente dal suo valore.

Ad esempio, questa istruzione di policy chiave consente una richiesta `GenerateDataKey` con due coppie di contesto di crittografia, `AppName=Helper` e `Project=Alpha`, perché la prima coppia di contesto di crittografia soddisfa la condizione. Una richiesta con solo `Project=Alpha` o senza contesto di crittografia avrebbe esito negativo.

Poiché l'operazione di [StringEquals](#) condizione fa distinzione tra maiuscole e minuscole, questa dichiarazione di policy richiede l'ortografia e la minuscola della chiave di contesto di crittografia. Tuttavia, è possibile utilizzare un operatore della condizione che ignora se la chiave ha caratteri in minuscolo o maiuscolo, ad esempio `StringEqualsIgnoreCase`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:EncryptionContextKeys": "AppName"
    }
  }
}
```

Inoltre, è possibile utilizzare la chiave di condizione `kms:EncryptionContextKeys` per richiedere un qualsiasi contesto di crittografia nelle operazioni di crittografia che utilizzano la chiave KMS.

La seguente istruzione di policy della chiave di esempio utilizza la chiave di condizione `kms:EncryptionContextKeys` con [Null condition operator](#) per consentire l'accesso alla chiave KMS solo quando il contesto di crittografia nella richiesta dell'API è diverso da null. Questa condizione non verifica le chiavi o i valori del contesto di crittografia. Verifica solo che il contesto di crittografia esista.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
```

```
"Action": [
  "kms:Encrypt",
  "kms:GenerateDataKey*"
],
"Resource": "*",
"Condition": {
  "Null": {
    "kms:EncryptionContextKeys": false
  }
}
```

Consulta anche

- [kms:: chiave contestuale EncryptionContext](#)
- [kms:GrantConstraintType](#)

kms:ExpirationModel

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:ExpirationModel	Stringa	A valore singolo	ImportKeyMaterial	Policy delle chiavi e policy IAM

Il tasto kms:ExpirationModel condition controlla l'accesso all'[ImportKeyMaterial](#) operazione in base al valore del [ExpirationModel](#) parametro nella richiesta.

ExpirationModel è un parametro opzionale che determina se il materiale della chiave importato scade. I valori validi sono KEY_MATERIAL_EXPIRES e KEY_MATERIAL_DOES_NOT_EXPIRE. Il valore predefinito è KEY_MATERIAL_EXPIRES.

La data e l'ora di scadenza sono determinate dal valore del [ValidTo](#) parametro. Il parametro ValidTo è obbligatorio a meno che il valore del parametro ExpirationModel non sia KEY_MATERIAL_DOES_NOT_EXPIRE. Puoi anche utilizzare la chiave [kms: ValidTo](#) condition per richiedere una data di scadenza particolare come condizione per l'accesso.

La seguente istruzione di policy di esempio utilizza la chiave di condizione `kms:ExpirationModel` per consentire agli utenti di importare materiale della chiave in una chiave KMS solo quando la richiesta include il parametro `ExpirationModel` e il suo valore è `KEY_MATERIAL_DOES_NOT_EXPIRE`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ExpirationModel": "KEY_MATERIAL_DOES_NOT_EXPIRE"
    }
  }
}
```

È inoltre possibile utilizzare la chiave di condizione `kms:ExpirationModel` per consentire agli utenti di importare il materiale chiave solo quando il materiale della chiave scade. La seguente istruzione di policy di esempio utilizza la chiave di condizione `kms:ExpirationModel` con [Null condition operator](#) per consentire agli utenti di importare materiale chiave solo quando la richiesta non include il parametro `ExpirationModel`. Il valore predefinito per `ExpirationModel` è `KEY_MATERIAL_EXPIRES`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:ExpirationModel": true
    }
  }
}
```

Consulta anche

- [kms:ValidTo](#)
- [kms:WrappingAlgorithm](#)
- [kms:WrappingKeySpec](#)

kms:GrantConstraintType

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:GrantConstraintType	Stringa	A valore singolo	CreateGrant RetireGrant	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per controllare l'accesso all'[CreateGrant](#) operazione in base al tipo di [vincolo di concessione](#) contenuto nella richiesta.

Quando crei una concessione, opzionalmente è possibile specificare un vincolo di concessione per consentire le operazioni che la concessione consente solo quando è presente un determinato [contesto di crittografia](#). Il vincolo di concessione può essere di due tipi: `EncryptionContextEquals` o `EncryptionContextSubset`. È possibile utilizzare questa chiave di condizione per controllare che la richiesta contenga uno dei due tipi.

Important

Non includere informazioni riservate o sensibili in questo campo. Questo campo può essere visualizzato in testo semplice nei CloudTrail log e in altri output.

La seguente istruzione di policy delle chiavi di esempio utilizza la chiave di condizione `kms:GrantConstraintType` per consentire agli utenti di creare concessioni solo quando la richiesta include un vincolo di concessione `EncryptionContextEquals`. L'esempio illustra un'istruzione di policy in una policy delle chiavi.

```
{
  "Effect": "Allow",
  "Principal": {
```

```

    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:GrantConstraintType": "EncryptionContextEquals"
    }
  }
}

```

Consulta anche

- [kms:: chiave contestuale EncryptionContext](#)
- [kms:EncryptionContextKeys](#)
- [kms:GrantIsForAWSResource](#)
- [kms:GrantOperations](#)
- [kms:GranteePrincipal](#)
- [kms:RetiringPrincipal](#)

kms:GrantIsForAWSResource

AWS KMS tasti di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:GrantIsForAWSResource	Booleano	A valore singolo	CreateGrant ListGrants RevokeGrant	Policy delle chiavi e policy IAM

Consente o nega l'[CreateGrant](#) autorizzazione per [RevokeGrant](#) le operazioni solo quando un [AWS servizio integrato con AWS KMS](#) richiama l'operazione per conto dell'utente. [ListGrants](#) Questa condizione di policy non consente all'utente di chiamare direttamente queste operazioni di concessione.

La seguente istruzione di policy della chiave di esempio utilizza la chiave di condizione `kms:GrantIsForAWSResource`. Consente AWS ai servizi integrati con AWS KMS, come Amazon EBS, di creare sovvenzioni su questa chiave KMS per conto del committente specificato.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:GrantIsForAWSResource": true
    }
  }
}
```

Consulta anche

- [kms:GrantConstraintType](#)
- [kms:GrantOperations](#)
- [kms:GranteePrincipal](#)
- [kms:RetiringPrincipal](#)

kms:GrantOperations

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
<code>kms:GrantOperations</code>	Stringa	Multivalore	<code>CreateGrant</code>	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per controllare l'accesso all'[CreateGrant](#) operazione in base alle [operazioni di concessione](#) contenute nella richiesta. Ad esempio, è possibile consentire

agli utenti di creare concessioni che delegano l'autorizzazione di crittografare ma non decrittografare. Per ulteriori informazioni sulle autorizzazioni, consulta [Utilizzo di concessioni](#)

Questa è una [chiave di condizione multivalore](#). `kms:GrantOperations` confronta l'insieme delle operazioni di concessione nella richiesta `CreateGrant` all'insieme di operazioni di concessione nella policy. Per determinare il modo in cui questi set vengono confrontati, devi fornire un operatore `ForAnyValue` o `ForAllValues` nella condizione di policy. Per dettagli sugli operatori, consulta [Utilizzo di più chiavi e valori](#) nella Guida per l'utente di IAM.

- `ForAnyValue`: almeno un'operazione di concessione nella richiesta deve corrispondere a una delle operazioni di concessione nella condizione di policy. Sono ammesse altre operazioni di concessione.
- `ForAllValues`: Ogni operazione di concessione nella richiesta deve corrispondere a un'operazione di concessione nella condizione della politica. Questo operatore limita le operazioni di concessione a quelle specificate nella condizione della policy. Non richiede alcuna operazione di concessione, ma vieta le operazioni di concessione non specificate.

`ForAllValues` restituisce true anche quando non ci sono operazioni di concessione nella richiesta, ma `CreateGrant` non le consente. Se il parametro `Operations` è mancante o ha un valore nullo, la richiesta `CreateGrant` ha esito negativo.

La seguente istruzione di policy delle chiavi di esempio utilizza la chiave di condizione `kms:GrantOperations` per creare concessioni solo quando le operazioni di concessione sono `Encrypt`, `ReEncryptTo` o entrambi. Se la concessione include altre operazioni, la richiesta `CreateGrant` ha esito negativo.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "ForAllValues:StringEquals": {
      "kms:GrantOperations": [
        "Encrypt",
        "ReEncryptTo"
      ]
    }
  }
}
```

```

    }
  }
}
```

Se si modifica l'operatore nella condizione di policy in `ForAnyValue`, l'istruzione di policy richiederebbe che almeno una delle operazioni di concessione nella concessione sia `Encrypt` o `ReEncryptTo`, ma consentirebbe altre operazioni di concessione, come `Decrypt` o `ReEncryptFrom`.

Consulta anche

- [kms:GrantConstraintType](#)
- [kms:GrantIsForAWSResource](#)
- [kms:GranteePrincipal](#)
- [kms:RetiringPrincipal](#)

kms:GranteePrincipal

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
<code>kms:GranteePrincipal</code>	Stringa	A valore singolo	<code>CreateGrant</code>	Policy IAM e della chiave

È possibile utilizzare questa chiave di condizione per controllare l'accesso all'[CreateGrant](#) operazione in base al valore del [GranteePrincipal](#) parametro nella richiesta. Ad esempio, è possibile creare autorizzazioni per utilizzare una chiave KMS solo quando il principale assegnatario nella richiesta `CreateGrant` corrisponde al principale specificato nell'istruzione della condizione.

Per specificare il principale beneficiario, utilizza l'Amazon Resource Name (ARN) di un principale. AWS I principali validi includono utenti IAM Account AWS, ruoli IAM, utenti federati e utenti assunti. Per informazioni sulla sintassi ARN per un principale, consulta [IAM ARNs nella IAM User Guide](#).

La seguente istruzione di policy delle chiavi di esempio utilizza la chiave di condizione `kms:GranteePrincipal` per creare autorizzazioni per una chiave KMS solo quando il principale assegnatario nella concessione è `LimitedAdminRole`.


```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:GranteePrincipal": "arn:aws:iam::111122223333:role/LimitedAdminRole"
    }
  }
}
```

Consulta anche

- [kms:GrantConstraintType](#)
- [kms:GrantIsForAWSResource](#)
- [kms:GrantOperations](#)
- [kms:RetiringPrincipal](#)

kms:KeyAgreementAlgorithm

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:KeyAgreementAlgorithm	Stringa	A valore singolo	DeriveSharedSecret	Policy delle chiavi e policy IAM

È possibile utilizzare il tasto kms:KeyAgreementAlgorithm condition per controllare l'accesso all'[DeriveSharedSecret](#) operazione in base al valore del KeyAgreementAlgorithm parametro nella richiesta. L'unico valore valido per KeyAgreementAlgorithm èECDH.

Ad esempio, la seguente dichiarazione politica chiave utilizza la chiave kms:KeyAgreementAlgorithm condition per negare qualsiasi accesso a DeriveSharedSecret meno che non lo KeyAgreementAlgorithm siaECDH.

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:DeriveSharedSecret",
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "kms:KeyAgreementAlgorithm": "ECDH"
    }
  }
}
```

Consulta anche

- [the section called “kms:KeyUsage”](#)

kms:KeyOrigin

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:KeyOrigin	Stringa	A valore singolo	CreateKey Operazioni delle risorse delle chiavi KMS	Policy IAM Policy delle chiavi e policy IAM

La chiave di condizione kms:KeyOrigin controlla l'accesso alle operazioni in base al valore della proprietà Origin della chiave KMS creata o utilizzata nell'operazione. Funziona come una condizione di risorsa o una condizione di richiesta.

È possibile utilizzare questa chiave di condizione per controllare l'accesso all'[CreateKey](#) operazione in base al valore del parametro [Origin](#) nella richiesta. I valori validi di Origin sono AWS_KMS, AWS_CLOUDHSM ed EXTERNAL.

Ad esempio, è possibile creare una chiave KMS solo quando il materiale chiave viene generato in AWS KMS (AWS_KMS), solo quando il materiale chiave viene generato in un AWS CloudHSM cluster associato a un [archivio chiavi personalizzato](#) (AWS_CLOUDHSM) o solo quando il [materiale chiave viene importato](#) da una fonte esterna (EXTERNAL).

L'esempio seguente di dichiarazione politica chiave utilizza la chiave `kms:KeyOrigin` condition per creare una chiave KMS solo quando AWS KMS crea il materiale chiave.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
      },
      "Action": "kms:CreateKey",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:KeyOrigin": "AWS_KMS"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:GenerateDataKeyPair",
        "kms:GenerateDataKeyPairWithoutPlaintext",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
      "Condition": {
```

```

    "StringEquals": {
      "kms:KeyOrigin": "AWS_CLOUDHSM"
    }
  }
}
]
}

```

La chiave di condizione `kms:KeyOrigin` può essere utilizzata anche per controllare l'accesso alle operazioni che utilizzano o gestiscono una chiave KMS in base alla proprietà `Origin` della chiave KMS utilizzata per l'operazione. L'operazione deve essere un'operazione delle risorse delle chiavi KMS, ossia un'operazione autorizzata per una determinata chiave KMS. Per identificare le operazioni delle risorse delle chiavi KMS, nella [tabella Azioni e risorse](#), cerca un valore della KMS key nella colonna `Resources` per l'operazione.

Ad esempio, la policy IAM seguente consente ai principali di eseguire le operazioni specificate sulle risorse chiave KMS, ma solo con le chiavi KMS nell'account che sono state create in un archivio chiavi personalizzato.

```

{
  "Effect": "Allow",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:GenerateDataKey",
    "kms:GenerateDataKeyWithoutPlaintext",
    "kms:GenerateDataKeyPair",
    "kms:GenerateDataKeyPairWithoutPlaintext",
    "kms:ReEncrypt*"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "AWS_CLOUDHSM"
    }
  }
}

```

Consulta anche

- [kms:BypassPolicyLockoutSafetyCheck](#)

- [kms:KeySpec](#)
- [kms:KeyUsage](#)

kms:KeySpec

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:KeySpec	Stringa	A valore singolo	CreateKey	Policy IAM
			Operazioni delle risorse delle chiavi KMS	Policy delle chiavi e policy IAM

La chiave di condizione kms:KeySpec controlla l'accesso alle operazioni in base al valore della proprietà KeySpec della chiave KMS creata o utilizzata nell'operazione.

Puoi utilizzare questa chiave di condizione in una policy IAM per controllare l'accesso all'[CreateKey](#) operazione in base al valore del [KeySpec](#) parametro in una CreateKey richiesta. Ad esempio, puoi utilizzare questa condizione per permettere agli utenti di creare solo chiavi KMS di crittografia simmetrica o solo chiavi KMS HMAC.

La seguente istruzione di policy IAM di esempio utilizza la chiave di condizione kms:KeySpec per permettere ai principali di creare solo chiavi KMS asimmetriche RSA. L'autorizzazione è valida solo quando KeySpec nella richiesta inizia con RSA_.

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:KeySpec": "RSA_*"
    }
  }
}
```

La chiave di condizione `kms:KeySpec` può essere utilizzata anche per controllare l'accesso alle operazioni che utilizzano o gestiscono una chiave KMS in base alla proprietà `KeySpec` della chiave KMS utilizzata per l'operazione. L'operazione deve essere un'operazione delle risorse delle chiavi KMS, ossia un'operazione autorizzata per una determinata chiave KMS. Per identificare le operazioni delle risorse delle chiavi KMS, nella [tabella Azioni e risorse](#), cerca un valore della `KMS key` nella colonna `Resources` per l'operazione.

Ad esempio, la policy IAM seguente permette ai principali di eseguire le operazioni specificate sulle risorse chiave KMS, ma solo con le chiavi KMS di crittografia simmetrica presenti nell'account.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:DescribeKey"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:KeySpec": "SYMMETRIC_DEFAULT"
    }
  }
}
```

Consulta anche

- [kms:BypassPolicyLockoutSafetyCheck](#)
- [kms: \(obsoletoCustomerMasterKeySpec\)](#)
- [kms:DataKeyPairSpec](#)
- [kms:KeyOrigin](#)
- [kms:KeyUsage](#)

kms:KeyUsage

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:KeyUsage	Stringa	A valore singolo	CreateKey Operazioni delle risorse delle chiavi KMS	Policy IAM Policy delle chiavi e policy IAM

La chiave di condizione `kms:KeyUsage` controlla l'accesso alle operazioni in base al valore della proprietà `KeyUsage` della chiave KMS creata o utilizzata nell'operazione.

È possibile utilizzare questa chiave di condizione per controllare l'accesso all'[CreateKey](#) operazione in base al valore del [KeyUsage](#) parametro nella richiesta. I valori validi per `KeyUsage` sono `ENCRYPT_DECRYPT`, `SIGN_VERIFY`, `GENERATE_VERIFY_MAC`, e `KEY_AGREEMENT`.

Ad esempio, puoi creare una chiave KMS solo quando `KeyUsage` è `ENCRYPT_DECRYPT` o negare l'autorizzazione a un utente quando `KeyUsage` è `SIGN_VERIFY`.

La seguente istruzione di policy IAM di esempio utilizza la chiave di condizione `kms:KeyUsage` per creare una chiave KMS solo quando `KeyUsage` è `ENCRYPT_DECRYPT`.

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:KeyUsage": "ENCRYPT_DECRYPT"
    }
  }
}
```

La chiave di condizione `kms:KeyUsage` può essere utilizzata anche per controllare l'accesso alle operazioni che utilizzano o gestiscono una chiave KMS in base alla proprietà `KeyUsage` della chiave KMS nell'operazione. L'operazione deve essere un'operazione delle risorse delle chiavi KMS, ossia un'operazione autorizzata per una determinata chiave KMS. Per identificare le operazioni delle

risorse delle chiavi KMS, nella [tabella Azioni e risorse](#), cerca un valore della KMS key nella colonna Resources per l'operazione.

Ad esempio, la policy IAM seguente consente ai principali di eseguire le operazioni specificate sulle risorse chiave KMS, ma solo con le chiavi KMS nell'account che vengono utilizzate per la firma e la verifica.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:CreateGrant",
    "kms:DescribeKey",
    "kms:GetPublicKey",
    "kms:ScheduleKeyDeletion"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:KeyUsage": "SIGN_VERIFY"
    }
  }
}
```

Consulta anche

- [kms:BypassPolicyLockoutSafetyCheck](#)
- [kms: CustomerMasterKeyUsage](#) (obsoleto)
- [kms:KeyOrigin](#)
- [kms:KeySpec](#)

kms:MacAlgorithm

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:MacAlgorithm	Stringa	A valore singolo	GenerateMac VerifyMac	Policy delle chiavi e policy IAM

È possibile utilizzare il tasto `kms:MacAlgorithm` condition per controllare l'accesso alle [VerifyMac](#) operazioni [GenerateMac](#) and in base al valore del `MacAlgorithm` parametro nella richiesta.

La seguente policy della chiave di esempio permette agli utenti che possono assumere il ruolo `testers` di utilizzare la chiave KMS HMAC per generare e verificare i tag HMAC solo quando l'algoritmo MAC nella richiesta è `HMAC_SHA_384` o `HMAC_SHA_512`. Questa policy utilizza due istruzioni della policy separate, ciascuna con una propria condizione. Se si specificano più algoritmi MAC in una singola istruzione di condizione, la condizione richiede entrambi gli algoritmi anziché l'uno o l'altro.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/testers"
      },
      "Action": [
        "kms:GenerateMac",
        "kms:VerifyMac"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:MacAlgorithm": "HMAC_SHA_384"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/testers"
      },
      "Action": [
        "kms:GenerateMac",
        "kms:VerifyMac"
      ],
    },
  ]
}
```

```

    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "kms:MacAlgorithm": "HMAC_SHA_512"
      }
    }
  }
]
}

```

Consulta anche

- [the section called “kms:EncryptionAlgorithm”](#)
- [kms:SigningAlgorithm](#)

kms:MessageType

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:MessageType	Stringa	A valore singolo	Sign Verify	Policy delle chiavi e policy IAM

La chiave di condizione `kms:MessageType` controlla l'accesso alle operazioni [Sign](#) e [Verify](#) in base al valore del parametro `MessageType` nella richiesta. I valori validi di `MessageType` sono `RAW` e `DIGEST`.

Ad esempio, la seguente istruzione di policy delle chiavi utilizza la chiave di condizione `kms:MessageType` per utilizzare una chiave KMS asimmetrica per firmare un messaggio, ma non un digest di messaggi.

```

{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:Sign",

```

```

"Resource": "*",
"Condition": {
  "StringEquals": {
    "kms:MessageType": "RAW"
  }
}
}

```

Consulta anche

- [the section called “kms:SigningAlgorithm”](#)

kms:MultiRegion

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:MultiRegion	Booleano	A valore singolo	CreateKey Operazioni delle risorse delle chiavi KMS	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per consentire operazioni solo su chiavi di regione singola o solo su [chiavi multiregione](#). Il tasto kms:MultiRegion condition controlla l'accesso alle AWS KMS operazioni sulle chiavi KMS e all'[CreateKey](#) operazione in base al valore della MultiRegion proprietà della chiave KMS. I valori validi sono true (multiregione) e false (singola Regione). Tutte le chiavi KMS hanno una proprietà MultiRegion.

Ad esempio, la seguente istruzione di policy IAM usa la chiave di condizione kms:MultiRegion per consentire ai principali di creare solo chiavi a singola Regione.

```

{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:MultiRegion": false
    }
  }
}

```

```

    }
  }
}
```

kms:MultiRegionKeyType

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:MultiRegionKeyType	Stringa	A valore singolo	CreateKey Operazioni delle risorse delle chiavi KMS	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per consentire operazioni solo su [chiavi multiregione primarie](#) o solo su [chiavi multiregione di replica](#). Il tasto kms:MultiRegionKeyType condition controlla l'accesso alle AWS KMS operazioni sulle chiavi KMS e l'[CreateKey](#) operazione in base alla MultiRegionKeyType proprietà della chiave KMS. I valori validi sono PRIMARY e REPLICA. Solo le chiavi multiregione hanno una proprietà MultiRegionKeyType.

In genere si utilizza la chiave di condizione kms:MultiRegionKeyType in una policy di IAM per controllare l'accesso a più chiavi KMS. Tuttavia, poiché una determinata chiave multiregione può diventare primaria o di replica, è possibile utilizzare questa condizione in una policy chiave per consentire un'operazione solo quando la chiave multiregione specifica è una chiave primaria o di replica.

Ad esempio, la seguente istruzione di policy IAM utilizza la chiave di condizione kms:MultiRegionKeyType per consentire ai principali di pianificare e annullare l'eliminazione delle chiavi solo su chiavi di replica multiregione nell' Account AWS specificato.

```

{
  "Effect": "Allow",
  "Action": [
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "arn:aws:kms:*:111122223333:key/*",
  "Condition": {
```

```

    "StringEquals": {
      "kms:MultiRegionKeyType": "REPLICA"
    }
  }
}

```

Per consentire o negare l'accesso a tutte le chiavi multiregione, è possibile utilizzare entrambi i valori o un valore null con `kms:MultiRegionKeyType`. Tuttavia, la chiave [kms: MultiRegion](#) condition è consigliata a tale scopo.

kms:PrimaryRegion

AWS KMS tasti di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
<code>kms:PrimaryRegion</code>	Stringa (elenco)	A valore singolo	<code>UpdatePrimaryRegion</code>	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per limitare le regioni di destinazione in un'[UpdatePrimaryRegion](#) operazione. Queste sono quelle Regioni AWS che possono ospitare le tue chiavi primarie multiregionali.

Il tasto `kms:PrimaryRegion` condition controlla l'accesso all'[UpdatePrimaryRegion](#) operazione in base al valore del `PrimaryRegion` parametro. Il `PrimaryRegion` parametro specifica la [chiave Regione AWS di replica multiregionale](#) che viene promossa a primaria. Il valore della condizione è costituito da uno o più Regione AWS nomi, ad esempio `or, us-east-1` o da schemi di denominazione delle regioni `ap-southeast-2`, ad esempio `eu-*`

Ad esempio, la seguente istruzione di policy chiave utilizza la chiave di condizione `kms:PrimaryRegion` per consentire ai principali di aggiornare la Regione primaria di una chiave multiregione in una delle quattro Regioni specificate.

```

{
  "Effect": "Allow",
  "Action": "kms:UpdatePrimaryRegion",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Developer"
  },
}

```

```

"Resource": "*",
"Condition": {
  "StringEquals": {
    "kms:PrimaryRegion": [
      "us-east-1",
      "us-west-2",
      "eu-west-3",
      "ap-southeast-2"
    ]
  }
}
}

```

kms:ReEncryptOnSameKey

AWS KMS tasti di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:ReEncryptOnSameKey	Booleano	A valore singolo	ReEncrypt	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per controllare l'accesso all'[ReEncrypt](#) operazione a seconda che la richiesta specifichi una chiave KMS di destinazione che sia la stessa utilizzata per la crittografia originale.

Ad esempio, la seguente istruzione di policy delle chiavi utilizza la chiave di condizione `kms:ReEncryptOnSameKey` per ricrittografare solo quando la chiave KMS di destinazione corrisponde alla chiave utilizzata per la crittografia originale.

```

{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:ReEncrypt*",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:ReEncryptOnSameKey": true
    }
  }
}

```

```

    }
  }
}

```

kms:RequestAlias

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:RequestAlias	Stringa (elenco)	A valore singolo	Operazioni di crittografia DescribeKey GetPublicKey	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per consentire un'operazione solo quando la richiesta utilizza un determinato alias per identificare la chiave KMS. La chiave di condizione `kms:RequestAlias` controlla l'accesso a una chiave KMS utilizzata in un'operazione di crittografia, `GetPublicKey` oppure una `DescribeKey` basata sull'[alias](#) che identifichi la chiave KMS nella richiesta. (Questa condizione politica non ha alcun effetto sull'[GenerateRandom](#) operazione perché l'operazione non utilizza una chiave o un alias KMS.)

Questa condizione supporta il [controllo degli accessi basato sugli attributi](#) (ABAC) in AWS KMS, che consente di controllare l'accesso alle chiavi KMS in base ai tag e agli alias di una chiave KMS. È possibile utilizzare tag e alias per consentire o negare l'accesso a una chiave KMS senza modificare policy o concessioni. Per informazioni dettagliate, vedi [ABAC per AWS KMS](#).

Per specificare l'alias in questa condizione di policy, utilizza un [nome alias](#), ad esempio `alias/project-alpha`, o un modello di nome alias, ad esempio `alias/*test*`. Non è possibile specificare un [ARN alias](#) nel valore di questa chiave di condizione.

Per soddisfare questa condizione, il valore del parametro `KeyId` nella richiesta deve corrispondere a un nome alias o un ARN alias. Se la richiesta utilizza un [identificatore chiave](#), non soddisfa la condizione, anche se identifica la stessa chiave KMS.

Ad esempio, la seguente dichiarazione di politica chiave consente al principale di richiamare l'operazione sulla chiave KMS. [GenerateDataKey](#) Tuttavia questo è consentito solo quando il valore

del parametro `KeyId` nella richiesta è `alias/finance-key` o un ARN `alias` con quel nome `alias`, ad esempio `arn:aws:kms:us-west-2:111122223333:alias/finance-key`.

```
{
  "Sid": "Key policy using a request alias condition",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/developer"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:RequestAlias": "alias/finance-key"
    }
  }
}
```

Non è possibile utilizzare questa chiave di condizione per controllare l'accesso alle operazioni di `alias`, come [CreateAlias](#). [DeleteAlias](#) Per informazioni sul controllo dell'accesso a tutte le operazioni `alias`, consulta [Controllo dell'accesso agli alias](#).

kms:ResourceAliases

AWS KMS tasti di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
<code>kms:ResourceAliases</code>	Stringa (elenco)	Multivalore	Operazioni delle risorse delle chiavi KMS	Solo policy IAM

Utilizza questa chiave di condizione per controllare l'accesso a una chiave KMS in base agli [alias](#) associati alla chiave KMS. L'operazione deve essere un'operazione delle risorse delle chiavi KMS, ossia un'operazione autorizzata per una determinata chiave KMS. Per identificare le operazioni delle risorse delle chiavi KMS, nella [tabella Azioni e risorse](#), cerca un valore della KMS key nella colonna `Resources` per l'operazione.

Questa condizione supporta il controllo degli accessi basato su attributi (ABAC) in AWS KMS. Con ABAC è possibile controllare l'accesso alle chiavi KMS in base ai tag assegnati a una chiave KMS

e agli alias associati a una chiave KMS. È possibile utilizzare tag e alias per consentire o negare l'accesso a una chiave KMS senza modificare policy o concessioni. Per informazioni dettagliate, vedi [ABAC per AWS KMS](#).

Un alias deve essere univoco in una regione Account AWS and, ma questa condizione consente di controllare l'accesso a più chiavi KMS nella stessa regione (utilizzando l'operatore Regioni AWS di StringLike confronto) o a più chiavi KMS in diversi account.

 Note

[La ResourceAliases condizione kms: è efficace solo quando la chiave KMS è conforme agli alias per quota di chiavi KMS](#). Se una chiave KMS supera questa quota, alle entità principali autorizzate a usare la chiave KMS tramite la condizione kms:ResourceAliases viene negato l'accesso alla chiave KMS.

Per specificare l'alias in questa condizione di policy, utilizza un [nome alias](#), ad esempio alias/project-alpha, o un modello di nome alias, ad esempio alias/*test*. Non è possibile specificare un [ARN alias](#) nel valore di questa chiave di condizione. Per soddisfare la condizione, la chiave KMS utilizzata nell'operazione deve avere l'alias specificato. Non importa se o come la chiave KMS viene identificata nella richiesta per l'operazione.

Si tratta di una chiave di condizione multivalore che confronta il set di alias associato a una chiave KMS con il set di alias nella policy. Per determinare il modo in cui questi set vengono confrontati, devi fornire un operatore ForAnyValue o ForAllValues nella condizione di policy. Per dettagli sugli operatori, consulta [Utilizzo di più chiavi e valori](#) nella Guida per l'utente di IAM.

- ForAnyValue: Almeno un alias associato alla chiave KMS deve corrispondere a un alias nella condizione della politica. Sono consentiti altri alias. Se la chiave KMS non dispone di alias, la condizione non viene soddisfatta.
- ForAllValues: Ogni alias associato alla chiave KMS deve corrispondere a un alias nella politica. Questo operatore limita gli alias associati alla chiave KMS a quelli nella condizione della policy. Non richiede alcun alias, ma vieta alias non specificati.

Ad esempio, la seguente dichiarazione di policy IAM consente al principale di richiamare l'[GenerateDataKey](#) operazione su qualsiasi chiave KMS specificata Account AWS associata all'alias. finance-key (Le policy chiave delle chiavi KMS interessate devono inoltre consentire all'account del principale di utilizzarle per questa operazione.) Per indicare che la condizione è soddisfatta

quando uno dei molti alias che potrebbero essere associati alla chiave KMS è `alias/finance-key`, la condizione utilizza il set di operatori `ForAnyValue`.

Poiché la condizione `kms:ResourceAliases` si basa sulla risorsa, non sulla richiesta, una chiamata a `GenerateDataKey` ha esito positivo per qualsiasi chiave KMS associata all'alias `finance-key`, anche se la richiesta utilizza un [ID chiave](#) o un [ARN della chiave](#) per identificare la chiave KMS.

```
{
  "Sid": "AliasBasedIAMPolicy",
  "Effect": "Allow",
  "Action": "kms:GenerateDataKey",
  "Resource": [
    "arn:aws:kms:*:111122223333:key/*",
    "arn:aws:kms:*:444455556666:key/*"
  ],
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:ResourceAliases": "alias/finance-key"
    }
  }
}
```

L'istruzione della policy IAM di esempio seguente consente all'entità principale di abilitare e disabilitare le chiavi KMS, ma solo quando tutti gli alias delle chiavi KMS includono `"Test"`. Questa istruzione di policy utilizza due condizioni. La condizione con l'operatore `ForAllValues` richiede che tutti gli alias associati alla chiave KMS includano `"Test"`. La condizione con l'operatore `ForAnyValue` richiede che la chiave KMS abbia almeno un alias con `"Test"`. Senza la condizione `ForAnyValue`, questa istruzione di policy consentirebbe all'entità principale di utilizzare chiavi KMS prive di alias.

```
{
  "Sid": "AliasBasedIAMPolicy",
  "Effect": "Allow",
  "Action": [
    "kms:EnableKey",
    "kms:DisableKey"
  ],
  "Resource": "arn:aws:kms:*:111122223333:key/*",
  "Condition": {
    "ForAllValues:StringLike": {
      "kms:ResourceAliases": [
```

```

    "alias/*Test*"
  ],
  },
  "ForAnyValue:StringLike": {
    "kms:ResourceAliases": [
      "alias/*Test*"
    ]
  }
}
}
}

```

kms:ReplicaRegion

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:ReplicaRegion	Stringa (elenco)	A valore singolo	Replicate Key	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per limitare i limiti Regioni AWS in cui un principale può replicare una chiave [multiregionale](#). La chiave kms:ReplicaRegion condizionale controlla l'accesso all'[ReplicateKey](#) operazione in base al valore del [ReplicaRegion](#) parametro nella richiesta. Questo parametro specifica la Regione AWS per la nuova [Chiave di replica](#).

Il valore della condizione è costituito da uno o più Regione AWS nomi, ad esempio us-east-1 o ap-southeast-2, o da modelli di nomi, ad esempio eu-*. Per un elenco dei nomi di tali Regioni AWS AWS KMS supporti, consulta [AWS Key Management Service endpoints e quote](#) in. Riferimenti generali di AWS

Ad esempio, la seguente dichiarazione di politica chiave utilizza la chiave kms:ReplicaRegion condition per consentire ai principali di richiamare l'[ReplicateKey](#) operazione solo quando il valore del ReplicaRegion parametro è una delle regioni specificate.

```

{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Administrator"
  },

```

```

{
  "Action": "kms:ReplicateKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ReplicaRegion": [
        "us-east-1",
        "eu-west-3",
        "ap-southeast-2"
      ]
    }
  }
}

```

Questa chiave condizionale controlla l'accesso solo all'[ReplicateKey](#) operazione. Per controllare l'accesso all'[UpdatePrimaryRegion](#) operazione, usa il tasto [kms: PrimaryRegion](#) condition.

kms:RetiringPrincipal

AWS KMS tasti di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:RetiringPrincipal	Stringa (elenco)	A valore singolo	CreateGrant	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per controllare l'accesso all'[CreateGrant](#) operazione in base al valore del [RetiringPrincipal](#) parametro nella richiesta. Ad esempio, puoi creare autorizzazioni per utilizzare una chiave KMS solo quando `RetiringPrincipal` nella richiesta `CreateGrant` corrisponde a `RetiringPrincipal` nell'istruzione della condizione.

Per specificare il principale che va in pensione, utilizza l'Amazon Resource Name (ARN) di AWS un principale. I principali validi includono utenti IAM Account AWS, ruoli IAM, utenti federati e utenti assunti. Per informazioni sulla sintassi ARN per un principale, consulta [IAM ARNs nella IAM User Guide](#).

L'esempio seguente di dichiarazione politica chiave consente a un utente di creare concessioni per la chiave KMS. La chiave `kms:RetiringPrincipal` condizionale limita l'autorizzazione alle `CreateGrant` richieste in cui il principale beneficiario della concessione ritirante è il `LimitedAdminRole`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:RetiringPrincipal": "arn:aws:iam::111122223333:role/LimitedAdminRole"
    }
  }
}
```

Consulta anche

- [kms:GrantConstraintType](#)
- [kms:GrantIsForAWSResource](#)
- [kms:GrantOperations](#)
- [kms:GranteePrincipal](#)

kms:RotationPeriodInDays

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:RotationPeriodInDays	Numerico	A valore singolo	EnableKeyRotation	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per limitare i valori che i committenti possono specificare nel RotationPeriodInDays parametro di una [EnableKeyRotation](#) richiesta.

RotationPeriodInDaysspecifica il numero di giorni tra ogni data di rotazione automatica dei tasti. AWS KMS consente di specificare un periodo di rotazione compreso tra 90 e 2560 giorni, ma è possibile utilizzare il tasto kms:RotationPeriodInDays condition per limitare ulteriormente il periodo di rotazione, ad esempio imporre un periodo di rotazione minimo entro l'intervallo valido.

Ad esempio, la seguente dichiarazione politica chiave utilizza la chiave `kms:RotationPeriodInDays` condition per impedire ai principali di abilitare la rotazione dei tasti se il periodo di rotazione è inferiore o uguale a 180 giorni.

```
{
  "Effect": "Deny",
  "Action": "kms:EnableKeyRotation",
  "Principal": "*",
  "Resource": "*",
  "Condition" : {
    "NumericLessThanEquals" : {
      "kms:RotationPeriodInDays" : "180"
    }
  }
}
```

kms:ScheduleKeyDeletionPendingWindowInDays

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:ScheduleKeyDeletionPendingWindowInDays	Numerico	A valore singolo	ScheduleKeyDeletion	Policy delle chiavi e policy IAM

È possibile utilizzare questa chiave di condizione per limitare i valori che i committenti possono specificare nel `PendingWindowInDays` parametro di una [ScheduleKeyDeletion](#) richiesta.

`PendingWindowInDays` Specifica il numero di giorni che AWS KMS attendono prima di eliminare una chiave. AWS KMS consente di specificare un periodo di attesa compreso tra 7 e 30 giorni, ma è possibile utilizzare la chiave di `kms:ScheduleKeyDeletionPendingWindowInDays` condizione per limitare ulteriormente il periodo di attesa, ad esempio imporre un periodo di attesa minimo entro l'intervallo valido.

Ad esempio, la seguente istruzione della policy della chiave utilizza la chiave di condizione `kms:ScheduleKeyDeletionPendingWindowInDays` per impedire ai principali di pianificare l'eliminazione della chiave se il periodo di attesa è minore o uguale a 21 giorni.

```
{
  "Effect": "Deny",
  "Action": "kms:ScheduleKeyDeletion",
  "Principal": "*",
  "Resource": "*",
  "Condition" : {
    "NumericLessThanEquals" : {
      "kms:ScheduleKeyDeletionPendingWindowInDays" : "21"
    }
  }
}
```

kms:SigningAlgorithm

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:SigningAlgorithm	Stringa	A valore singolo	Sign Verify	Policy delle chiavi e policy IAM

È possibile utilizzare il tasto `kms:SigningAlgorithm` condition per controllare l'accesso alle operazioni di [firma](#) e [verifica](#) in base al valore del [SigningAlgorithm](#) parametro nella richiesta. Questa chiave condizionale non ha effetto sulle operazioni eseguite all'esterno AWS KMS, come la verifica delle firme con la chiave pubblica in una coppia di chiavi KMS asimmetrica all'esterno di AWS KMS.

La seguente policy delle chiavi di esempio consente agli utenti che possono assumere il ruolo `testers` di utilizzare la chiave KMS per firmare i messaggi solo quando l'algoritmo di firma utilizzato per la richiesta è un algoritmo RSASSA_PSS, ad esempio RSASSA_PSS_SHA512.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/testers"
  },
  "Action": "kms:Sign",
  "Resource": "*",
  "Condition": {
    "StringLike": {
```

```

    "kms:SigningAlgorithm": "RSASSA_PSS*"
  }
}
}

```

Consulta anche

- [kms:EncryptionAlgorithm](#)
- [the section called “kms:MacAlgorithm”](#)
- [the section called “kms:MessageType”](#)

kms:ValidTo

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:ValidTo	Time stamp	A valore singolo	ImportKeyMaterial	Policy delle chiavi e policy IAM

Il tasto kms:ValidTo condition controlla l'accesso all'[ImportKeyMaterial](#) operazione in base al valore del [ValidTo](#) parametro nella richiesta, che determina la scadenza del materiale chiave importato. Il valore viene espresso in formato [Unix](#).

Come impostazione predefinita, il parametro ValidTo è obbligatorio in una richiesta ImportKeyMaterial. Tuttavia, se il valore del [ExpirationModel](#) parametro è KEY_MATERIAL_DOES_NOT_EXPIRE, il ValidTo parametro non è valido. Puoi anche usare la chiave [kms: ExpirationModel](#) condition per richiedere il ExpirationModel parametro o un valore di parametro specifico.

La seguente istruzione di policy di esempio consente a un utente di importare il materiale della chiave in una chiave KMS. La chiave di condizione kms:ValidTo limita l'autorizzazione alle richieste ImportKeyMaterial nelle quali il valore ValidTo è minore o uguale a 1546257599.0 (31 dicembre 2018, 23:59:59).

```

{
  "Effect": "Allow",

```



```
"Principal": {
  "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
},
"Action": "kms:ImportKeyMaterial",
"Resource": "*",
"Condition": {
  "NumericLessThanEquals": {
    "kms:ValidTo": "1546257599.0"
  }
}
}
```

Consulta anche

- [kms:ExpirationModel](#)
- [kms:WrappingAlgorithm](#)
- [kms:WrappingKeySpec](#)

kms:ViaService

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:ViaService	Stringa	A valore singolo	Operazioni delle risorse delle chiavi KMS	Policy delle chiavi e policy IAM

La chiave di `kms:ViaService` condizione limita l'uso di una chiave KMS alle richieste provenienti da specifiche Servizi AWS. Questa chiave condizionale si applica solo [alle sessioni di accesso diretto](#). È possibile specificare uno o più servizi in ciascuna chiave di condizione `kms:ViaService`. L'operazione deve essere un'operazione delle risorse delle chiavi KMS, ossia un'operazione autorizzata per una determinata chiave KMS. Per identificare le operazioni delle risorse delle chiavi KMS, nella [tabella Azioni e risorse](#), cerca un valore della KMS key nella colonna Resources per l'operazione.

Ad esempio, la seguente dichiarazione sulla politica chiave utilizza la chiave di `kms:ViaService` condizione per consentire l'utilizzo di una [chiave gestita dal cliente](#) per le azioni specificate solo

quando la richiesta proviene da Amazon EC2 o Amazon RDS nella regione Stati Uniti occidentali (Oregon) per conto di `ExampleRole`

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:CreateGrant",
    "kms:ListGrants",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
        "ec2.us-west-2.amazonaws.com",
        "rds.us-west-2.amazonaws.com"
      ]
    }
  }
}
```

Puoi inoltre utilizzare una chiave di condizione `kms:ViaService` per negare l'autorizzazione a utilizzare una chiave KMS quando la richiesta proviene da determinati servizi. Ad esempio, la seguente istruzione di policy da una policy delle chiavi utilizza una chiave di condizione `kms:ViaService` per evitare che una chiave gestita dal cliente venga utilizzata per le operazioni `Encrypt` quando la richiesta proviene da AWS Lambda per conto di `ExampleRole`.

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:Encrypt"
  ],
```

```
"Resource": "*",
"Condition": {
  "StringEquals": {
    "kms:ViaService": [
      "lambda.us-west-2.amazonaws.com"
    ]
  }
}
```

Important

Quando si utilizza la chiave di condizione `kms:ViaService`, il servizio effettua la richiesta per conto di un principale in Account AWS. Questi principali devono disporre delle autorizzazioni seguenti:

- Autorizzazione a utilizzare la chiave KMS. Il principale deve concedere le autorizzazioni al servizio integrato in modo che il servizio possa utilizzare la chiave gestita dal cliente per conto del principale. Per ulteriori informazioni, consulta [Utilizzo della AWS KMS crittografia con AWS i servizi](#).
- Autorizzazione a utilizzare il servizio integrato. Per informazioni dettagliate su come fornire agli utenti l'accesso a un AWS servizio che si integra con AWS KMS, consulta la documentazione del servizio integrato.

Tutte le [Chiavi gestite da AWS](#) utilizzano una chiave di condizione `kms:ViaService` nel documento della policy delle chiavi. Questa condizione consente alla chiave KMS di essere utilizzata solo per le richieste che provengono dal servizio che ha creato la chiave KMS. Per vedere la politica chiave di una Chiave gestita da AWS, usa l'[GetKeyPolicy](#) operazione.

La chiave di condizione `kms:ViaService` è valida nelle istruzioni IAM e della policy delle chiavi. I servizi specificati devono essere [integrati con AWS KMS](#) e supportare la chiave di condizione `kms:ViaService`.

I servizi che supportano la chiave di condizione **`kms:ViaService`**

La tabella seguente elenca AWS i servizi che sono integrati AWS KMS e supportano l'uso della chiave di `kms:ViaService` condizione nelle chiavi gestite dal cliente. I servizi in questa tabella potrebbero non essere disponibili in tutte le aree. Utilizza il `.amazonaws.com` suffisso del AWS KMS `ViaService` nome in tutte le AWS partizioni.

 Note

Potrebbe essere necessario scorrere orizzontalmente o verticalmente per visualizzare tutti i dati di questa tabella.

Nome del servizio	AWS KMS ViaService nome
Operazioni Amazon AI	aiops. <i>AWS_region</i> .amazonaws.com
AWS App Runner	apprunner. <i>AWS_region</i> .amazonaws.com
AWS AppFabric	appfabric. <i>AWS_region</i> .amazonaws.com
Amazon AppFlow	appflow. <i>AWS_region</i> .amazonaws.com
AWS Application Migration Service	mgn. <i>AWS_region</i> .amazonaws.com
Amazon Athena	athena. <i>AWS_region</i> .amazonaws.com
AWS Audit Manager	auditmanager. <i>AWS_region</i> .amazonaws.com
Amazon Aurora	rds. <i>AWS_region</i> .amazonaws.com
AWS Backup	backup. <i>AWS_region</i> .amazonaws.com
AWS Backup Gateway	backup-gateway. <i>AWS_region</i> .amazonaws.com
Copia del modello Amazon Bedrock	bedrock. <i>AWS_region</i> .amazonaws.com
SDK Amazon Chime	chimevoiceconnector. <i>AWS_region</i> .amazonaws.com
AWS Clean Rooms ML	cleanrooms-ml. <i>AWS_region</i> .amazonaws.com

Nome del servizio	AWS KMS ViaService nome
AWS CodeArtifact	codeartifact. <i>AWS_region</i> .amazonaws.com
CodeGuru Revisore Amazon	codeguru-reviewer. <i>AWS_region</i> .amazonaws.com
Amazon Comprehend	comprehend. <i>AWS_region</i> .amazonaws.com
Amazon Connect	connect. <i>AWS_region</i> .amazonaws.com
Customer Profiles Amazon Connect	profile. <i>AWS_region</i> .amazonaws.com
Amazon Q in Connect	wisdom. <i>AWS_region</i> .amazonaws.com
AWS Database Migration Service (AWS DMS)	dms. <i>AWS_region</i> .amazonaws.com
AWS DeepRacer	deepracer. <i>AWS_region</i> .amazonaws.com
AWS Directory Service	directoryservice. <i>AWS_region</i> .amazonaws.com
Amazon DocumentDB	docdb-elastic. <i>AWS_region</i> .amazonaws.com
Amazon DynamoDB	dynamodb. <i>AWS_region</i> .amazonaws.com
Amazon EC2 Systems Manager (SSM)	ssm. <i>AWS_region</i> .amazonaws.com
Amazon Elastic Block Store (Amazon EBS)	ec2. <i>AWS_region</i> .amazonaws.com (solo EBS)
Amazon Elastic Container Registry (Amazon ECR)	ecr. <i>AWS_region</i> .amazonaws.com

Nome del servizio	AWS KMS ViaService nome
Amazon Elastic File System (Amazon EFS)	elasticfilesystem. <i>AWS_region</i> <i>n</i> .amazonaws.com
Amazon ElastiCache	Includi entrambi ViaService i nomi nel valore della chiave condizionale: • elasticache. <i>AWS_region</i> .amazonaws.com • dax.<i>AWS_region</i> .amazonaws.com
AWS Elemental MediaTailor	mediatailor. <i>AWS_region</i> .amazonaws.com
AWS Risoluzione dell'entità	entityresolution. <i>AWS_region</i> <i>n</i> .amazonaws.com
Amazon EventBridge	events. <i>AWS_region</i> .amazonaws.com
Amazon FinSpace	finspace. <i>AWS_region</i> .amazonaws.com
Amazon Forecast	forecast. <i>AWS_region</i> .amazonaws.com
Amazon FSx	fsx. <i>AWS_region</i> .amazonaws.com
AWS Glue	glue. <i>AWS_region</i> .amazonaws.com
AWS Ground Station	groundstation. <i>AWS_region</i> .amazonaws.com
Amazon GuardDuty	malware-protection. <i>AWS_region</i> <i>n</i> .amazonaws.com
AWS HealthLake	healthlake. <i>AWS_region</i> .amazonaws.com

Nome del servizio	AWS KMS ViaService nome
AWS IoT SiteWise	iotsitewise. <i>AWS_region</i> .amazonaws.com
Amazon Kendra	kendra. <i>AWS_region</i> .amazonaws.com
Amazon Keyspaces (per Apache Cassandra)	cassandra. <i>AWS_region</i> .amazonaws.com
Amazon Kinesis	kinesis. <i>AWS_region</i> .amazonaws.com
Amazon Data Firehose	firehose. <i>AWS_region</i> .amazonaws.com
Amazon Kinesis Video Streams	kinesisvideo. <i>AWS_region</i> .amazonaws.com
AWS Lambda	lambda. <i>AWS_region</i> .amazonaws.com
Amazon Lex	lex. <i>AWS_region</i> .amazonaws.com
AWS License Manager	license-manager. <i>AWS_region</i> .amazonaws.com
Servizio di posizione Amazon	geo. <i>AWS_region</i> .amazonaws.com
Amazon Lookout per le apparecchiature	lookoutequipment. <i>AWS_region</i> .amazonaws.com
Amazon Lookout per le metriche	lookoutmetrics. <i>AWS_region</i> .amazonaws.com
Amazon Lookout per Vision	lookoutvision. <i>AWS_region</i> .amazonaws.com
Amazon Macie	macie. <i>AWS_region</i> .amazonaws.com
Modernizzazione del mainframe AWS	m2. <i>AWS_region</i> .amazonaws.com

Nome del servizio	AWS KMS ViaService nome
Modernizzazione del mainframe AWS Test delle applicazioni	apptest. <i>AWS_region</i> .amazonaws.com
Blockchain gestita da Amazon	managedblockchain. <i>AWS_regio</i> <i>n</i> .amazonaws.com
Amazon Managed Streaming for Apache Kafka (Amazon MSK)	kafka. <i>AWS_region</i> .amazonaws.com
Amazon Managed Workflows for Apache Airflow (MWAA)	airflow. <i>AWS_region</i> .amazonaws.com
Amazon MemoryDB	memorydb. <i>AWS_region</i> .amazonaw s.com
Amazon Monitron	monitron. <i>AWS_region</i> .amazonaw s.com
Amazon MQ	mq. <i>AWS_region</i> .amazonaws.com
Amazon Neptune	rds. <i>AWS_region</i> .amazonaws.com
Amazon Nimble Studio	nimble. <i>AWS_region</i> .amazonaws.com
AWS HealthOmics	omics. <i>AWS_region</i> .amazonaws.com
OpenSearch Servizio Amazon	es. <i>AWS_region</i> .amazonaws.com , aoss. <i>AWS_region</i> .amazonaws.com
Pacchetti OpenSearch personalizzati Amazon	custom-packages. <i>AWS_regio</i> <i>n</i> .amazonaws.com
AWS Proton	proton. <i>AWS_region</i> .amazonaws.com
Database Amazon Quantum Ledger (Amazon QLDB)	qldb. <i>AWS_region</i> .amazonaws.com

Nome del servizio	AWS KMS ViaService nome
Approfondimenti sulle prestazioni di Amazon RDS	<code>rds.AWS_region .amazonaws.com</code>
Amazon Redshift	<code>redshift.AWS_region .amazonaws.com</code>
Editor di query Amazon Redshift V2	<code>sqlworkbench.AWS_region .amazonaws.com</code>
Amazon Redshift Serverless	<code>redshift-serverless.AWS_region .amazonaws.com</code>
Amazon Rekognition	<code>rekognition.AWS_region .amazonaws.com</code>
Amazon Relational Database Service (Amazon RDS)	<code>rds.AWS_region .amazonaws.com</code>
Datastore replicato di Amazon	<code>ards.AWS_region .amazonaws.com</code>
Amazon SageMaker AI	<code>sagemaker.AWS_region .amazonaws.com</code>
Gestione dei segreti AWS	<code>secretsmanager.AWS_region .amazonaws.com</code>
Amazon Security Lake	<code>securitylake.AWS_region .amazonaws.com</code>
Amazon Simple Email Service (Amazon SES)	<code>ses.AWS_region .amazonaws.com</code>
Servizio di notifica semplice Amazon (Amazon Simple Notification Service (Amazon SNS))	<code>sns.AWS_region .amazonaws.com</code>
Amazon Simple Queue Service (Amazon SQS)	<code>sqs.AWS_region .amazonaws.com</code>
Amazon Simple Storage Service (Amazon S3)	<code>s3.AWS_region .amazonaws.com</code>

Nome del servizio	AWS KMS ViaService nome
Tabelle di Amazon S3	s3tables. <i>AWS_region</i> .amazonaws.com
AWS Snowball Edge	importexport. <i>AWS_region</i> .amazonaws.com
AWS Step Functions	states. <i>AWS_region</i> .amazonaws.com
Gateway di archiviazione AWS	storagegateway. <i>AWS_region</i> .amazonaws.com
Strumento di gestione degli incidenti AWS Systems Manager	ssm-incidents. <i>AWS_region</i> .amazonaws.com
Strumento di gestione degli incidenti AWS Systems Manager Contatti	ssm-contacts. <i>AWS_region</i> .amazonaws.com
Amazon Timestream	timestream. <i>AWS_region</i> .amazonaws.com
Amazon Translate	translate. <i>AWS_region</i> .amazonaws.com
Accesso verificato da AWS	verified-access. <i>AWS_region</i> .amazonaws.com
Amazon WorkMail	workmail. <i>AWS_region</i> .amazonaws.com
Amazon WorkSpaces	workspaces. <i>AWS_region</i> .amazonaws.com
Amazon WorkSpaces Thin Client	thinclient. <i>AWS_region</i> .amazonaws.com
Amazon WorkSpaces Web	workspaces-web. <i>AWS_region</i> .amazonaws.com

Nome del servizio	AWS KMS ViaService nome
AWS X-Ray	xray. <i>AWS_region</i> .amazonaws.com

kms:WrappingAlgorithm

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:WrappingAlgorithm	Stringa	A valore singolo	GetParametersForImport	Policy delle chiavi e policy IAM

Questa chiave di condizione controlla l'accesso all'[GetParametersForImport](#) operazione in base al valore del [WrappingAlgorithm](#) parametro nella richiesta. È possibile utilizzare questa condizione per richiedere che i principali utilizzino un determinato algoritmo per crittografare il materiale chiave durante il processo di importazione. Le richieste della chiave pubblica e del token di importazione non riescono se viene specificato un diverso algoritmo di wrapping.

La seguente istruzione di policy chiave di esempio utilizza la chiave di condizione kms:WrappingAlgorithm per fornire all'utente l'autorizzazione a richiamare l'operazione GetParametersForImport, ma gli impedisce di utilizzare l'algoritmo di wrapping RSAES_OAEP_SHA_1. Quando WrappingAlgorithm nella richiesta GetParametersForImport è RSAES_OAEP_SHA_1, l'operazione non riesce.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:GetParametersForImport",
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "kms:WrappingAlgorithm": "RSAES_OAEP_SHA_1"
    }
  }
}
```

```
}
```

Consulta anche

- [kms:ExpirationModel](#)
- [kms:ValidTo](#)
- [kms:WrappingKeySpec](#)

kms:WrappingKeySpec

AWS KMS chiavi di condizione	Tipo di condizioni	Value type (Tipo di valore)	Operazioni API	Tipo di politica
kms:WrappingKeySpec	Stringa	A valore singolo	GetParametersForImport	Policy delle chiavi e policy IAM

Questa chiave di condizione controlla l'accesso all'[GetParametersForImport](#) operazione in base al valore del [WrappingKeySpec](#) parametro nella richiesta. È possibile utilizzare questa condizione per richiedere che i principali utilizzino un determinato tipo di chiave pubblica durante il processo di importazione. Se la richiesta specifica un tipo di chiave diversa, ha esito negativo.

Poiché l'unico valore valido per il valore del parametro `WrappingKeySpec` è `RSA_2048`, impedendo agli utenti di utilizzare questo valore efficacemente, si impedisce loro di utilizzare l'operazione `GetParametersForImport`.

La seguente istruzione di policy di esempio utilizza la chiave di condizione `kms:WrappingAlgorithm` per richiedere che `WrappingKeySpec` nella richiesta sia `RSA_4096`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:GetParametersForImport",
  "Resource": "*",
  "Condition": {
```

```
"StringEquals": {  
  "kms:WrappingKeySpec": "RSA_4096"  
}  
}
```

Consulta anche

- [kms:ExpirationModel](#)
- [kms:ValidTo](#)
- [kms:WrappingAlgorithm](#)

AWS KMS chiavi di condizione per piattaforme attestate

AWS KMS fornisce chiavi di condizione per supportare l'attestazione crittografica per [AWS Nitro Enclaves e NitroTPM](#). AWS Nitro Enclaves è una EC2 funzionalità di Amazon che consente di creare ambienti di elaborazione isolati chiamati [enclavi](#) per proteggere ed elaborare dati altamente sensibili. NitroTPM estende funzionalità di attestazione simili alle istanze. EC2

Quando si richiamano le operazioni [Decrypt](#), [DeriveSharedSecret](#), [GenerateDataKey](#), [GenerateDataKeyPair](#), o [GenerateRandom](#) API con un documento di attestazione firmato, queste APIs crittografano il testo in chiaro nella risposta utilizzando la chiave pubblica del documento di attestazione e restituiscono testo cifrato anziché testo semplice. Questo testo criptato può essere decrittato solo utilizzando la chiave privata nell'enclave. Per ulteriori informazioni, consulta [Supporto per l'attestazione crittografica in AWS KMS](#).

Note

Se non fornisci una politica di chiave quando crei una chiave, ne crea una automaticamente. AWS KMS AWS Questa [policy chiave predefinita](#) concede al proprietario della Account AWS chiave KMS l'accesso completo alla chiave e consente all'account di utilizzare le policy IAM per consentire l'accesso alla chiave. [Questa politica consente tutte le azioni come Decrypt](#). AWS consiglia di applicare principal of [Autorizzazioni con privilegi minimi](#) alle politiche chiave del KMS. Puoi anche limitare l'accesso [modificando l'azione politica chiave del KMS per](#) to. kms : * [NotAction](#) : kms : Decrypt

Le seguenti chiavi di condizione consentono di limitare le autorizzazioni per queste operazioni in base al contenuto del documento di attestazione firmato. Prima di consentire un'operazione, AWS KMS confronta il documento di attestazione con i valori di queste chiavi di condizione. AWS KMS

Chiavi condizionali per Nitro Enclaves

Le seguenti chiavi di condizione sono specifiche dell'attestazione di Nitro Enclaves:

km: 384 RecipientAttestation ImageSha

AWS KMS Chiavi di condizione	Tipo di condizion i	Value type (Tipo di valore)	Operazioni API	Tipo di policy
kms:Recip ientAttes tation:Im ageSha384	Stringa	A valore singolo	Decrypt DeriveSha redSecret Generated ataKey Generated ataKeyPair GenerateR andom	Policy delle chiavi e policy IAM

La chiave kms:RecipientAttestation:ImageSha384 condizionale controlla l'accesso a DecryptDeriveSharedSecret, GenerateDataKeyGenerateDataKeyPair, e GenerateRandom con una chiave KMS quando l'immagine digest del documento di attestazione firmato nella richiesta corrisponde al valore nella chiave di condizione. Il valore ImageSha384 corrisponde a PCR0 nel documento di attestazione. Questa chiave condizionale è efficace solo quando il Recipient parametro nella richiesta specifica un documento di attestazione firmato per un'enclave Nitro. AWS

Questo valore è incluso anche negli [CloudTrail eventi](#) per le richieste alle enclavi Nitro. AWS KMS

Ad esempio, la seguente dichiarazione di policy chiave consente al data-processing ruolo di utilizzare la chiave KMS per [Decrypt](#), [DeriveSharedSecret](#) e le operazioni [GenerateDataKey](#), [GenerateDataKeyPair](#) e [GenerateRandom](#). La chiave di condizione `kms:RecipientAttestation:ImageSha384` consente le operazioni solo quando il valore del digest immagine (PCR0) del documento di attestazione nella richiesta corrisponde al valore del digest nella condizione. Questa chiave condizionale è efficace solo quando il `Recipient` parametro nella richiesta specifica un documento di attestazione firmato per un'enclave Nitro. AWS

Se la richiesta non include un documento di attestazione valido proveniente da un'enclave AWS Nitro, l'autorizzazione viene negata perché questa condizione non è soddisfatta.

```
{
  "Sid" : "Enable enclave data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": [
    "kms:Decrypt",
    "kms:DeriveSharedSecret",
    "kms:GenerateDataKey",
    "kms:GenerateDataKeyPair",
    "kms:GenerateRandom"
  ],
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:ImageSha384":
        "9fedcba8abcdef7abcdef6abcdef5abcdef4abcdef3abcdef2abcdef1abcdef1abcdef0abcdef1abcdef2abcdef3a
    }
  }
}
```

kms: PCR RecipientAttestation <PCR_ID>

AWS KMS Chiavi di condizione	Tipo di condizio ni	Value type (Tipo di valore)	Operazioni API	Tipo di policy
kms:Recip ientAttes tation:PC R<PCR_ID>	Stringa	A valore singolo	Decrypt DeriveSha redSecret Generated ataKey Generated ataKeyPair GenerateR andom	Policy delle chiavi e policy IAM

La chiave `kms:RecipientAttestation:PCR<PCR_ID>` condizionale controlla l'accesso a `DecryptDeriveSharedSecret`, `GenerateDataKey`, `GenerateDataKeyPair`, e `GenerateRandom` con una chiave KMS solo quando la configurazione della piattaforma registra (PCRs) dal documento di attestazione firmato nella richiesta corrisponde a quello nella chiave di condizione. PCRs Questa chiave condizionale è efficace solo quando il `Recipient` parametro nella richiesta specifica un documento di attestazione firmato da un'enclave Nitro. AWS

Questo valore è incluso anche negli [CloudTrail eventi](#) che rappresentano le richieste per le enclavi Nitro. AWS KMS

Per specificare un valore PCR, utilizzare il formato seguente. Concatena l'ID PCR al nome della chiave di condizione. È possibile specificare un ID PCR che identifichi una delle [sei misurazioni dell'enclave](#) o un ID PCR personalizzato definito per un caso d'uso specifico. Il valore PCR deve essere una stringa esadecimale minuscola di un massimo di 96 byte.

```
"kms:RecipientAttestation:PCRPCR_ID": "PCR_value"
```


Ad esempio, la seguente chiave condizionale specifica un valore particolare per PCR1, che corrisponde all'hash del kernel utilizzato per l'enclave e il processo di bootstrap.

```
kms:RecipientAttestation:PCR1:
  "abc1abcdef2abcdef3abcdef4abcdef5abcdef6abcdef7abcdef8abcdef9abcdef8abcdef7abcdef6abcdef5abcde
```

Il seguente esempio di istruzione della policy della chiave consente al ruolo data-processing di utilizzare la chiave KMS per l'operazione [Decrypt](#).

La chiave di kms:RecipientAttestation:PCR condiziona in questa istruzione consente l'operazione solo quando il PCR1 valore nel documento di attestazione firmato nella richiesta corrisponde al valore nella condizione. kms:RecipientAttestation:PCR1 Usa l'operatore di policy StringEqualsIgnoreCase per richiedere un confronto senza distinzione tra maiuscole e minuscole dei valori PCR.

Se la richiesta non include un documento di attestazione, l'autorizzazione viene negata perché questa condizione non è soddisfatta.

```
{
  "Sid" : "Enable enclave data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": "kms:Decrypt",
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:PCR1":
      "abc1de4f2dcf774f6e3b679f62e5f120065b2e408dcea327bd1c9dddaea6664e7af7935581474844767453082c6f1
    }
  }
}
```

Chiavi di condizione per NitroTPM

Le seguenti chiavi di condizione sono specifiche dell'attestazione NitroTPM:

kmsRecipientAttestation:: NitroTPMPCR <PCR_ID>

AWS KMS Chiavi di condizione	Tipo di condiziona i	Value type (Tipo di valore)	Operazioni API	Tipo di policy
kms:RecipientAttestation:NitroTPMPCR<PCR_ID>	Stringa	A valore singolo	Decrypt DeriveSharedSecret GenerateDataKey GenerateDataKeyPair GenerateRandom	Policy delle chiavi e policy IAM

La chiave `kms:RecipientAttestation:NitroTPMPCR<PCR_ID>` condizionale controlla l'accesso a `DecryptDeriveSharedSecret`, `GenerateDataKeyGenerateDataKeyPair`, e `GenerateRandom` con una chiave KMS solo quando la configurazione della piattaforma registra (PCRs) dal documento di attestazione firmato nella richiesta corrisponde a quello nella chiave di condizione. PCRs Questa chiave condizionale è efficace solo quando il `Recipient` parametro nella richiesta specifica un documento di attestazione firmato da NitroTPM.

Questo valore è incluso anche negli [CloudTrail eventi](#) che rappresentano le richieste di NitroTPM.

AWS KMS

Per specificare un valore PCR, utilizzare il formato seguente. Concatena l'ID PCR al nome della chiave di condizione. Il valore PCR deve essere una stringa esadecimale minuscola di un massimo di 96 byte.

```
"kms:RecipientAttestation:NitroTPMPCRPCR_ID": "PCR_value"
```

Ad esempio, la seguente chiave condizionale specifica un valore particolare per: PCR4

```
kms:RecipientAttestation:NitroTPMPCR4:
"abc1abcdef2abcdef3abcdef4abcdef5abcdef6abcdef7abcdef8abcdef9abcdef8abcdef7abcdef6abcdef5abcde
```

Il seguente esempio di istruzione della policy della chiave consente al ruolo data-processing di utilizzare la chiave KMS per l'operazione [Decrypt](#).

La chiave di `kms:RecipientAttestation:NitroTPMPCR` condiziona in questa istruzione consente l'operazione solo quando il PCR4 valore nel documento di attestazione firmato nella richiesta corrisponde al `kms:RecipientAttestation:NitroTPMPCR4` valore nella condizione. Usa l'operatore di policy `StringEqualsIgnoreCase` per richiedere un confronto senza distinzione tra maiuscole e minuscole dei valori PCR.

Se la richiesta non include un documento di attestazione, l'autorizzazione viene negata perché questa condizione non è soddisfatta.

```
{
  "Sid" : "Enable NitroTPM data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": "kms:Decrypt",
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:NitroTPMPCR4":
"abc1de4f2dcf774f6e3b679f62e5f120065b2e408dcea327bd1c9dddaea6664e7af7935581474844767453082c6f1
    }
  }
}
```

Autorizzazioni con privilegi minimi

Poiché le tue chiavi KMS proteggono le informazioni sensibili, ti consigliamo di seguire il principio dell'accesso con privilegi minimi. Delega le autorizzazioni minime richieste per eseguire un'attività quando definisci le tue politiche chiave. Consenti tutte le azioni (`kms:*`) su una policy chiave KMS solo se prevedi di limitare ulteriormente le autorizzazioni con politiche IAM aggiuntive. Se prevedi di gestire le autorizzazioni con le policy IAM, limita chi ha la capacità di creare e collegare le policy IAM ai principi IAM e [monitora](#) le modifiche alle policy.

Se consenti tutte le azioni (kms : *) sia nella policy chiave che nella policy IAM, il principale dispone sia delle autorizzazioni amministrative che di utilizzo per la chiave KMS. Come best practice in materia di sicurezza, consigliamo di delegare queste autorizzazioni solo a responsabili specifici. Puoi farlo nominando esplicitamente il principale nella policy chiave o limitando i principali a cui è associata la policy IAM. Puoi anche usare i [tasti condizionali](#) per limitare le autorizzazioni. Ad esempio, è possibile utilizzare [aws:PrincipalTag](#) per consentire tutte le azioni se il principale che effettua la chiamata API ha il tag specificato nella regola di condizione.

Per aiutarti a capire come vengono valutate le dichiarazioni politiche AWS, consulta la [logica di valutazione delle politiche](#) nella Guida per l'utente IAM. Ti consigliamo di leggere questo argomento prima di scrivere le politiche per ridurre la possibilità che la tua politica abbia effetti indesiderati, come consentire l'accesso a soggetti che non dovrebbero avere accesso.

Tip

Quando testate un'applicazione in un ambiente non di produzione, utilizzate [IAM Access Analyzer](#) per applicare i privilegi minimi alle vostre policy IAM.

Se utilizzi utenti IAM anziché ruoli IAM, ti consigliamo vivamente di abilitare [l'AWS autenticazione a più fattori](#) (MFA) per mitigare la vulnerabilità delle credenziali a lungo termine. È possibile usare l'MFA per le seguenti operazioni:

- Richiedi agli utenti di convalidare le proprie credenziali con MFA prima di eseguire azioni privilegiate, come la pianificazione dell'eliminazione delle chiavi.
- Dividi la proprietà di una password dell'account amministratore e del dispositivo MFA tra individui per implementare l'autorizzazione suddivisa.

Ulteriori informazioni

- [AWS politiche gestite per le funzioni lavorative](#)
- [Techniques for writing least privilege IAM policies](#)

Implementazione di autorizzazioni con privilegio minimo

Quando concedi a un AWS servizio l'autorizzazione a utilizzare una chiave KMS, assicurati che l'autorizzazione sia valida solo per le risorse a cui il servizio deve accedere per tuo conto. Questa

strategia con privilegi minimi aiuta a prevenire l'uso non autorizzato di una chiave KMS quando le richieste vengono passate tra i servizi. AWS

Per implementare una strategia con privilegi minimi, utilizza «consigliamo di utilizzare le chiavi delle condizioni del contesto di AWS KMS crittografia e l'ARN di origine globale o le chiavi di condizione dell'account di origine».

Utilizzo delle chiavi di condizione del contesto di crittografia

Il modo più efficace per implementare le autorizzazioni con privilegi minimi quando si utilizzano le AWS KMS risorse consiste nell'includere le chiavi [kms:EncryptionContext:context-key](#) o le chiavi di [kms:EncryptionContextKeys](#) condizione nella politica che consente ai principali di richiamare operazioni crittografiche. AWS KMS Queste chiavi di condizione sono particolarmente efficaci perché associano l'autorizzazione al [contesto di crittografia](#) che è legato al testo criptato quando la risorsa è crittografata.

[Utilizzate le chiavi relative al contesto di crittografia solo quando l'azione nella dichiarazione politica è CreateGrant o un'operazione crittografica AWS KMS simmetrica che richiede un EncryptionContext parametro, ad esempio operazioni come Decrypt o Decrypt. GenerateDataKey](#) (Per un elenco delle operazioni supportate, consultare [kms:EncryptionContext:context-key](#) o [kms:EncryptionContextKeys](#)). Se si utilizzano queste chiavi di condizione per consentire altre operazioni, ad esempio, l'autorizzazione verrà negata [DescribeKey](#).

Impostare il valore sul contesto di crittografia utilizzato dal servizio quando crittografa la risorsa. Queste informazioni sono generalmente disponibili nel capitolo Sicurezza della documentazione del servizio. Ad esempio, il [contesto di crittografia per AWS Proton](#) identifica la risorsa AWS Proton e il modello associato. Il [contesto di crittografia di Gestione dei segreti AWS](#) identifica il segreto e la sua versione. Il [contesto di crittografia per Amazon Location](#) identifica la localizzazione o la raccolta.

Il seguente esempio di istruzione della chiave delle policy permette ad Amazon Location Service di creare concessioni per conto di utenti autorizzati. Questa dichiarazione politica limita l'autorizzazione utilizzando le chiavi [kms: ViaService](#), [kms: CallerAccount](#) e `kms:EncryptionContext:context-key` condition per collegare l'autorizzazione a una particolare risorsa tracker.

```
{
  "Sid": "Allow Amazon Location to create grants on behalf of authorized users",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/LocationTeam"
```

```
},
"Action": "kms:CreateGrant",
"Resource": "*",
"Condition": {
  "StringEquals": {
    "kms:ViaService": "geo.us-west-2.amazonaws.com",
    "kms:CallerAccount": "111122223333",
    "kms:EncryptionContext:aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:tracker/
SAMPLE-Tracker"
  }
}
```

Utilizzo delle chiavi di condizione **aws:SourceArn** o **aws:SourceAccount**

Quando il principale in un'istruzione della policy della chiave è un [principale del servizio AWS](#), consigliamo vivamente di utilizzare le chiavi di condizione globale [aws:SourceArn](#) o [aws:SourceAccount](#) oltre alla chiave di condizione `kms:EncryptionContext:context-key`. L'ARN e i valori dell'account sono inclusi nel contesto di autorizzazione solo quando arriva una richiesta AWS KMS da un altro AWS servizio. Questa combinazione di condizioni implementa autorizzazioni meno privilegiate ed evita un potenziale [scenario "confused deputy"](#). I service principal non vengono in genere utilizzati come principali in una policy chiave, ma alcuni AWS servizi, ad esempio AWS CloudTrail, lo richiedono.

Per utilizzare le chiavi di condizione globali `aws:SourceArn` o `aws:SourceAccount`, impostare il valore sul nome della risorsa Amazon (ARN) o sull'account della risorsa crittografata. Ad esempio, nell'istruzione di una policy della chiave che fornisce l'autorizzazione AWS CloudTrail per crittografare un percorso, impostare sull'ARN del percorso il valore di `aws:SourceArn`. Quando possibile, utilizzare `aws:SourceArn`, che è più specifico. Impostare il valore sull'ARN o un modello ARN con caratteri jolly. Se non si conosce l'ARN della risorsa, utilizzare `aws:SourceAccount`.

Note

Se l'ARN di una risorsa include caratteri non consentiti in una politica AWS KMS chiave, non è possibile utilizzare l'ARN di tale risorsa nel valore della chiave di condizione. `aws:SourceArn` Devi invece utilizzare la chiave di condizione `aws:SourceAccount`. Per maggiori informazioni sulle regole delineate nel documento delle policy delle chiavi, consulta la sezione [Formato della policy della chiave](#).

Nell'esempio seguente di policy della chiave, il principale che ottiene le autorizzazioni è il principale del servizio AWS CloudTrail, `cloudtrail.amazonaws.com`. Per implementare il privilegio minimo, questa policy utilizza le chiavi di condizione `aws:SourceArn` e `kms:EncryptionContext:context-key`. L'informativa sulla politica consente di CloudTrail utilizzare la chiave KMS per [generare la chiave dati](#) utilizzata per crittografare una traccia. Le condizioni `aws:SourceArn` e `kms:EncryptionContext:context-key` sono valutate in modo indipendente. Qualsiasi richiesta di utilizzare la chiave KMS per l'operazione specificata deve soddisfare entrambe le condizioni.

Per limitare l'autorizzazione del servizio al percorso `finance` nell'account di esempio (111122223333) e nella Regione `us-west-2`, questa istruzione di policy definisce la condizione della chiave `aws:SourceArn` per l'ARN di un particolare percorso. L'istruzione `condition` utilizza l'[ArnEquals](#) operatore per garantire che ogni elemento dell'ARN venga valutato indipendentemente durante la corrispondenza. L'esempio utilizza anche la chiave di condizione `kms:EncryptionContext:context-key` per limitare l'autorizzazione ai percorsi in un determinato account e Regione.

Prima di utilizzare questa policy della chiave, è necessario sostituire l'ID dell'account, la Regione e il nome del percorso di esempio con valori validi riferiti al proprio account.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudTrailToEncryptLogs",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
      "Action": "kms:GenerateDataKey",
      "Resource": "*",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": [
            "arn:aws:cloudtrail:us-west-2:111122223333:trail/finance"
          ]
        }
      },
      "StringLike": {
```

```
    "kms:EncryptionContext:aws:cloudtrail:arn": [  
        "arn:aws:cloudtrail:*:111122223333:trail/*"  
    ]  
  }  
}  
]  
}
```

ABAC per AWS KMS

Il controllo degli accessi basato sugli attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. AWS KMS supporta ABAC consentendovi di controllare l'accesso alle chiavi gestite dai clienti in base ai tag e agli alias associati alle chiavi KMS. Le chiavi di condizione dei tag e degli alias che consentono ad ABAC di AWS KMS fornire un modo potente e flessibile per autorizzare i mandanti a utilizzare le chiavi KMS senza modificare le politiche o gestire le sovvenzioni. Ma occorre usare queste funzionalità con cura in modo che ai principali non sia inavvertitamente autorizzato o negato l'accesso.

Se si utilizza ABAC, tenere presente che l'autorizzazione per gestire tag e alias è ora un'autorizzazione di controllo di accesso. Assicurarsi di conoscere i tag e gli alias esistenti in tutte le chiavi KMS prima di distribuire una policy che dipende da tag o alias. Prendere ragionevoli precauzioni quando si aggiungono, eliminano e aggiornano gli alias e quando si taggano e si rimuovono i tag delle chiavi. Assegna le autorizzazioni per gestire tag e alias solo alle entità che ne hanno bisogno e limita i tag e gli alias che possono gestire.

Note

Quando usi ABAC for AWS KMS, fai attenzione a non concedere ai principali il permesso di gestire tag e alias. La modifica di un tag o di un alias potrebbe consentire o negare l'autorizzazione a una chiave KMS. Gli amministratori chiave che non dispongono dell'autorizzazione per modificare le policy chiave o creare concessioni possono controllare l'accesso alle chiavi KMS se dispongono dell'autorizzazione per gestire tag o alias. È possibile che occorranza fino a cinque minuti affinché le modifiche a tag e alias diventano effettive per l'autorizzazione delle chiavi KMS. Le modifiche recenti potrebbero essere visibili nelle operazioni API prima che influiscano sull'autorizzazione. Per controllare l'accesso a una chiave KMS in base al relativo alias, è necessario utilizzare una chiave di condizione. Non è possibile utilizzare un alias per rappresentare una chiave

KMS nell'elemento Resource di un'istruzione di policy. Quando viene visualizzato un alias nell'elemento Resource, l'istruzione di policy si applica all'alias e non alla chiave KMS associata.

Ulteriori informazioni

- Per dettagli sul AWS KMS supporto per ABAC, inclusi esempi, consulta e. [Usa gli alias per controllare l'accesso alle chiavi KMS](#) [Usa i tag per controllare l'accesso alle chiavi KMS](#)
- Per informazioni più generali sull'uso dei tag per controllare l'accesso alle AWS risorse, vedi A [cosa serve ABAC? AWS e Controllo dell'accesso alle AWS risorse utilizzando i tag delle risorse](#) nella Guida per l'utente IAM.

Chiavi di condizione ABAC per AWS KMS

Per autorizzare l'accesso alle chiavi del servizio di gestione delle chiavi in base ai relativi tag e alias, utilizzare le seguenti chiavi di condizione in una policy chiave o in una policy IAM.

Chiavi di condizione ABAC	Description	Tipo di politica	AWS KMS operazioni
leggi: ResourceTag	Il tag (chiave e valore) sulla chiave KMS corrisponde al tag (chiave e valore) o al modello di tag nella policy	Solo policy IAM	Operazioni delle risorse delle chiavi KMS ²
aws:RequestTag//tag-key	Il tag (chiave e valore) nella richiesta corrisponde al tag (chiave e valore) o al modello di tag nella policy	Policy chiave e policy IAM ¹	TagResource , UntagResource
aws: TagKeys	Le chiavi tag nella richiesta corrispon	Policy chiave e policy IAM ¹	TagResource , UntagResource

Chiavi di condizione ABAC	Description	Tipo di politica	AWS KMS operazioni
	dono alle chiavi tag nella policy		
km: ResourceAliases	Gli alias associati alla chiave KMS corrispondono agli alias o ai modelli di alias nella policy	Solo policy IAM	Operazioni delle risorse delle chiavi KMS ²
km: RequestAlias	L'alias che rappresenta la chiave KMS nella richiesta corrisponde agli alias o ai modelli di alias nella policy.	Policy chiave e policy IAM ¹	Operazioni crittografiche , DescribeKey , GetPublicKey

¹Qualsiasi chiave di condizione che può essere utilizzata in una policy chiave può essere utilizzata anche in una policy IAM, ma solo se [la policy chiave lo consente](#).

²Un'operazione delle risorse delle chiavi KMS è un'operazione autorizzata per una determinata chiave KMS. Per identificare le operazioni delle risorse delle chiavi KMS, nella [tabella delle autorizzazioni AWS KMS](#), cerca un valore della chiave KMS nella colonna Resources per l'operazione.

Ad esempio, è possibile utilizzare queste chiavi di condizione per creare le seguenti policy.

- Una policy IAM con `kms:ResourceAliases` che consente di utilizzare le chiavi KMS con un particolare alias o modello di alias. Si tratta di una politica leggermente diversa dalle policy che si basano sui tag: sebbene sia possibile utilizzare modelli di alias in una policy, ogni alias deve essere unico in una Account AWS regione. Ciò consente di applicare una politica a un set selezionato di chiavi KMS senza elencare la chiave ARNs delle chiavi KMS nell'informativa sulla politica. Per aggiungere o rimuovere chiavi KMS dal set, modificare l'alias della chiave KMS.
- Una policy chiave con `kms:RequestAlias` che consente ai principali di utilizzare una chiave KMS in un'operazione Encrypt, ma solo quando la richiesta Encrypt utilizza tale alias per identificare la chiave KMS.

- Una policy IAM con `aws:ResourceTag/tag-key` che nega l'autorizzazione a utilizzare le chiavi KMS con una chiave di tag e un valore di tag specifici. Ciò consente di applicare una politica a un set selezionato di chiavi KMS senza elencare la chiave ARNs delle chiavi KMS nell'informativa sulla politica. Per aggiungere o rimuovere le chiavi del servizio di gestione delle chiavi dal set, taggare o rimuovere la chiave KMS.
- Una policy IAM con `aws:RequestTag/tag-key` che consente ai gruppi di eliminare solo i tag della chiave KMS `"Purpose"="Test"`.
- Una policy IAM con `aws:TagKeys` che nega l'autorizzazione ad aggiungere o eliminare un tag a una chiave KMS con una chiave tag `Restricted`.

ABAC rende la gestione degli accessi flessibile e scalabile. Ad esempio, puoi utilizzare la chiave di condizione `aws:ResourceTag/tag-key` per creare una policy IAM che consente ai principali di utilizzare una chiave KMS per le operazioni specificate solo quando la chiave KMS ha un tag `Purpose=Test`. La policy si applica a tutte le chiavi KMS in tutte le Regioni dell' Account AWS.

Quando è collegata a un utente o a un ruolo, la policy IAM seguente consente alle entità principali di utilizzare tutte le chiavi KMS esistenti con un tag `Purpose=Test` per le operazioni specificate. Per fornire questo accesso a chiavi KMS nuove o esistenti, non è necessario modificare le policy. È sufficiente collegare il tag `Purpose=Test` alle chiavi KMS. Allo stesso modo, per rimuovere questo accesso dalle chiavi KMS con un tag `Purpose=Test`, modifica o elimina il tag.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AliasBasedIAMPolicy",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Purpose": "Test"
        }
      }
    }
  ]
}
```

```
}  
  }  
    }  
      ]  
        }
```

Tuttavia, se si utilizza questa funzione, fare attenzione nella gestione di tag e alias. L'aggiunta, la modifica o l'eliminazione di un tag o di un alias può inavvertitamente consentire o negare l'accesso a una chiave KMS. Gli amministratori chiave che non dispongono dell'autorizzazione per modificare le policy chiave o creare concessioni possono controllare l'accesso alle chiavi KMS se dispongono dell'autorizzazione per gestire tag e alias. Per mitigare questo rischio, considera di [limitare le autorizzazioni per la gestione di tag e alias](#). Ad esempio, potrebbe essere necessario consentire solo ai principali di gestire la gestione dei tag `Purpose=Test`. Per informazioni dettagliate, consulta [Usa gli alias per controllare l'accesso alle chiavi KMS](#) e [Usa i tag per controllare l'accesso alle chiavi KMS](#).

Tag o alias?

AWS KMS supporta ABAC con tag e alias. Entrambe le opzioni offrono una strategia di controllo degli accessi flessibile e scalabile, ma sono leggermente diverse l'una dall'altra.

Potresti decidere di utilizzare tag o utilizzare alias in base ai tuoi modelli di utilizzo particolari AWS . Ad esempio, se sono già state concesse autorizzazioni di assegnazione di tag alla maggior parte degli amministratori, potrebbe essere più semplice controllare una strategia di autorizzazione basata sugli alias. Oppure, se stai per raggiungere la quota di [alias per chiave KMS](#), potresti preferire una strategia di autorizzazione basata sui tag.

I seguenti vantaggi sono di interesse generale.

Vantaggi del controllo degli accessi basato su tag

- Stesso meccanismo di autorizzazione per diversi tipi di AWS risorse.

Puoi utilizzare lo stesso tag o codice tag per controllare l'accesso a più tipi di risorse, ad esempio un cluster Amazon Relational Database Service (Amazon RDS), un volume Amazon Elastic Block Store (Amazon EBS) e una chiave KMS. Questa funzione consente diversi modelli di autorizzazione più flessibili rispetto ai tradizionali controlli di accesso basati sui ruoli.

- Autorizzare l'accesso a un gruppo di chiavi KMS.

È possibile utilizzare i tag per gestire l'accesso a un gruppo di chiavi KMS nello stesso Account AWS e nella stessa Regione. Assegnare lo stesso tag o chiave di tag alle chiavi KMS scelte. Quindi crea una semplice dichiarazione easy-to-maintain politica basata sul tag o sulla chiave del tag. Per aggiungere o rimuovere una chiave KMS dal gruppo di autorizzazioni, aggiungere o rimuovere il tag. Non è necessario modificare la policy.

Vantaggi del controllo degli accessi basato su alias

- Autorizza l'accesso alle operazioni di crittografia in base agli alias.

La maggior parte delle condizioni delle policy basate sulla richiesta per gli attributi, incluso [aws:RequestTag/tag-key](#), influiscono solo sulle operazioni che aggiungono, modificano o eliminano l'attributo. Ma la chiave [kms:RequestAlias](#) condition controlla l'accesso alle operazioni crittografiche in base all'alias utilizzato per identificare la chiave KMS nella richiesta. Ad esempio, è possibile concedere a un'entità principale l'autorizzazione per utilizzare una chiave KMS in u'operazione Encrypt ma solo quando il valore del parametro KeyId è alias/restricted-key-1. Per soddisfare questa condizione, è necessario quanto segue:

- La chiave KMS deve essere associata a tale alias.
- La richiesta deve utilizzare l'alias per identificare la chiave KMS.
- Il principale deve disporre dell'autorizzazione per utilizzare la chiave KMS soggetta alla condizione `kms:RequestAlias`.

Ciò è particolarmente utile se le applicazioni utilizzano comunemente nomi alias o alias ARNs per fare riferimento alle chiavi KMS.

- Fornisci autorizzazioni molto limitate.

Un alias deve essere univoco in una regione and. Account AWS Di conseguenza, concedere ai principali l'accesso a una chiave KMS basata su un alias può essere molto più restrittivo rispetto a concedere loro l'accesso basato su un tag. Diversamente dagli alias, i tag possono essere assegnati a più chiavi KMS nello stesso account e nella stessa Regione. Se si sceglie, è possibile utilizzare un modello alias, ad esempio `alias/test*`, per consentire agli entità principali di accedere a un gruppo di chiavi KMS nello stesso account e nella stessa Regione. Tuttavia, l'autorizzazione o la negazione dell'accesso a un alias specifico consente un controllo molto rigoroso sulle chiavi KMS.

Risoluzione dei problemi ABAC per AWS KMS

Controllare l'accesso alle chiavi KMS in base ai tag e agli alias è comodo e potente. Tuttavia, è incline a alcuni errori prevedibili che vorrai prevenire.

Accesso modificato a causa della modifica dei tag

Se un tag viene eliminato o il relativo valore viene modificato, ai principali che hanno accesso a una chiave KMS basata solo su tale tag verrà negato l'accesso alla chiave KMS. Ciò può verificarsi anche quando un tag incluso in un'istruzione di policy di negazione viene aggiunto a una chiave KMS. L'aggiunta di un tag relativo alla policy a una chiave KMS può consentire l'accesso a principali a cui è necessario negare l'accesso a una chiave KMS.

Si supponga, ad esempio, che un principale abbia accesso a una chiave KMS in base al tag `Project=Alpha`, ad esempio l'autorizzazione fornita dalla seguente istruzione della policy IAM di esempio.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyWithResourceTag",
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:ap-southeast-1:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "Alpha"
        }
      }
    }
  ]
}
```

Se il tag viene eliminato dalla chiave KMS o il valore del tag viene modificato, l'entità principale non dispone più dell'autorizzazione per utilizzare la chiave KMS per le operazioni specificate. [Ciò potrebbe diventare evidente quando il responsabile tenta di leggere o scrivere dati in un AWS servizio che utilizza una chiave gestita dal cliente. Per tracciare la modifica del tag, controlla CloudTrail i log TagResourceo UntagResource le immissioni.](#)

Per ripristinare l'accesso senza aggiornare la policy, modifica i tag sulla chiave KMS. Questa azione ha un impatto minimo diverso dal breve periodo in cui sta entrando in vigore AWS KMS. Per evitare un errore come questo, dai le autorizzazioni per l'assegnazione e l'eliminazione di tag solo ai principali che ne hanno bisogno e [limita le autorizzazioni per l'assegnazione di tag](#) ai tag che devono gestire. Prima di modificare un tag, cerca le policy per rilevare l'accesso che dipende dal tag e ottenere le chiavi KMS in tutte le Regioni che dispongono del tag. Potresti prendere in considerazione la creazione di un CloudWatch allarme Amazon quando vengono modificati determinati tag.

Modifica dell'accesso a causa della modifica degli alias

Se un alias viene eliminato o associato a una chiave KMS diversa, ai principali che hanno accesso alla chiave KMS basata solo su tale alias verrà negato l'accesso alla chiave KMS. Ciò può verificarsi anche quando un alias associato a una chiave KMS è incluso in un'istruzione della policy di negazione. L'aggiunta di un alias relativo alla policy a una chiave KMS può inoltre consentire l'accesso a principali a cui è necessario negare l'accesso a una chiave KMS.

Ad esempio, la seguente dichiarazione politica IAM utilizza la chiave [kms: ResourceAliases](#) condition per consentire l'accesso alle chiavi KMS in diverse regioni dell'account con uno qualsiasi degli alias specificati.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AliasBasedIAMPolicy",
      "Effect": "Allow",
      "Action": [
        "kms:List*",
        "kms:Describe*",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
    }
  ]
}
```

```
"Condition": {
  "ForAnyValue:StringEquals": {
    "kms:ResourceAliases": [
      "alias/ProjectAlpha",
      "alias/ProjectAlpha_Test",
      "alias/ProjectAlpha_Dev"
    ]
  }
}
```

Per tracciare la modifica dell'alias, esamina i CloudTrail log e le [CreateAlias](#) immissioni.

[UpdateAliasDeleteAlias](#)

Per ripristinare l'accesso senza aggiornare la policy, modifica gli alias associati alla chiave KMS. Poiché ogni alias può essere associato a una sola chiave KMS in un account e in una Regione, la gestione degli alias è un po' più difficile della gestione dei tag. Il ripristino dell'accesso ad alcune entità principali a una chiave KMS può negare l'accesso alla stessa o ad altre entità principali a una chiave KMS diversa.

Per evitare questo errore, assegna autorizzazioni di gestione degli alias solo alle entità principali che ne hanno bisogno e [limita le autorizzazioni per la gestione degli alias](#) agli alias che devono gestire. Prima di aggiornare o eliminare un alias, cerca le policy per rilevare l'accesso che dipende dall'alias e trova le chiavi KMS in tutte le Regioni associate all'alias.

Accesso negato a causa di quota alias

Gli utenti autorizzati a utilizzare una chiave KMS entro una ResourceAliases condizione [kms:](#) riceveranno un'AccessDenied eccezione se la chiave KMS supera gli [alias predefiniti per quota di chiavi KMS per quell'account e quella regione](#).

Per ripristinare l'accesso, elimina gli alias associati alla chiave KMS in modo da rispettare la quota. In alternativa, utilizza un meccanismo alternativo per consentire agli utenti di accedere alla chiave KMS.

Modifica dell'autorizzazione ritardata

Le modifiche apportate a tag e alias possono richiedere fino a cinque minuti per influenzare l'autorizzazione delle chiavi KMS. Di conseguenza, una modifica di tag o alias potrebbe riflettersi nelle risposte delle operazioni API prima che influiscano sull'autorizzazione. È probabile che questo

ritardo sia più lungo dell'eventuale breve ritardo di coerenza che influisce sulla maggior parte delle operazioni. AWS KMS

Ad esempio, potrebbe esserci una policy IAM che consente a determinate entità principali di utilizzare una chiave KMS con un tag "Purpose"="Test". Quindi aggiungi il tag "Purpose"="Test" su una chiave KMS. Sebbene l'[TagResource](#) operazione sia stata completata e la [ListResourceTags](#) risposta confermi che il tag è stato assegnato alla chiave KMS, i responsabili potrebbero non avere accesso alla chiave KMS per un massimo di cinque minuti.

Per evitare errori, inserisci questo ritardo previsto nel tuo codice.

Richieste non riuscite a causa di aggiornamenti alias

Quando aggiorni un alias, associ un alias esistente a una chiave KMS diversa.

La [decrittografia](#) e [ReEncrypt](#) le richieste che specificano il [nome alias](#) o l'alias [ARN](#) potrebbero non riuscire perché l'alias è ora associato a una chiave KMS che non crittografava il testo cifrato. Questa situazione in genere restituisce `IncorrectKeyException` o `NotFoundException`. O se la richiesta non ha un parametro `KeyId` o `DestinationKeyId`, l'operazione potrebbe avere esito negativo con l'eccezione `AccessDenied` perché il chiamante non ha più accesso alla chiave KMS che ha crittografato il testo cifrato.

È possibile tracciare la modifica esaminando i log e le voci di registro. CloudTrail [CreateAliasUpdateAliasDeleteAlias](#) È inoltre possibile utilizzare il valore del `LastUpdatedDate` campo nella [ListAliases](#) risposta per rilevare una modifica.

Ad esempio, la seguente risposta di [ListAliases](#) esempio mostra che l'`ProjectAlpha_Test` alias nella `kms:ResourceAliases` condizione è stato aggiornato. Di conseguenza, le entità principali che hanno un accesso basato sugli alias perdono l'accesso alla chiave KMS associata in precedenza. Al contrario, hanno accesso alla nuova chiave KMS associata.

```
$ aws kms list-aliases --query 'Aliases[?starts_with(AliasName, `alias/ProjectAlpha`)]'

{
  "Aliases": [
    {
      "AliasName": "alias/ProjectAlpha_Test",
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ProjectAlpha_Test",
      "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
      "CreationDate": 1566518783.394,
      "LastUpdatedDate": 1605308931.903
    }
  ]
}
```

```
    },  
    {  
      "AliasName": "alias/ProjectAlpha_Restricted",  
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/  
ProjectAlpha_Restricted",  
      "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
      "CreationDate": 1553410800.010,  
      "LastUpdatedDate": 1553410800.010  
    }  
  ]  
}
```

Non è semplice rimediare a questa modifica. È possibile aggiornare nuovamente l'alias per associarlo alla chiave KMS originale. Tuttavia, prima di agire, è necessario considerare l'effetto di tale modifica sulla chiave KMS attualmente associata. Se le entità utilizzavano quest'ultima chiave KMS nelle operazioni di crittografia, potrebbe essere necessario continuare ad accedervi. In questo caso, è possibile aggiornare la policy per assicurarsi che le entità principali dispongano dell'autorizzazione per utilizzare entrambe le chiavi KMS.

È possibile evitare un errore come questo: prima di aggiornare un alias, cerca le policy per rilevare l'accesso che dipende dall'alias. Quindi ottieni le chiavi KMS in tutte le Regioni associate all'alias. Assegna autorizzazioni di gestione degli alias solo alle entità principali che ne hanno bisogno e [limita le autorizzazioni per la gestione degli alias](#) agli alias che devono gestire.

RBAC per AWS KMS

Il controllo degli accessi basato sui ruoli (RBAC) è una strategia di autorizzazione che fornisce agli utenti solo le autorizzazioni necessarie per svolgere le proprie mansioni lavorative e nient'altro. AWS KMS [supporta RBAC permettendo di controllare l'accesso alle chiavi specificando autorizzazioni granulari sull'utilizzo delle chiavi all'interno delle policy chiave](#). Le politiche chiave specificano una risorsa, un'azione, un effetto, una condizione principale e facoltativa per concedere l'accesso alle chiavi.

Per implementare RBAC in AWS KMS, consigliamo di separare le autorizzazioni per gli utenti chiave e gli amministratori chiave.

Key users

Il seguente esempio di policy chiave consente al ruolo `ExampleUserRole` IAM di utilizzare la chiave KMS.

```
{
    "Sid": "Allow use of the key",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws: iam::111122223333:role/ExampleUserRole"
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

I tuoi utenti principali potrebbero aver bisogno di meno autorizzazioni rispetto all'utente in questo esempio. Assegna solo le autorizzazioni di cui l'utente ha bisogno. Utilizza le seguenti domande per perfezionare ulteriormente le autorizzazioni.

- Quali presidi IAM (ruoli o utenti) devono accedere alla chiave?
- Quali azioni deve eseguire ogni principale con la chiave? Ad esempio, il preside necessita solo delle autorizzazioni Encrypt and Sign?
- L'utente è un essere umano o un AWS servizio? Se si tratta di un AWS servizio, puoi utilizzare la [chiave di condizione](#) per limitare l'utilizzo della chiave a un AWS servizio specifico.

Key administrators

Il seguente esempio di policy chiave consente al ruolo ExampleAdminRole IAM di amministrare la chiave KMS.

```
{
    "Sid": "Allow access for Key Administrators",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws: iam::111122223333:role/ExampleAdminRole"
    },
    "Action": [
        "kms:Create*",
        "kms:Describe*",

```

```
        "kms:Enable*",
        "kms:List*",
        "kms:Put*",
        "kms:Update*",
        "kms:Revoke*",
        "kms:Disable*",
        "kms:Get*",
        "kms:Delete*",
        "kms:TagResource",
        "kms:UntagResource",
        "kms:ScheduleKeyDeletion",
        "kms:CancelKeyDeletion"
    ],
    "Resource": "*"
}
```

In questo esempio, gli amministratori chiave potrebbero aver bisogno di meno autorizzazioni rispetto all'amministratore. Assegna solo le autorizzazioni di cui hanno bisogno i tuoi amministratori chiave.

Concedi agli utenti solo le autorizzazioni necessarie per svolgere i loro ruoli. Le autorizzazioni di un utente possono variare a seconda che la chiave venga utilizzata in ambienti di test o di produzione. Se utilizzi autorizzazioni meno restrittive in determinati ambienti non di produzione, implementa un processo per testare le politiche prima che vengano rilasciate in produzione.

Ulteriori informazioni

- [Identità IAM \(utenti, gruppi di utenti e ruoli\)](#)
- [Tipi di controllo degli accessi](#)

Autorizzazione per gli utenti in altri account di utilizzare una chiave KMS

Puoi consentire a utenti o ruoli diversi di Account AWS utilizzare una chiave KMS nel tuo account. L'accesso tra account richiede l'autorizzazione nella policy chiave della chiave dKMS e in una policy IAM nell'account dell'utente esterno.

L'autorizzazione tra account è valida solo per le seguenti operazioni:

- [Operazioni di crittografia](#)
- [CreateGrant](#)
- [DescribeKey](#)
- [GetKeyRotationStatus](#)
- [GetPublicKey](#)
- [ListGrants](#)
- [RetireGrant](#)
- [RevokeGrant](#)

Se concedi a un utente in un account diverso l'autorizzazione per altre operazioni, tali autorizzazioni non hanno effetto. Ad esempio, se concedi a un account principale [kms: ListKeys](#) permission in una policy IAM o [kms: ScheduleKeyDeletion permission on una chiave KMS](#) in una policy chiave, i tentativi dell'utente di richiamare tali operazioni sulle tue risorse continueranno a fallire.

Per i dettagli sull'utilizzo delle chiavi KMS in diversi account per AWS KMS le operazioni, consulta la colonna Utilizzo tra account nella pagina e. [AWS KMS autorizzazioni Utilizzo delle chiavi KMS in altri account](#) C'è anche una sezione Utilizzo tra account in ogni descrizione API nella [Documentazione di riferimento dell'API AWS Key Management Service](#).

Warning

Fai attenzione a concedere ai principali le autorizzazioni per utilizzare le chiavi KMS. Quando possibile, segui il principio del privilegio minimo. Offri agli utenti l'accesso solo alle chiavi KMS necessarie solo per le operazioni che richiedono.

Inoltre, presta cautela nell'utilizzare qualsiasi chiave KMS sconosciuta, in particolare una chiave KMS in un account diverso. Gli utenti malintenzionati potrebbero concederti le autorizzazioni per utilizzare la loro chiave KMS e ottenere informazioni su di te o sul tuo account.

Per informazioni sull'utilizzo delle policy per proteggere le risorse dell'account, consulta [Best practice per le policy IAM](#).

Per concedere l'autorizzazione a utilizzare una chiave KMS a utenti e ruoli in un altro account, è necessario utilizzare due diversi tipi di policy:

- La policy chiave per la chiave KMS deve fornire all'account esterno (o utenti e ruoli nell'account esterno) l'autorizzazione per utilizzare la chiave KMS. La policy delle chiavi si trova nell'account proprietario della chiave KMS.
- Le policy IAM nell'account esterno devono delegare le autorizzazioni delle policy delle chiavi ai relativi utenti e ruoli. Queste policy sono impostate nell'account esterno e concedono autorizzazioni agli utenti e ai ruoli in tale account.

La policy chiave determina chi può avere accesso alla chiave KMS. La policy IAM determina chi ha accesso alla chiave KMS. Né la policy chiave né la policy IAM sono sufficienti, è necessario modificarle entrambe.

Per modificare la politica chiave, puoi utilizzare la [Policy View](#) in Console di gestione AWS o utilizzare le operazioni [CreateKey](#) o [PutKeyPolicy](#).

Per informazioni sulla modifica delle policy IAM, consulta [Utilizzo delle politiche IAM con AWS KMS](#).

Per un esempio che mostra come la policy chiave e le policy IAM collaborano per consentire l'utilizzo di una chiave KMS in un altro account, consulta [Esempio 2: L'utente assume un ruolo con il permesso di utilizzare una chiave KMS in un altro Account AWS](#).

[Puoi visualizzare le AWS KMS operazioni risultanti su più account sulla chiave KMS nei tuoi AWS CloudTrail registri](#). Le operazioni che usano le chiavi KMS in altri account vengono registrate sia nell'account del chiamante che nell'account del proprietario della chiave KMS.

Argomenti

- [Fase 1: aggiungere una dichiarazione di policy delle chiavi nell'account locale](#)
- [Fase 2: aggiungere le policy IAM nell'account esterno](#)
- [Autorizzazione per l'utilizzo di chiavi KMS esterne con Servizi AWS](#)
- [Utilizzo delle chiavi KMS in altri account](#)

Note

Gli esempi in questo argomento mostrano come utilizzare insieme una policy chiave e una policy IAM per fornire e limitare l'accesso a una chiave KMS. Questi esempi generici non intendono rappresentare le autorizzazioni richieste da un particolare utente su Servizio AWS

una chiave KMS. Per informazioni sulle autorizzazioni richieste da un Servizio AWS utente, consulta l'argomento relativo alla crittografia nella documentazione del servizio.

Fase 1: aggiungere una dichiarazione di policy delle chiavi nell'account locale

La policy delle chiavi per una chiave KMS è il principale determinante di chi può accedere alla chiave KMS e quali operazioni può eseguire. La policy della chiave è sempre definita nell'account proprietario della chiave KMS. A differenza delle policy IAM, le policy delle chiavi non specificano una risorsa. La risorsa è la chiave KMS associata alla policy delle chiavi. Nel processo di assegnazione delle autorizzazioni multi-account, la policy delle chiavi per la chiave KMS deve fornire all'account esterno (o utenti e ruoli nell'account esterno) l'autorizzazione per utilizzare la chiave KMS.

Per concedere a un account esterno l'autorizzazione per utilizzare la chiave KMS, aggiungi una dichiarazione alla policy delle chiavi che specifica l'account esterno. Nell'elemento `Principal` della policy delle chiavi, immetti l'Amazon Resource Name (ARN) dell'account esterno.

Quando specifichi un account esterno in una policy chiave, gli amministratori IAM nell'account esterno possono utilizzare le policy IAM per delegare tali autorizzazioni a qualsiasi utente e ruolo nell'account esterno. Possono anche decidere quali delle operazioni specificate nella policy delle chiavi possono eseguire gli utenti e i ruoli.

Le autorizzazioni concesse all'account esterno e alle relative entità principali sono valide solo se l'account esterno è abilitato nella Regione che ospita la chiave KMS e le relative policy chiave. Per informazioni sulle regioni non abilitate per impostazione predefinita ("regioni attivate"), consulta [Gestione delle Regioni AWS](#) nella Riferimenti generali di AWS.

Ad esempio, supponiamo che tu voglia permettere all'account 444455556666 di utilizzare una chiave KMS di crittografia simmetrica nell'account 111122223333. A tale scopo, aggiungi una dichiarazione di policy come quella nell'esempio seguente alla policy delle chiavi per la chiave KMS nell'account 111122223333. Questa istruzione di policy concede all'account esterno, 444455556666, l'autorizzazione per utilizzare la chiave KMS nelle operazioni di crittografia per le chiavi KMS di crittografia simmetrica.

 Note

L'esempio seguente rappresenta un esempio di policy delle chiavi per la condivisione di una chiave KMS con un altro account. Sostituisci i valori `Sid`, `Principal` e `Action` di esempio con valori validi per l'uso previsto della tua chiave KMS.

```
{
  "Sid": "Allow an external account to use this KMS key",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::444455556666:root"
    ]
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

Invece di concedere l'autorizzazione all'account esterno, puoi specificare particolari utenti e ruoli esterni nella policy delle chiavi. Tuttavia, tali utenti e ruoli non possono utilizzare la chiave KMS finché gli amministratori IAM nell'account esterno non collegano le policy IAM appropriate alle loro identità. Le policy IAM possono concedere l'autorizzazione a tutti o a un sottoinsieme di utenti e ruoli esterni specificati nella policy delle chiavi. Inoltre, possono consentire tutte o un sottoinsieme delle operazioni specificate nella policy delle chiavi.

La specifica delle identità in una policy delle chiavi limita le autorizzazioni che gli amministratori IAM nell'account esterno possono fornire. Tuttavia, rende più complessa la gestione delle policy con due account. Ad esempio, supponiamo che sia necessario aggiungere un utente o un ruolo. È necessario aggiungere tale identità alla policy delle chiavi nell'account proprietario della chiave KMS e creare policy IAM nell'account dell'identità.

Per specificare particolari utenti o ruoli esterni in una policy delle chiavi, nell'elemento `Principal` immettere l'Amazon Resource Name (ARN) di un utente o un ruolo nell'account esterno.

Ad esempio, la seguente istruzione di policy delle chiavi di esempio consente a `ExampleRole` nell'account 444455556666 di utilizzare una chiave KMS nell'account 111122223333. Questa istruzione di policy della chiave concede all'account esterno, 444455556666, l'autorizzazione per utilizzare la chiave KMS nelle operazioni di crittografia per le chiavi KMS di crittografia simmetrica.

Note

L'esempio seguente rappresenta un esempio di policy delle chiavi per la condivisione di una chiave KMS con un altro account. Sostituisci i valori `Sid`, `Principal` e `Action` di esempio con valori validi per l'uso previsto della tua chiave KMS.

```
{
  "Sid": "Allow an external account to use this KMS key",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::444455556666:role/ExampleRole"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

Note

Non impostare il principale su un asterisco (*) in un'istruzione della policy della chiave che consenta autorizzazioni, a meno che non utilizzi [condizioni](#) per limitare la policy della chiave. Un asterisco indica ogni identità in ogni Account AWS autorizzazione all'uso della chiave KMS, a meno che un'altra dichiarazione politica non lo neghi esplicitamente. Gli utenti di altri paesi Account AWS possono utilizzare la tua chiave KMS ogni volta che dispongono delle autorizzazioni corrispondenti nel proprio account.

È inoltre necessario decidere quali autorizzazioni concedere all'account esterno. Ad esempio, potresti voler concedere agli utenti l'autorizzazione per decrittare ma non crittografare, oppure l'autorizzazione per visualizzare la chiave KMS ma non utilizzarla. Per l'elenco delle autorizzazioni sulle chiavi KMS, consulta [AWS KMS autorizzazioni](#).

Impostazione della politica chiave quando si crea una chiave KMS

Quando si utilizza l'[CreateKey](#) operazione per creare una chiave KMS, è possibile utilizzare il relativo Policy parametro per specificare una politica chiave che consenta a un account esterno, o a utenti e ruoli esterni, l'autorizzazione a utilizzare la chiave KMS.

Quando si crea una chiave KMS in Console di gestione AWS, si crea anche la relativa politica chiave. Quando selezioni le identità nelle sezioni Amministratori delle chiavi e Utenti delle chiavi, AWS KMS aggiunge istruzioni di policy per tali identità alla policy chiave della chiave KMS. La sezione Key Users (Utenti chiave) consente inoltre di aggiungere account esterni come utenti delle chiavi.

Quando inserisci l'ID dell'account di un account esterno, AWS KMS aggiunge due dichiarazioni alla politica chiave. Questa operazione influisce solo sulla policy delle chiavi. Gli utenti e i ruoli nell'account esterno non possono utilizzare la chiave KMS finché non colleghi le policy IAM per concedere loro alcune o tutte queste autorizzazioni.

La prima istruzione di policy chiave concede all'account esterno l'autorizzazione per utilizzare la chiave KMS nelle operazioni di crittografia. La seconda dichiarazione politica chiave consente all'account esterno di creare, visualizzare e revocare le concessioni sulla chiave KMS, ma solo quando la richiesta proviene da un [AWS servizio](#) integrato con. AWS KMS Queste autorizzazioni consentono ad altri AWS servizi che crittografano i dati degli utenti di utilizzare la chiave KMS. Queste autorizzazioni sono progettate per le chiavi KMS che crittografano i dati degli utenti nei servizi AWS

Fase 2: aggiungere le policy IAM nell'account esterno

La policy delle chiavi nell'account proprietario della chiave KMS imposta l'intervallo valido per le autorizzazioni. Tuttavia, gli utenti e i ruoli nell'account esterno non possono utilizzare la chiave KMS finché non si collegano alle policy IAM che delegano tali autorizzazioni o utilizzano le concessioni per gestire l'accesso alla chiave KMS. Le policy IAM sono impostate nell'account esterno.

Se la policy delle chiavi concede l'autorizzazione all'account esterno, puoi collegare le policy IAM a qualsiasi utente o ruolo nell'account. Tuttavia, se la policy delle chiavi concede l'autorizzazione

a determinati utenti o ruoli, la policy IAM può concedere tali autorizzazioni solo a tutti o a un sottoinsieme degli utenti e dei ruoli specificati. Se una policy IAM concede alla chiave KMS l'accesso ad altri utenti o ruoli esterni, non ha alcun effetto.

La policy delle chiavi limita anche le operazioni nella policy IAM. La policy IAM può delegare tutte le operazioni o un sottoinsieme di quelle specificate nella policy delle chiavi. Se la policy IAM elenca le operazioni che non sono specificate nella policy delle chiavi, tali autorizzazioni non sono valide.

La policy IAM dell'esempio seguente consente al principale di utilizzare la chiave KMS nell'account 111122223333 per le operazioni di crittografia. Per concedere questa autorizzazione a utenti e ruoli nell'account 444455556666, [collega la policy](#) agli utenti o ai ruoli nell'account 444455556666.

Note

L'esempio seguente rappresenta un esempio di policy IAM per la condivisione di una chiave KMS con un altro account. Sostituisci i valori `Sid`, `Resource` e `Action` di esempio con valori validi per l'uso previsto della tua chiave KMS.

```
{
  "Sid": "AllowUseOfKeyInAccount111122223333",
  "Effect": "Allow",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Tieni presente quanto segue riguardo a questa policy:

- A differenza delle policy delle chiavi, le istruzioni delle policy IAM non contengono l'elemento `Principal`. Nelle policy IAM, il principale è implicito nell'identità a cui è collegata la policy.
- L'elemento `Resource` nella policy IAM identifica la chiave KMS che l'entità principale può utilizzare. Per specificare una chiave KMS, aggiungi l'[ARN chiave](#) all'elemento `Resource`.

- È possibile specificare più di una chiave KMS nell'elemento `Resource`. Tuttavia, se non specifichi particolari chiavi KMS nell'elemento `Resource`, potresti inavvertitamente concedere l'accesso a più chiavi KMS del previsto.
- Per consentire all'utente esterno di utilizzare la chiave KMS con i [servizi AWS che si integrano con AWS KMS](#), potrebbe essere necessario aggiungere autorizzazioni alla policy delle chiavi o alla policy IAM. Per informazioni dettagliate, consultare [Autorizzazione per l'utilizzo di chiavi KMS esterne con Servizi AWS](#).

Per ulteriori informazioni sull'utilizzo di policy IAM, consulta [Policy IAM](#).

Autorizzazione per l'utilizzo di chiavi KMS esterne con Servizi AWS

Puoi concedere a un utente in un account diverso l'autorizzazione per utilizzare la chiave KMS con un servizio integrato con AWS KMS. Ad esempio, un utente di un account esterno può utilizzare la tua chiave KMS per crittografare gli oggetti in un bucket Amazon S3 o per crittografare i segreti in cui sono archiviati. Gestione dei segreti AWS

La policy delle chiavi deve fornire all'utente esterno o all'account dell'utente esterno l'autorizzazione per utilizzare la chiave KMS. Inoltre, è necessario collegare policy IAM all'identità che concede all'utente l'autorizzazione per utilizzare il Servizio AWS. Il servizio potrebbe richiedere che gli utenti dispongano di autorizzazioni aggiuntive nella policy chiave o nella policy IAM. Per un elenco delle autorizzazioni Servizio AWS richieste su una chiave gestita dal cliente, consulta l'argomento Protezione dei dati nel capitolo Sicurezza della guida per l'utente o della guida per sviluppatori del servizio.

Utilizzo delle chiavi KMS in altri account

Se disponi dell'autorizzazione per utilizzare una chiave KMS in un'altra chiave Account AWS, puoi utilizzare la chiave KMS in Console di gestione AWS, AWS SDKs, AWS CLI e. AWS Strumenti per PowerShell

Per identificare una chiave KMS in un account diverso in un comando della shell o in una richiesta API, utilizzare i seguenti [identificatori chiave](#).

- Per [le operazioni di crittografia](#), e [DescribeKeyGetPublicKey](#), utilizzare la [chiave ARN](#) o l'[alias ARN](#) della chiave KMS.
- Per [CreateGrant](#), [GetKeyRotationStatusListGrants](#), e [RevokeGrant](#), usa la chiave ARN della chiave KMS.

Se inserisci solo un ID chiave o un nome alias, AWS presuppone che la chiave KMS sia nel tuo account.

La AWS KMS console non mostra le chiavi KMS in altri account, anche se sei autorizzato a usarle. Inoltre, gli elenchi di chiavi KMS visualizzati nelle console di altri servizi AWS non includono chiavi KMS in altri account.

Per specificare una chiave KMS in un altro account nella console di un AWS servizio, è necessario inserire la chiave ARN o l'alias ARN della chiave KMS. L'identificatore chiave richiesto varia a seconda del servizio e potrebbe differire tra la console del servizio e le relative operazioni API. Per dettagli, consultare la documentazione del servizio.

Controlla l'accesso alle chiavi multiregionali

È possibile utilizzare chiavi multi-regione in scenari di conformità, ripristino di emergenza e backup più complessi con le chiavi di singola regione. Tuttavia, poiché le proprietà di protezione delle chiavi multi-regione sono significativamente diverse da quelle delle chiavi di regione singola, si consiglia di prestare attenzione quando si autorizza la creazione, la gestione e l'utilizzo di chiavi multi-regione.

Note

Dichiarazioni di policy IAM esistenti con caratteri jolly nel campo Resource ora si applicano sia alle chiavi di regione singola che a quelle multi-regione. [Per limitarle alle chiavi KMS a regione singola o alle chiavi multiregione, usa la chiave kms: condition. MultiRegion](#)

Utilizza gli strumenti di autorizzazione per impedire la creazione e l'utilizzo di chiavi multi-regione in qualsiasi scenario in cui una singola regione è sufficiente. Consenti ai principali di replicare una chiave multiregionale solo in quelle che li richiedono. Regioni AWS Concedere l'autorizzazione per le chiavi multi-regione solo ai principali che ne hanno bisogno e solo per le attività che le richiedono.

Puoi utilizzare le policy chiave, le policy IAM e le sovvenzioni per consentire ai responsabili IAM di gestire e utilizzare chiavi multiregionali nel tuo Account AWS. Ogni chiave multi-regione è una risorsa indipendente con una chiave univoca ARN e una policy delle chiavi. È necessario stabilire e gestire una policy delle chiavi per ogni chiave e assicurarsi che le policy IAM nuove ed esistenti implementino la strategia di autorizzazione.

Per supportare le chiavi multiregionali, AWS KMS utilizza un ruolo collegato al servizio IAM. Questo ruolo dà a AWS KMS i permessi necessari per sincronizzare le [proprietà condivise](#). Per ulteriori informazioni, consulta [Autorizzazione AWS KMS alla sincronizzazione di chiavi multiregionali](#).

Argomenti

- [Nozioni di base sull'autorizzazione per le chiavi multi-regione](#)
- [Autorizzazione degli amministratori e degli utenti delle chiavi multi-regione](#)

Nozioni di base sull'autorizzazione per le chiavi multi-regione

Quando si progettano policy delle chiavi e policy IAM per chiavi multi-regione, considerare i seguenti principi.

- Policy delle chiavi — Ogni chiave multi-regione è una risorsa della chiave KMS indipendente con la propria [Policy delle chiavi](#). È possibile applicare la stessa policy o una policy delle chiavi diversa a ogni chiave nel set di chiavi multi-regione correlate. Le politiche chiave non sono [proprietà condivise](#) delle chiavi multiregionali. AWS KMS non copia o sincronizza le politiche chiave tra le chiavi multiregionali correlate.

Quando si crea una chiave di replica nella AWS KMS console, per comodità, la console visualizza la politica delle chiavi corrente della chiave primaria. È possibile utilizzare questa policy delle chiavi, modificarla o eliminarla e sostituirla. Ma anche se accetti invariata la politica della chiave primaria, AWS KMS non sincronizza le politiche. Ad esempio, se si modifica la policy delle chiavi della chiave primaria, la policy delle chiavi della chiave di replica rimane invariato.

- Politica di chiave predefinita: quando si creano chiavi multiregionali utilizzando le `ReplicateKey` operazioni [CreateKey](#)and, viene applicata la [politica di chiave predefinita](#) a meno che non si specifichi una politica chiave nella richiesta. Si tratta della stessa policy delle chiavi predefinita applicata alle chiavi di singola regione.
- Policy IAM — Come per tutte le chiavi KMS, è possibile utilizzare le policy IAM per controllare l'accesso alle chiavi multi-regione solo quando la [policy delle chiavi lo consente](#). [Le politiche IAM](#) si applicano a tutti per Regioni AWS impostazione predefinita. Tuttavia, puoi utilizzare chiavi condizionali, come [aws: RequestedRegion](#), per limitare le autorizzazioni a una particolare regione.

Per creare chiavi primarie e di replica, i principali devono disporre di autorizzazione `kms:CreateKey` in una policy IAM che si applica alla regione in cui viene creata la chiave.

- Sovvenzioni: le AWS KMS sovvenzioni [sono](#) regionali. Ogni concessione consente autorizzazioni per una chiave KMS. È possibile utilizzare le concessioni per consentire le autorizzazioni a una

chiave primaria o a una chiave di replica multi-regione. Tuttavia, non è possibile utilizzare una singola concessione per consentire le autorizzazioni a più chiavi KMS, anche se sono chiavi multi-regione correlate.

- ARN della chiave — Ogni chiave multi-regione ha un [ARN della chiave unico](#). La chiave ARNs delle chiavi multiregionali correlate ha la stessa partizione, lo stesso account e lo stesso ID di chiave, ma regioni diverse.

Per applicare un'istruzione delle policy IAM a una determinata chiave multi-regione, utilizza il relativo ARN della chiave o un pattern di ARN della chiave che include la regione. Per applicare un'istruzione di policy IAM a tutte le chiavi multi-regione correlate, utilizza un carattere jolly (*) nell'elemento Regione dell'ARN, come mostrato nell'esempio seguente.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:Describe*",
    "kms:List*"
  ],
  "Resource": {
    "arn:aws:kms:*::111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab"
  }
}
```

Per applicare una dichiarazione politica a tutte le chiavi multiregionali presenti Account AWS, puoi utilizzare la condizione [kms: MultiRegion](#) policy o un modello di ID chiave che include il prefisso distintivo. `mrk-`

- Ruolo collegato ai servizi: [i responsabili che creano chiavi primarie multiregionali devono disporre dell'autorizzazione iam: CreateServiceLinkedRole](#)

[Per sincronizzare le proprietà condivise delle chiavi multiregionali correlate, AWS KMS presuppone un ruolo collegato ai servizi IAM.](#) AWS KMS crea il ruolo collegato al servizio in Account AWS ogni volta che si crea una chiave primaria multiregionale. (Se il ruolo esiste, AWS KMS lo ricrea, cosa che non ha alcun effetto dannoso.) Il ruolo è valido in tutte le Regioni. AWS KMS [Per consentire la creazione \(o la ricreazione\) del ruolo collegato al servizio, i responsabili che creano chiavi primarie multiregionali devono disporre dell'autorizzazione iam: CreateServiceLinkedRole](#)

Autorizzazione degli amministratori e degli utenti delle chiavi multi-regione

I principali che creano e gestiscono chiavi multi-regione necessitano delle seguenti autorizzazioni nelle regioni primarie e di replica:

- kms:CreateKey
- kms:ReplicateKey
- kms:UpdatePrimaryRegion
- iam:CreateServiceLinkedRole

Creazione di una chiave primaria

Per [creare una chiave primaria multiregionale](#), il principale necessita delle CreateServiceLinkedRole autorizzazioni [kms: CreateKey](#) e [iam:](#) in una policy IAM efficace nella regione della chiave primaria. I principali che dispongono di queste autorizzazioni possono creare chiavi di regione singola e multi-regione a meno che non si limitino le autorizzazioni.

L'iam:CreateServiceLinkedRole autorizzazione consente di AWS KMS creare il [AWSServiceRoleForKeyManagementServiceMultiRegionKeysruolo](#) per sincronizzare le [proprietà condivise delle relative chiavi multiregionali](#).

Ad esempio, questa policy IAM consente a un preside di creare chiavi multiregionali, allegare politiche per tali chiavi e ruoli collegati al servizio per chiavi multiregionali.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Action": [
      "kms:CreateKey",
      "iam:CreateServiceLinkedRole"
    ],
    "Effect": "Allow",
    "Resource": "*"
  }
}
```


Per consentire o negare l'autorizzazione alla creazione di chiavi primarie multiregionali, usa la chiave [kms:condition](#). MultiRegion I valori validi sono `true` (Chiave multi-regione) o `false` (Chiave di singola regione). Ad esempio, la seguente istruzione di policy IAM utilizza un'operazione Deny con la chiave di condizione `kms:MultiRegion` per impedire ai principali di creare chiavi multi-regione.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "kms:CreateKey",
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "Bool": {
          "kms:MultiRegion": true
        }
      }
    }
  ]
}
```

Replica delle chiavi

Per [creare una chiave di replica multi-regione](#), il principale richiede le seguenti autorizzazioni:

- [kms: ReplicateKey](#) autorizzazione nella politica chiave della chiave primaria.
- [kms: CreateKey](#) autorizzazione in una politica IAM efficace nella regione della chiave di replica.

Presta attenzione quando consenti queste autorizzazioni. Consentono ai principali di creare chiavi KMS e le policy delle chiavi che ne autorizzano l'utilizzo. L'autorizzazione `kms:ReplicateKey` autorizza inoltre il trasferimento di materiale chiave oltre i confini della regione all'interno di AWS KMS.

Per limitare il numero Regioni AWS di repliche di una chiave multiregionale, usa la chiave [kms:condition](#). ReplicaRegion Limita solo l'autorizzazione `kms:ReplicateKey`. Altrimenti, non ha efficacia. Ad esempio, la seguente policy delle chiavi consente al principale di replicare questa chiave primaria, ma solo nelle regioni specificate.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Administrator"
  },
  "Action": "kms:ReplicateKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ReplicaRegion": [
        "us-east-1",
        "eu-west-3",
        "ap-southeast-2"
      ]
    }
  }
}
```

Aggiornamento della regione primaria

I principali autorizzati possono convertire una chiave di replica in una chiave primaria, modificando la chiave primaria precedente in una replica. Questa azione è nota come [aggiornamento della regione primaria](#). Per aggiornare la regione principale, la principale necessita dell'`UpdatePrimaryRegion` autorizzazione [kms](#): in entrambe le regioni. Puoi fornire queste autorizzazioni in una policy delle chiavi o in una policy IAM.

- `kms:UpdatePrimaryRegion` sulla chiave primaria. Questa autorizzazione deve essere valida nella regione chiave primaria.
- `kms:UpdatePrimaryRegion` sulla chiave di replica. Questa autorizzazione deve essere valida nella regione chiave di replica.

Ad esempio, la seguente policy delle chiavi consente agli utenti che possono assumere il ruolo di Amministratore di aggiornare la regione primaria della chiave KMS. Questa chiave KMS può essere la chiave primaria o una chiave di replica in questa operazione.

```
{
  "Effect": "Allow",
  "Resource": "*",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Administrator"
  }
}
```

```
},  
  "Action": "kms:UpdatePrimaryRegion"  
}
```

Per limitare la Regione AWS capacità di ospitare una chiave primaria, usa la chiave [kms:PrimaryRegion](#) condition. Ad esempio, la seguente dichiarazione politica IAM consente ai responsabili di aggiornare la regione principale delle chiavi multiregione nella Account AWS, ma solo quando la nuova regione primaria è una delle regioni specificate.

```
{  
  "Effect": "Allow",  
  "Action": "kms:UpdatePrimaryRegion",  
  "Resource": {  
    "arn:aws:kms:*:111122223333:key/*"  
  },  
  "Condition": {  
    "StringEquals": {  
      "kms:PrimaryRegion": [  
        "us-west-2",  
        "sa-east-1",  
        "ap-southeast-1"  
      ]  
    }  
  }  
}
```

Utilizzo e gestione di chiavi multi-regione

Per impostazione predefinita, i principali che dispongono dell'autorizzazione per utilizzare e gestire le chiavi KMS in un Account AWS e Regione hanno anche l'autorizzazione per utilizzare e gestire chiavi multi-regione. Tuttavia, puoi utilizzare la chiave [kms: MultiRegion](#) condition per consentire solo chiavi a regione singola o solo chiavi multiregione. Oppure usa la chiave [kms: MultiRegionKeyType](#) condition per consentire solo chiavi primarie multiregionali o solo chiavi di replica. [Entrambi i tasti condizionali controllano l'accesso all'CreateKeyoperazione e a qualsiasi operazione che utilizza una chiave KMS esistente, come Encrypt o. EnableKey](#)

La seguente istruzione di policy IAM utilizza la chiave di condizione `kms:MultiRegion` per impedire ai principali di utilizzare o gestire qualsiasi chiave multi-regione.

```
{  
  "Effect": "Deny",
```

```
"Action": "kms:*",
"Resource": "*",
"Condition": {
  "Bool": "kms:MultiRegion": true
}
}
```

In questo esempio, l'istruzione di policy IAM utilizza la condizione `kms:MultiRegionKeyType` per consentire ai principali di pianificare e annullare l'eliminazione delle chiavi, ma solo sulle chiavi di replica multi-regione.

```
{
  "Effect": "Allow",
  "Action": [
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": {
    "arn:aws:kms:us-west-2:111122223333:key/*"
  },
  "Condition": {
    "StringEquals": "kms:MultiRegionKeyType": "REPLICA"
  }
}
```

Determinare l'accesso a AWS KMS keys

Per determinare la portata completa di chi o cosa ha attualmente accesso a una chiave AWS KMS key, è necessario esaminare la politica chiave della chiave KMS, tutte le [concessioni](#) che si applicano alla chiave KMS e potenzialmente tutte le politiche AWS Identity and Access Management (IAM). Ciò potrebbe essere necessario per determinare la portata dell'utilizzo potenziale di una chiave KMS oppure per aiutarti a soddisfare i requisiti di conformità o di verifica. I seguenti argomenti aiutano a generare un elenco completo dei principali AWS (identità) che attualmente hanno accesso a una chiave KMS.

Argomenti

- [Analisi della policy delle chiavi](#)
- [Analisi delle policy IAM](#)
- [Analisi delle concessioni](#)

Analisi della policy delle chiavi

Le [policy chiave](#) sono lo strumento principale per controllare l'accesso alle chiavi KMS. Ogni chiave KMS ha esattamente una policy chiave.

Quando una policy delle chiavi è costituita da o include la [policy chiave predefinita](#), la policy chiave consente agli amministratori IAM dell'account di utilizzare le policy IAM per controllare l'accesso alla chiave KMS. Inoltre, se la policy delle chiavi concede a un [altro Account AWS](#) l'autorizzazione per utilizzare la chiave KMS, gli amministratori IAM dell'account esterno possono utilizzare le policy IAM per delegare tali autorizzazioni. Per determinare l'elenco completo dei principali che possono accedere alla chiave KMS, [esamina le policy IAM](#).

Per visualizzare la politica chiave di una [chiave gestita AWS KMS dal cliente](#) o [Chiave gestita da AWS](#) nel tuo account, utilizza l'[GetKeyPolicy](#) operazione Console di gestione AWS o nell'API. AWS KMS Per visualizzare la policy delle chiavi, è necessario disporre delle autorizzazioni `kms:GetKeyPolicy` per la chiave KMS. Per istruzioni sulla visualizzazione delle policy delle chiavi per una chiave KMS, consulta [the section called “Visualizza una politica chiave”](#).

Esamina il documento di policy delle chiavi e prendi nota di tutti i principali specificati in ciascun elemento `Principal` dell'istruzione di policy. In una dichiarazione politica con `Allow` effetto, gli utenti IAM, i ruoli IAM e Account AWS nell'`Principal` elemento hanno accesso a questa chiave KMS.

Note

Non impostare il principale su un asterisco (*) in un'istruzione della policy della chiave che consenta autorizzazioni, a meno che non utilizzi [condizioni](#) per limitare la policy della chiave. Un asterisco indica ogni identità in ogni Account AWS autorizzazione all'uso della chiave KMS, a meno che un'altra dichiarazione politica non lo neghi esplicitamente. Gli utenti di altri paesi Account AWS possono utilizzare la tua chiave KMS ogni volta che dispongono delle autorizzazioni corrispondenti nel proprio account.

I seguenti esempi utilizzano le istruzioni di policy incluse nella [policy delle chiavi predefinita](#) per mostrare come eseguire questa operazione.

Example Istruzione di policy 1

```
{
```

```
"Sid": "Enable IAM User Permissions",
"Effect": "Allow",
"Principal": {"AWS": "arn:aws:iam::111122223333:root"},
"Action": "kms:*",
"Resource": "*"
}
```

Nella dichiarazione politica 1, `arn:aws:iam::111122223333:root` è l'intestazione [AWS del conto che si](#) riferisce al Account AWS 111122223333. (Non è l'utente root dell'account). Per impostazione predefinita, una dichiarazione politica come questa viene inclusa nel documento della politica chiave quando si crea una nuova chiave KMS con o si crea una nuova chiave KMS a livello di codice Console di gestione AWS, ma non si fornisce una politica chiave.

Un documento strategico chiave con una dichiarazione che consente l'accesso alle [politiche IAM Account AWS abilitate all'account per consentire l'accesso alla](#) chiave KMS. Pertanto, gli utenti e i ruoli nell'account potrebbero avere accesso alla chiave KMS anche se non sono elencati in modo esplicito come principali nel documento di policy delle chiavi. Assicurati di [esaminare tutte le policy IAM](#) Account AWS elencate come principali per determinare se consentono l'accesso a questa chiave KMS.

Example Istruzione di policy 2

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/KMSKeyAdmins"},
  "Action": [
    "kms:Describe*",
    "kms:Put*",
    "kms:Create*",
    "kms:Update*",
    "kms:Enable*",
    "kms:Revoke*",
    "kms:List*",
    "kms:Disable*",
    "kms:Get*",
    "kms>Delete*",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "*"
}
```

```
}
```

Nella dichiarazione politica 2, `arn:aws:iam::111122223333:role/KMSKeyAdmins` fa riferimento al ruolo IAM denominato KMSKey Admins in 111122223333. Account AWS Gli utenti autorizzati ad assumere questo ruolo sono autorizzati ad eseguire le operazioni elencate nell'istruzione della policy, che sono le operazioni amministrative per la gestione di una chiave KMS.

Example Istruzione di policy 3

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/EncryptionApp"},
  "Action": [
    "kms:DescribeKey",
    "kms:GenerateDataKey*",
    "kms:Encrypt",
    "kms:ReEncrypt*",
    "kms:Decrypt"
  ],
  "Resource": "*"
}
```

Nella dichiarazione politica 3, `arn:aws:iam::111122223333:role/EncryptionApp` fa riferimento al ruolo IAM denominato in 111122223333. EncryptionApp Account AWS I principali autorizzati ad assumere questo ruolo sono autorizzati a eseguire le operazioni riportate nell'istruzione della policy, che includono le [operazioni di crittografia](#) per una chiave KMS di crittografia simmetrica.

Example Istruzione di policy 4

```
{
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/EncryptionApp"},
  "Action": [
    "kms:ListGrants",
    "kms:CreateGrant",
    "kms:RevokeGrant"
  ],
  "Resource": "*",
  "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
```

```
}
```

Nella dichiarazione politica 4, `arn:aws:iam::111122223333:role/EncryptionApp` fa riferimento al ruolo IAM denominato EncryptionApp in Account AWS 111122223333. I principali autorizzati ad assumere questo ruolo sono autorizzati a eseguire le operazioni elencate nell'istruzione della policy. Queste operazioni, quando combinate con le operazioni consentite nell'Esempio di istruzione 3 della policy, sono quelle necessarie per delegare l'utilizzo della chiave KMS alla maggior parte dei servizi [AWS che si integrano con AWS KMS](#), in particolare i servizi che utilizzano [concessioni](#). Il `GrantIsFor AWSResource` valore `kms:` nell'`Condition` elemento assicura che la delega sia consentita solo quando il delegato è un AWS servizio che si integra AWS KMS e utilizza le concessioni per l'autorizzazione.

Per informazioni sui diversi modi in cui è possibile specificare un principale in un documento della policy chiave, consulta [Specifica un principale](#) nella Guida per l'utente IAM.

Per ulteriori informazioni sulle politiche AWS KMS chiave, consulta [Politiche chiave in AWS KMS](#)

Analisi delle policy IAM

Oltre alla policy delle chiavi e alle concessioni, puoi anche utilizzare le [policy IAM](#) per consentire l'accesso a una chiave KMS. Per ulteriori informazioni sull'utilizzo congiunto delle policy IAM e delle policy chiave, consulta [Risoluzione dei problemi relativi alle autorizzazioni AWS KMS](#).

Per determinare quali principali hanno attualmente accesso a una chiave KMS tramite le policy IAM, puoi utilizzare lo strumento [Simulatore di policy IAM](#) basato su browser o puoi effettuare richieste all'API IAM.

Modi per esaminare le policy IAM

- [Analisi delle policy IAM con il simulatore di policy IAM](#)
- [Analisi delle policy IAM con l'API IAM](#)

Analisi delle policy IAM con il simulatore di policy IAM

Il simulatore di policy IAM può aiutarti a scoprire i principali che possono accedere a una chiave KMS tramite una policy IAM.

Per utilizzare il simulatore di policy IAM per determinare l'accesso a una chiave KMS

1. Accedi a Console di gestione AWS e poi apri IAM Policy Simulator all'indirizzo <https://policysim.aws.amazon.com/>.
2. Nel riquadro Users, Groups, and Roles (Utenti, gruppi e ruoli), scegliere l'utente, il gruppo o il ruolo del quale si intende simulare le policy.
3. (Opzionale) Deseleziona la casella di controllo accanto a qualsiasi policy che desideri omettere dalla simulazione. Per simulare tutte le policy, lascia tutte le policy selezionate.
4. Nel riquadro Policy Simulator (Simulatore di policy), seguire la procedura riportata di seguito:
 - a. Per Select service (Seleziona servizio), scegliere Key Management Service.
 - b. Per simulare AWS KMS azioni specifiche, in Seleziona azioni, scegli le azioni da simulare. Per simulare tutte le AWS KMS azioni, scegliete Seleziona tutto.
5. (Opzionale) Il simulatore di policy simula l'accesso a tutte le chiavi KMS per impostazione predefinita. Per simulare l'accesso a una determinata chiave KMS, scegli Impostazioni di simulazione, quindi digita l'Amazon Resource Name (ARN) della chiave KMS da simulare.
6. Scegliere Run Simulation (Esegui simulazione).

È possibile visualizzare i risultati della simulazione nella sezione Results (Risultati). Ripeti le fasi da 2 a 6 per ogni utente, gruppo e ruolo nell' Account AWS.

Analisi delle policy IAM con l'API IAM

È possibile utilizzare l'API IAM per esaminare le policy IAM a livello di codice. Le seguenti fasi forniscono una panoramica generale su come eseguire questa operazione:

1. Per ogni account Account AWS elencato come principale nella policy chiave (ovvero, ogni [AWS account principal](#) specificato in questo formato: "Principal": {"AWS": "arn:aws:iam::111122223333:root"}), utilizza le [ListRoles](#) operazioni [ListUsers](#) and nell'API IAM per inserire tutti gli utenti e i ruoli nell'account.
2. Per ogni utente e ruolo nell'elenco, utilizza l'[SimulatePrincipalPolicy](#) operazione nell'API IAM, passando i seguenti parametri:
 - Per PolicySourceArns specificare il nome ARN (Amazon Resource Name) di un utente o un ruolo dal tuo elenco. Puoi specificare un solo PolicySourceArn per ogni richiesta SimulatePrincipalPolicy, pertanto è necessario chiamare questa operazione più volte, una volta per ogni utente e ruolo nell'elenco.

- Per l'ActionNameselenco, specifica ogni azione AWS KMS API da simulare. Per simulare tutte le azioni AWS KMS dell'API, usa `kms : *` Per testare le singole azioni AWS KMS API, fai precedere ogni azione API da `"kms : «`, ad esempio `"kms : ListKeys`. Per un elenco completo delle operazioni API AWS KMS , consulta [Azioni](#) nella Documentazione di riferimento dell'API AWS Key Management Service .
- (Facoltativo) Per determinare se gli utenti o i ruoli hanno accesso a chiavi KMS specifiche, utilizza il `ResourceArns` parametro per specificare un elenco di Amazon Resource Names (ARNs) delle chiavi KMS. Per determinare se gli utenti o i ruoli possono accedere a una chiave KMS, ometti il parametro `ResourceArns`.

IAM risponde a ogni richiesta `SimulatePrincipalPolicy` con una valutazione: `allowed`, `explicitDeny` o `implicitDeny`. Per ogni risposta che contiene una decisione di valutazione `diallowed`, la risposta include il nome della specifica operazione AWS KMS API consentita. Eventualmente, include l'ARN della chiave KMS usata nella valutazione.

Analisi delle concessioni

Le sovvenzioni sono meccanismi avanzati per specificare le autorizzazioni che l'utente o un AWS servizio integrato AWS KMS può utilizzare per specificare come e quando è possibile utilizzare una chiave KMS. Le concessioni sono collegate a una chiave KMS; ogni concessione, inoltre, contiene il principale che riceve l'autorizzazione per utilizzare la chiave KMS e un elenco di operazioni consentite. Le concessioni sono un'alternativa alla policy delle chiavi e sono utili per casi d'uso specifici. Per ulteriori informazioni, consulta [Sovvenzioni in AWS KMS](#).

Per ottenere un elenco di concessioni per una chiave KMS, usa l'operazione. AWS KMS [ListGrants](#) Puoi esaminare le concessioni per una chiave KMS per determinare chi o cosa dispone attualmente dell'autorizzazione per utilizzare la chiave KMS tramite tali concessioni. L'esempio seguente è una rappresentazione JSON di una concessione che è stata ottenuta dal comando [list-grants](#) nella AWS CLI.

```
{"Grants": [{
  "Operations": ["Decrypt"],
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Name": "0d8aa621-43ef-4657-b29c-3752c41dc132",
  "RetiringPrincipal": "arn:aws:iam::123456789012:root",
  "GranteePrincipal": "arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/
i-5d476fab",
```

```
"GrantId": "dc716f53c93acacf291b1540de3e5a232b76256c83b2ecb22cdefa26576a2d3e",
"IssuingAccount": "arn:aws:iam::111122223333:root",
"CreationDate": 1.444151834E9,
"Constraints": {"EncryptionContextSubset": {"aws:ebs:id": "vol-5cccfb4e"}}
}]}
```

Per scoprire chi o cosa dispone dell'autorizzazione per utilizzare la chiave KMS, cerca l'elemento `"GranteePrincipal"`. Nell'esempio precedente, il beneficiario principale è un utente con ruolo assunto associato all'istanza `i-5d476fab`. EC2 L' EC2 infrastruttura utilizza questo ruolo per collegare il volume EBS crittografato `vol-5cccfb4e` all'istanza. In questo caso, il ruolo dell' EC2 infrastruttura è autorizzato a utilizzare la chiave KMS perché in precedenza hai creato un volume EBS crittografato protetto da questa chiave KMS. Hai quindi collegato il volume a un'istanza. EC2

L'esempio seguente è un'altra rappresentazione JSON di una concessione che è stata ottenuta dal comando [list-grants](#) nella AWS CLI. Nell'esempio seguente, il titolare del beneficiario è un altro. Account AWS

```
{"Grants": [{
  "Operations": ["Encrypt"],
  "KeyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Name": "",
  "GranteePrincipal": "arn:aws:iam::444455556666:root",
  "GrantId": "f271e8328717f8bde5d03f4981f06a6b3fc18bcae2da12ac38bd9186e7925d11",
  "IssuingAccount": "arn:aws:iam::111122223333:root",
  "CreationDate": 1.444151269E9
}]}
```

Contesto di crittografia

Note

Non è possibile specificare un contesto di crittografia in un'operazione di crittografia con una [chiave KMS asimmetrica](#) o una [chiave KMS HMAC](#). Gli algoritmi asimmetrici e gli algoritmi MAC non supportano un contesto di crittografia.

Tutte le [operazioni AWS KMS crittografiche](#) con [chiavi KMS a crittografia simmetrica](#) accettano un contesto di crittografia, un set opzionale di coppie chiave-valore non segrete che possono contenere

informazioni contestuali aggiuntive sui dati. È possibile inserire un contesto di crittografia nelle Encrypt operazioni AWS KMS per migliorare l'autorizzazione e la verificabilità delle chiamate di decrittografia delle API. AWS KMS utilizza il contesto di crittografia come dati autenticati aggiuntivi (AAD) per supportare la crittografia autenticata. Il contesto di crittografia è associato crittograficamente al testo cifrato in modo che lo stesso contesto di crittografia sia necessario per decrittografare i dati.

Il contesto di crittografia non è segreto e non è crittografato. Appare in testo normale nei [log AWS CloudTrail](#) in modo da poterlo utilizzare per individuare e categorizzare le operazioni di crittografia. Il contesto di crittografia non dovrebbe includere informazioni sensibili. È consigliabile che il contesto di crittografia descriva i dati crittografati o decrittografati. Ad esempio, quando si esegue la crittografia di un file, è possibile utilizzare parte del percorso di file come contesto di crittografia.

```
"encryptionContext": {
  "department": "10103.0"
}
```

Ad esempio, quando crittografa volumi e snapshot creati con l'operazione [Amazon Elastic Block Store](#) (Amazon EBS) [CreateSnapshot](#), Amazon EBS utilizza l'ID del volume come valore del contesto di crittografia.

```
"encryptionContext": {
  "aws:ebs:id": "vol-abcde12345abc1234"
}
```

Puoi anche utilizzare il contesto di crittografia per perfezionare o limitare l'accesso al tuo account. AWS KMS keys È possibile utilizzare il contesto di crittografia [come vincolo nelle concessioni](#) e come una [condizione nelle istruzioni di policy](#). Le chiavi del contesto di crittografia e i relativi valori possono essere stringhe arbitrarie con. aws [Questi valori contrastano con tag AWS generati come aws:cloudformation:stack-name](#). Per ulteriori informazioni, consulta [kms:: chiave contestuale EncryptionContext](#)

Per informazioni su come utilizzare il contesto di crittografia per proteggere l'integrità dei dati crittografati, consulta il post [Come proteggere l'integrità dei dati crittografati utilizzando e sul](#) blog sulla sicurezza. AWS Key Management Service EncryptionContext AWS

Regole sul contesto di crittografia

AWS KMS applica le seguenti regole per le chiavi e i valori del contesto di crittografia.

- La chiave e il valore in una coppia del contesto di crittografia devono essere stringhe letterali semplici. Se utilizzi un tipo diverso, ad esempio integer o float, AWS KMS lo interpreta come una stringa.
- Le chiavi e i valori in un contesto di crittografia possono includere caratteri Unicode. Se un contesto di crittografia include caratteri non consentiti nelle policy della chiave o nelle policy IAM, non sarà possibile specificare il contesto di crittografia nelle chiavi di condizioni della policy, ad esempio [kms:EncryptionContext:context-key](#) e [kms:EncryptionContextKeys](#). Per maggiori informazioni sulle regole delineate nel documento delle policy delle chiavi, consulta la sezione [Formato della policy della chiave](#). Per informazioni dettagliate sulle regole delineate nel documento delle policy IAM, consulta la sezione [Requisiti del nome IAM](#) nella Guida per l'utente di IAM.

Contesto di crittografia nelle policy

Il contesto di crittografia viene utilizzato principalmente per verificare l'integrità e l'autenticità. Tuttavia, puoi anche utilizzare il contesto di crittografia per controllare l'accesso alle AWS KMS keys di crittografia simmetrica nelle policy delle chiavi e nelle policy IAM.

Le chiavi [kms:EncryptionContext:](#) e [kms: EncryptionContextKeys](#) condition consentono (o negano) un'autorizzazione solo quando la richiesta include particolari chiavi di contesto di crittografia o coppie chiave-valore.

Ad esempio, la seguente istruzione della policy delle chiavi consente al ruolo RoleForExampleApp di utilizzare la chiave KMS nelle operazioni Decrypt. Utilizza la chiave di condizione `kms:EncryptionContext:context-key` per concedere questa autorizzazione solo quando il contesto di crittografia nella richiesta include una coppia di contesto di crittografia `AppName:ExampleApp`.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    }
  }
}
```

```
}
```

Per ulteriori informazioni su queste chiavi di condizione del contesto di crittografia, consulta [Chiavi di condizione per AWS KMS](#).

Contesto di crittografia nelle concessioni

Quando [crei una concessione, puoi includere vincoli di concessione che stabiliscono](#) le condizioni per le autorizzazioni di concessione. AWS KMS supporta due vincoli di concessione `EncryptionContextEquals` ed `EncryptionContextSubset` entrambi coinvolgono il [contesto di crittografia](#) in una richiesta di operazione crittografica. Quando utilizzi questi vincoli di concessione, le autorizzazioni nella concessione sono valide solo quando il contesto di crittografia nella richiesta per l'operazione di crittografia soddisfa i requisiti dei vincoli di concessione.

Ad esempio, è possibile aggiungere un vincolo di `EncryptionContextEquals` concessione a una concessione che consente l'operazione. [GenerateDataKey](#) Con questo vincolo, la concessione consente l'operazione solo quando il contesto di crittografia nella richiesta corrisponde a livello di maiuscole e minuscole al contesto di crittografia nel vincolo di concessione.

```
$ aws kms create-grant \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --grantee-principal arn:aws:iam::111122223333:user/exampleUser \  
  --retiring-principal arn:aws:iam::111122223333:role/adminRole \  
  --operations GenerateDataKey \  
  --constraints EncryptionContextEquals={Purpose=Test}
```

Una richiesta come la seguente dal principale beneficiario soddisferebbe il `EncryptionContextEquals` vincolo.

```
$ aws kms generate-data-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --key-spec AES_256 \  
  --encryption-context Purpose=Test
```

Per ulteriori informazioni sui vincoli della concessione, consulta [Utilizzo dei vincoli di concessione](#). Per informazioni dettagliate sugli alias, consulta [the section called “Concessioni”](#).

Registrazione del contesto di crittografia

AWS KMS utilizza AWS CloudTrail per registrare il contesto di crittografia in modo da poter determinare a quali chiavi e dati KMS è stato effettuato l'accesso. La voce di log mostra esattamente la chiave KMS utilizzata per crittografare o decrittare dati specifici a cui fa riferimento il contesto di crittografia nella voce di log.

Important

Poiché il contesto di crittografia viene registrato, non deve contenere informazioni sensibili.

Archiviazione del contesto di crittografia

Per semplificare l'utilizzo di qualsiasi contesto di crittografia quando chiami le operazioni [Decrypt](#) o [ReEncrypt](#), puoi archiviare il contesto di crittografia insieme ai dati crittografati. Ti consigliamo di archiviare solo il contesto di crittografia sufficiente per aiutarti a creare il contesto di crittografia completo quando ne hai bisogno per la crittografia o la decrittografia.

Ad esempio, se il contesto di crittografia è il percorso completo di un file, archivia solo la parte del percorso con i contenuti crittografati del file. Quando ti occorre il contesto di crittografia completo, puoi ricostruirlo dal frammento archiviato. Se qualcuno altera il file, ad esempio lo rinomina o lo sposta in un percorso diverso, il valore del contesto di crittografia cambia e la richiesta di decrittografia ha esito negativo.

Test delle autorizzazioni

Per utilizzarle AWS KMS, devi disporre di credenziali che AWS possano essere utilizzate per autenticare le tue richieste API. Le credenziali devono includere l'autorizzazione per accedere alle chiavi KMS e agli alias. Le autorizzazioni sono determinate dalle policy delle chiavi, dalle policy IAM, dalle concessioni e dai controlli di accesso multi-account. Oltre a controllare l'accesso alle chiavi KMS, puoi controllare l'accesso al tuo CloudHSM e ai tuoi archivi di chiavi personalizzate.

Puoi specificare il parametro dell'API `DryRun` per controllare se disponi delle autorizzazioni necessarie a utilizzare le chiavi AWS KMS. È inoltre possibile utilizzare `DryRun` per verificare che i parametri di richiesta in una chiamata AWS KMS API siano specificati correttamente.

Argomenti

- [Qual è il DryRun parametro?](#)
- [Specificazione DryRun con l'API](#)

Qual è il DryRun parametro?

DryRun è un parametro dell'API opzionale specificato per controllare se l'esito delle chiamate API AWS KMS sarà positivo. Usa DryRun per testare la chiamata API, prima di effettuare realmente la chiamata a AWS KMS. Puoi effettuare i controlli seguenti:

- che disponi delle autorizzazioni necessarie per utilizzare le chiavi AWS KMS ;
- che hai specificato correttamente i parametri nella chiamata.

AWS KMS supporta l'utilizzo del DryRun parametro in determinate azioni API:

- [CreateGrant](#)
- [Decrypt](#)
- [DeriveSharedSecret](#)
- [Encrypt](#)
- [GenerateDataKey](#)
- [GenerateDataKeyPair](#)
- [GenerateDataKeyPairWithoutPlaintext](#)
- [GenerateDataKeyWithoutPlaintext](#)
- [GenerateMac](#)
- [ReEncrypt](#)
- [RetireGrant](#)
- [RevokeGrant](#)
- [Sign](#)
- [Verify](#)
- [VerifyMac](#)

L'utilizzo del parametro DryRun comporterà dei costi e verrà fatturato come richiesta API standard. Per ulteriori informazioni sui AWS KMS prezzi, consulta la sezione [AWS Key Management Service Prezzi](#).

Tutte le richieste API con il parametro `DryRun` si applicano alla quota della richiesta API e possono comportare un'eccezione di limitazione della larghezza di banda della rete se superi la quota della richiesta API. Ad esempio, la chiamata [Decrypt](#) con `DryRun` o senza `DryRun` viene conteggiata sulla stessa quota delle operazioni crittografiche. Per ulteriori informazioni, consulta [Richieste di limitazione AWS KMS](#).

Ogni chiamata a un'operazione AWS KMS API viene acquisita come evento e registrata in un AWS CloudTrail registro. L'output di tutte le operazioni che specificano il `DryRun` parametro viene visualizzato nel CloudTrail registro. Per ulteriori informazioni, consulta [Registrazione delle chiamate AWS KMS API con AWS CloudTrail](#).

Specificazione DryRun con l'API

Per utilizzarlo `DryRun`, specifica il `--dry-run` parametro nei AWS CLI comandi e nelle chiamate AWS KMS API che supportano il parametro. Quando lo AWS KMS farai, verificherà se la chiamata avrà esito positivo. AWS KMS le chiamate che vengono utilizzate `DryRun` avranno sempre esito negativo e restituiranno un messaggio con informazioni sul motivo per cui la chiamata non è riuscita. Il messaggio può includere le seguenti eccezioni:

- `DryRunOperationException`: l'esito della richiesta sarebbe stato positivo se non fosse stato specificato `DryRun`.
- `ValidationException`: l'esito della richiesta è negativo perché è stato specificato un parametro dell'API errato.
- `AccessDeniedException`: non disponi delle autorizzazioni per l'esecuzione dell'azione dell'API specificata sulla risorsa KMS.

Ad esempio, il comando seguente utilizza l'[CreateGrant](#) operazione e crea una concessione che consente agli utenti autorizzati ad assumere il `keyUserRole` ruolo di chiamare l'operazione [Decrypt](#) su una chiave KMS [simmetrica](#) specificata. Il parametro `DryRun` è specificato.

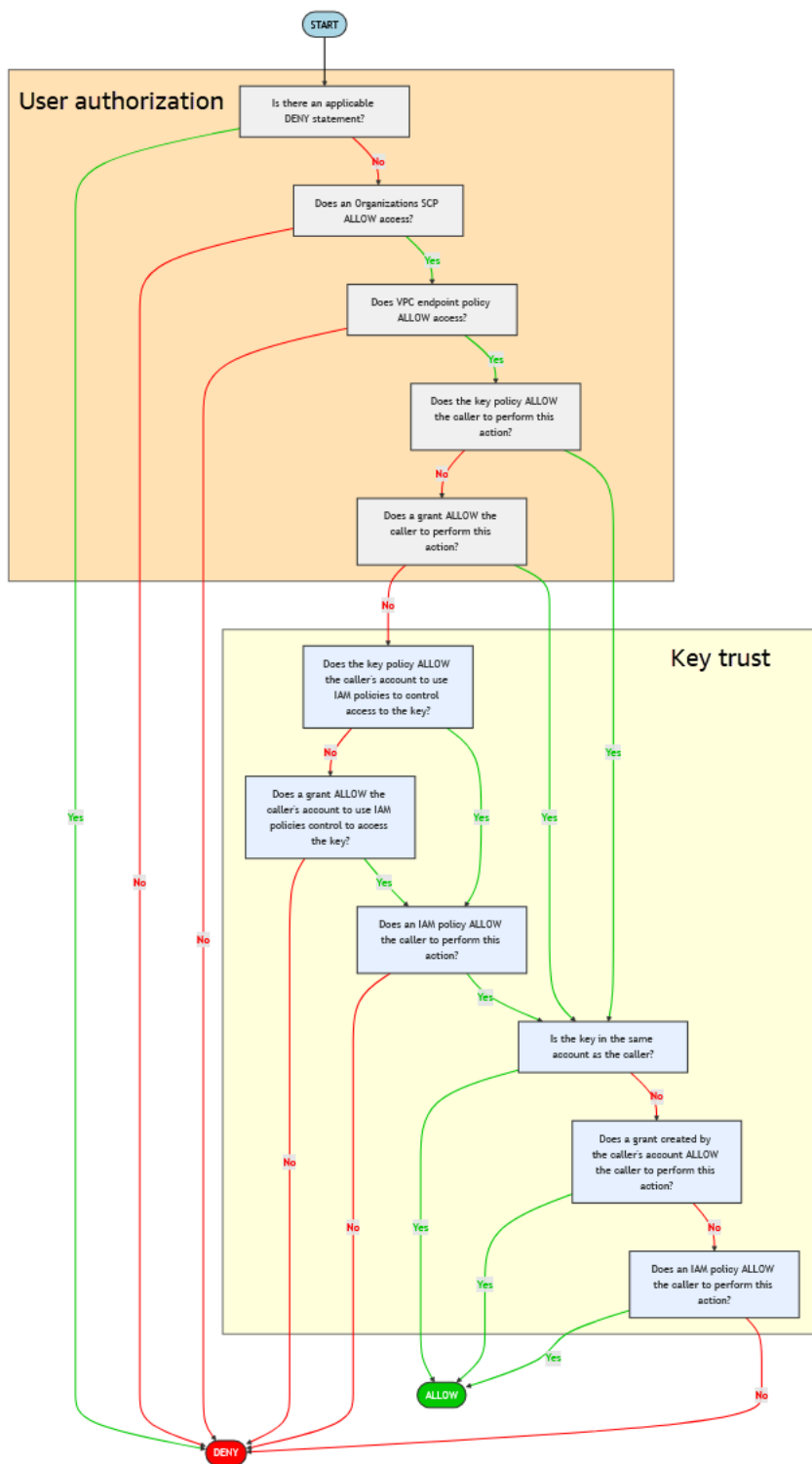
```
$ aws kms create-grant \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
  --operations Decrypt \
  --dry-run
```

Risoluzione dei problemi relativi alle autorizzazioni AWS KMS

Quando autorizza l'accesso a una chiave KMS, valuta quanto segue: AWS KMS

- La [policy chiave](#) collegata alla chiave KMS. La politica chiave è sempre definita nella regione Account AWS e che possiede la chiave KMS.
- Tutte le [policy IAM](#) collegate al ruolo o all'utente da cui proviene la richiesta. Le policy IAM che regolano l'uso da parte di un principale di una chiave KMS sono sempre definite nell' Account AWS del principale.
- Tutte le [concessioni](#) applicabili alla chiave KMS.
- Altri tipi di policy che potrebbero essere applicate alla richiesta di utilizzare la chiave KMS, ad esempio le [policy di controllo dei servizi AWS Organizations](#) e le [policy di endpoint VPC](#). Queste policy sono facoltative e consentono tutte le operazioni per impostazione predefinita, ma puoi utilizzarle per limitare le autorizzazioni altrimenti concesse ai principali.

AWS KMS valuta insieme questi meccanismi politici per determinare se l'accesso alla chiave KMS è consentito o negato. A tale scopo, AWS KMS utilizza un processo simile a quello illustrato nel seguente diagramma di flusso. Il seguente diagramma di flusso fornisce una rappresentazione visiva del processo di valutazione delle policy.



Questo diagramma di flusso è diviso in due parti. Le parti appaiono in sequenza, ma sono in genere valutate nello stesso momento.

- L'autorizzazione di utilizzo determina se è consentito l'utilizzo di una chiave KMS in base alla policy delle chiavi, alle policy IAM, alle concessioni e ad altre policy applicabili.
- Attendibilità della chiave determina se è necessario considerare attendibile una chiave KMS che è consentito utilizzare. In generale, ti fidi delle risorse di cui disponi. Account AWS Tuttavia, puoi sentirti sicuro di usare le chiavi KMS anche in un altro modo Account AWS se una policy di concessione o IAM del tuo account ti consente di utilizzare la chiave KMS.

Puoi usare questo diagramma di flusso per scoprire perché a un intermediario è stata concessa o negata l'autorizzazione a usare una chiave KMS. È anche possibile utilizzarlo per valutare le policy e le autorizzazioni. Ad esempio, il diagramma mostra che a un intermediario può essere negato l'accesso tramite una dichiarazione DENY esplicita o tramite l'assenza di una dichiarazione ALLOW esplicita, nella policy della chiave, nella policy IAM o nella concessione.

Il diagramma di flusso è in grado di spiegare alcuni scenari comuni.

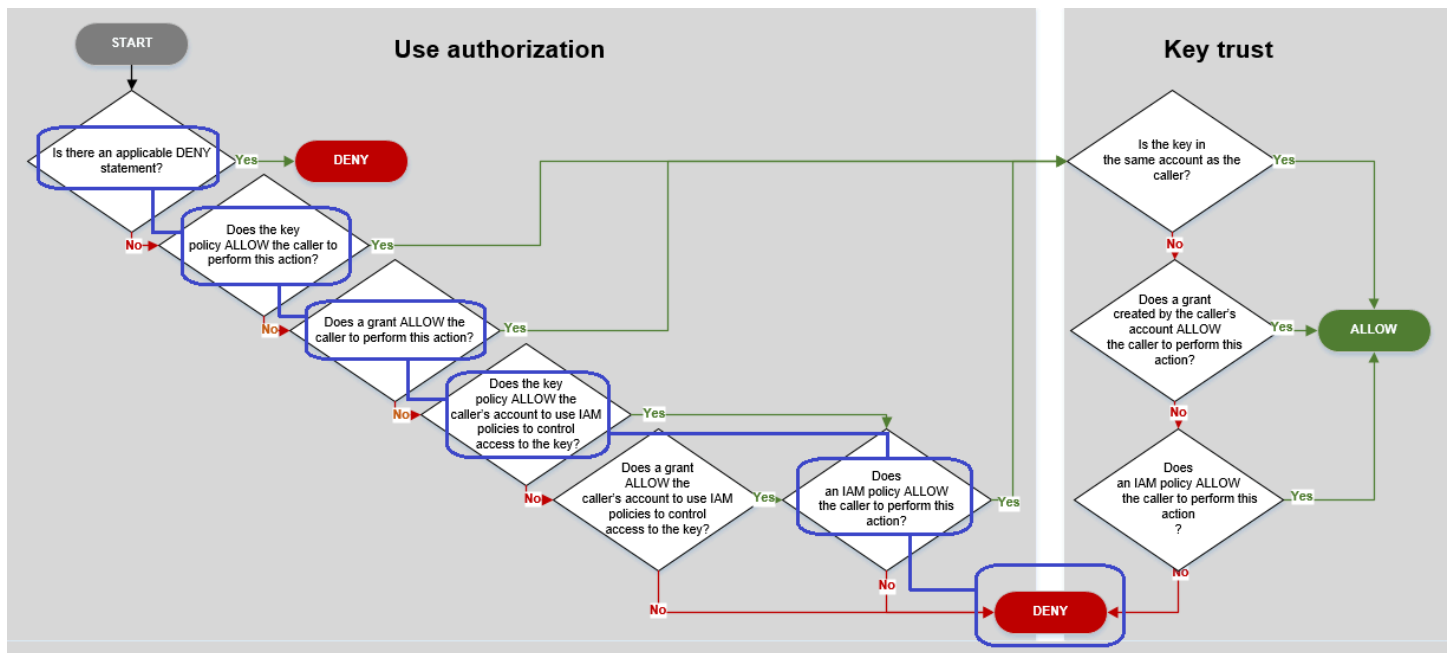
Esempi di autorizzazione

- [Esempio 1: all'utente viene negato l'accesso a una chiave KMS nel proprio Account AWS](#)
- [Esempio 2: L'utente assume un ruolo con il permesso di utilizzare una chiave KMS in un altro Account AWS](#)

Esempio 1: all'utente viene negato l'accesso a una chiave KMS nel proprio Account AWS

Alice è un utente IAM nel Account AWS 111122223333. Le è stato negato l'accesso a una chiave KMS nello stesso Account AWS. Perché Alice non è in grado di utilizzare la chiave KMS?

In questo caso, ad Alice è negato l'accesso alla chiave KMS perché non vi è alcuna policy delle chiavi, policy IAM o concessione che le offre le autorizzazioni necessarie. La politica chiave della chiave KMS consente di utilizzare le politiche IAM Account AWS per controllare l'accesso alla chiave KMS, ma nessuna politica IAM concede ad Alice il permesso di utilizzare la chiave KMS.



Considerare le relative policy per questo esempio.

- La chiave KMS che Alice desidera utilizzare ha la [policy delle chiavi predefinita](#). Questa policy [consente all' Account AWS](#) che possiede la chiave KMS di utilizzare le policy IAM per controllare l'accesso alla chiave KMS. Questa policy soddisfa la condizione nel diagramma La policy della chiave CONSENTE agli intermediari di utilizzare le policy IAM per controllare l'accesso alla chiave?.

JSON

```

{
  "Version": "2012-10-17",
  "Id": "key-test-1",
  "Statement": [ {
    "Sid": "DelegateToIAMPolicies",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:root"
    },
    "Action": "kms:*",
    "Resource": "*"
  } ]
}
  
```

- Non vi è alcuna policy delle chiavi, policy IAM o concessione che consente ad Alice di utilizzare la chiave KMS. Pertanto, ad Alice è negata l'autorizzazione di utilizzare la chiave KMS.

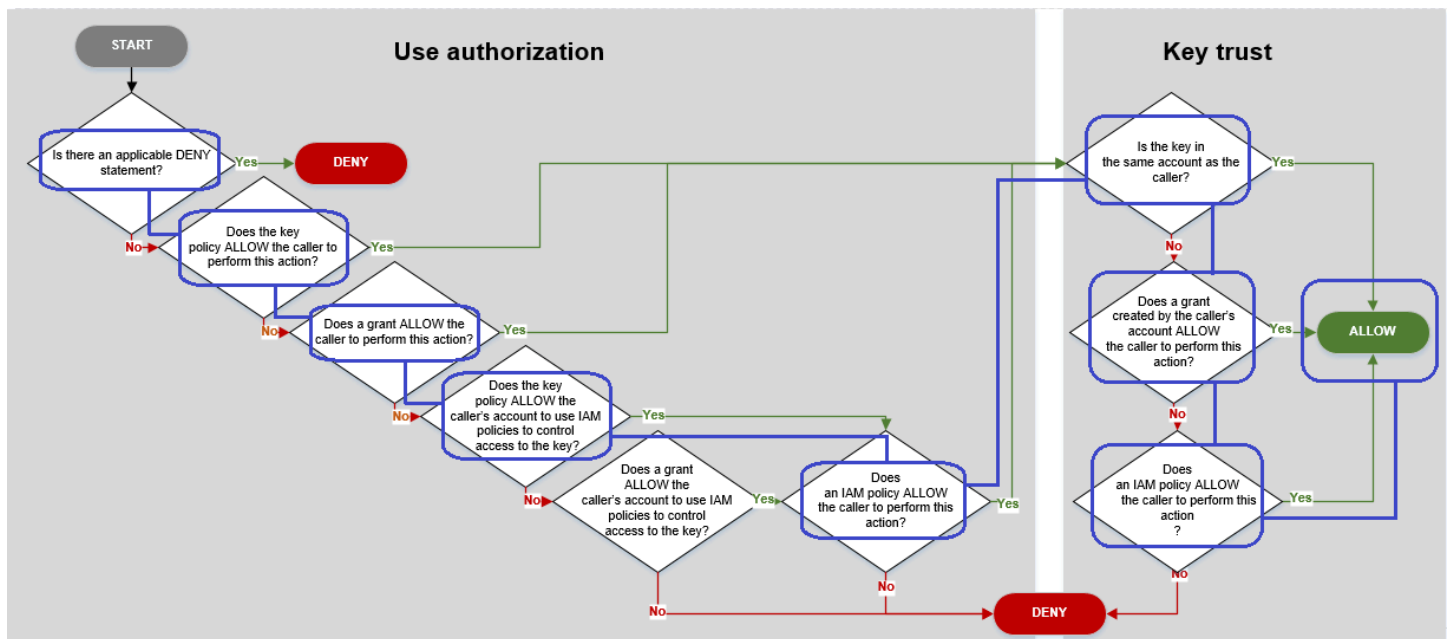
Esempio 2: L'utente assume un ruolo con il permesso di utilizzare una chiave KMS in un altro Account AWS

Bob è un utente nell'account 1 (111122223333). Può utilizzare una chiave KMS nell'account 2 (444455556666) nelle [operazioni di crittografia](#). In che modo è possibile?

Tip

Durante la valutazione di autorizzazioni tra più account, ricordare che la policy della chiave è specificata nell'account della chiave KMS. La policy IAM è specificato nell'account dell'intermediario, anche quando l'autore si trova in un altro account. Per informazioni dettagliate su come fornire l'accesso tra account alle chiavi KMS, consulta [Autorizzazione per gli utenti in altri account di utilizzare una chiave KMS](#).

- La policy delle chiavi per la chiave KMS consente all'account 2 di utilizzare le policy IAM per controllare l'accesso alla chiave KMS.
- La policy chiave per la chiave KMS nell'account 2 consente all'account 1 di utilizzare la chiave KMS nelle operazioni di crittografia. Tuttavia, l'account 1 deve utilizzare le policy IAM in modo da fornire ai principali l'accesso alla chiave KMS.
- Una policy IAM nell'account 1 consente al ruolo Engineering di utilizzare la chiave KMS nell'account 2 per operazioni di crittografia.
- Bob, un utente nell'account 1, ha l'autorizzazione per assumere il ruolo Engineering.
- Bob può considerare affidabile questa chiave KMS, anche se non si trova nel suo account, una policy IAM nel suo account gli fornisce un'autorizzazione esplicita a utilizzare questa chiave KMS.



Vediamo le policy che consentono a Bob, un utente nell'account 1, di utilizzare la chiave KMS nell'account 2.

- La policy delle chiavi per la chiave KMS consente all'account 2 (444455556666, l'account proprietario della chiave KMS) di utilizzare le policy IAM per controllare l'accesso alla chiave KMS. Questa policy delle chiavi consente inoltre all'account 1 (111122223333) di utilizzare la chiave KMS nelle operazioni di crittografia (specificate nell'elemento `Action` dell'istruzione della policy). Tuttavia, nessuno nell'account 1 può utilizzare la chiave KMS nell'account 2 finché l'account 1 definisce le policy IAM che offrono ai principali l'accesso alla chiave KMS.

Nel diagramma di flusso, questa policy della chiave nell'account 2 soddisfa la condizione La policy della chiave CONSENTE all'account dell'intermediario di utilizzare le policy IAM per controllare l'accesso alla chiave?.

JSON

```
{
  "Id": "key-policy-acct-2",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PermissionUseIAMpolicies",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::444455556666:root"
      }
    }
  ]
}
```

```

    },
    "Action": "kms:*",
    "Resource": "*"
  },
  {
    "Sid": "AllowAccount1UseThisKMSKeys",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:root"
    },
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncryptFrom",
      "kms:ReEncryptTo",
      "kms:GenerateDataKey",
      "kms:GenerateDataKeyWithoutPlaintext",
      "kms:DescribeKey"
    ],
    "Resource": "*"
  }
]
}

```

- Una policy IAM del chiamante Account AWS (account 1, 111122223333) fornisce l'autorizzazione principale per eseguire operazioni crittografiche utilizzando la chiave KMS nell'account 2 (444455556666). L'elemento `Action` delega al principale le stesse autorizzazioni che la policy della chiave nell'account 2 ha fornito all'account 1. Per dare queste autorizzazioni al ruolo Engineering nell'account 1, [questa policy in linea è incorporata](#) nel ruolo Engineering.

Le policy IAM tra più account come questa sono efficaci solo quando la policy della chiave per la chiave KMS nell'account 2 offre all'account 1 l'autorizzazione per utilizzare la chiave KMS. Inoltre, l'account 1 può dare ai propri principal solo l'autorizzazione per eseguire le azioni che la policy della chiave ha dato all'account.

Nel diagramma di flusso, questo soddisfa la condizione Una policy IAM consente all'intermediario di eseguire questa azione?.

JSON

```

{
  "Version": "2012-10-17",

```



```

    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "kms:Encrypt",
          "kms:Decrypt",
          "kms:ReEncryptFrom",
          "kms:ReEncryptTo",
          "kms:GenerateDataKey",
          "kms:GenerateDataKeyWithoutPlaintext",
          "kms:DescribeKey"
        ],
        "Resource": [
          "arn:aws:kms:us-west-2:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        ]
      }
    ]
  }
}

```

- Ultimo elemento obbligatorio è la definizione del ruolo Engineering nell'account 1. Il AssumeRolePolicyDocument nel ruolo consente a Bob di assumere il ruolo Engineering.

```

{
  "Role": {
    "Arn": "arn:aws:iam::111122223333:role/Engineering",
    "CreateDate": "2019-05-16T00:09:25Z",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": {
        "Principal": {
          "AWS": "arn:aws:iam::111122223333:user/bob"
        },
        "Effect": "Allow",
        "Action": "sts:AssumeRole"
      }
    },
    "Path": "/",
    "RoleName": "Engineering",
    "RoleId": "AR0A4KJY2TU23Y7NK62MV"
  }
}

```

AWS KMS glossario sul controllo degli accessi

L'argomento seguente descrive termini e concetti importanti nel controllo degli AWS KMS accessi.

Autenticazione

L'autenticazione è il processo di verifica della tua identità. Per inviare una richiesta a AWS KMS, devi accedere AWS utilizzando AWS le tue credenziali.

Autorizzazione

L'autorizzazione fornisce l'autorizzazione a inviare richieste per creare, gestire o utilizzare AWS KMS risorse. Ad esempio, devi avere l'autorizzazione per utilizzare una chiave KMS in un'operazione di crittografia.

Per controllare l'accesso alle AWS KMS risorse, utilizza le [policy chiave](#), le [policy IAM](#) e le [sovvenzioni](#). Ogni chiave KMS deve avere una policy delle chiavi. Se la policy delle chiavi lo consente, puoi anche utilizzare le policy IAM e le concessioni per consentire ai principali l'accesso alla chiave KMS. Per affinare l'autorizzazione, puoi utilizzare le [chiavi di condizione](#) che consentono o negano l'accesso solo quando una richiesta o una risorsa soddisfa le condizioni specificate. Puoi permettere l'accesso ai principali attendibili in [altri Account AWS](#).

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi autenticarti come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come AWS IAM Identity Center (IAM Identity Center), autenticazione Single Sign-On o credenziali. Google/Facebook Per ulteriori informazioni sull'accesso, consulta [Come accedere all' Account AWS](#) nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta [AWS Signature Version 4 per le richieste API](#) nella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Consigliamo vivamente di non

utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali come utente root, consulta [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente IAM.

Identità federata

Come procedura ottimale, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory aziendale, del provider di identità Web o Directory Service che accede Servizi AWS utilizzando le credenziali di una fonte di identità. Le identità federate assumono ruoli che forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Per ulteriori informazioni, consulta [Cos'è IAM Identity Center?](#) nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un [utente IAM](#) è un'identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta [Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee](#) nella Guida per l'utente IAM.

Un [gruppo IAM](#) specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta [Casi d'uso per utenti IAM](#) nella Guida per l'utente IAM.

Ruoli IAM

Un [ruolo IAM](#) è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo [passando da un ruolo utente a un ruolo IAM \(console\)](#) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta [Metodi per assumere un ruolo](#) nella Guida per l'utente di IAM.

I ruoli IAM sono utili per l'accesso federato degli utenti, le autorizzazioni utente IAM temporanee, l'accesso tra account, l'accesso tra servizi e le applicazioni in esecuzione su Amazon. EC2 Per maggiori informazioni, consultare [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e collegandole a identità o risorse. AWS Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consultare [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente IAM.

Le policy basate sull'identità possono essere policy in linea (incorporate direttamente in una singola identità) o policy gestite (policy autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consultare [Scelta fra policy gestite e policy inline](#) nella Guida per l'utente IAM.

Policy basate sulle risorse

Una [politica AWS KMS chiave è una politica](#) basata sulle risorse che controlla l'accesso a una chiave KMS. Ogni chiave KMS deve avere una policy delle chiavi. Puoi utilizzare un altro meccanismo di autorizzazione per consentire l'accesso alla chiave KMS, ma solo se la policy delle chiavi lo consente. (Puoi utilizzare una policy IAM per negare l'accesso a una chiave KMS anche se la policy delle chiavi non lo consente esplicitamente).

Le policy basate sulle risorse sono documenti di policy JSON che colleghi a una risorsa, come una chiave KMS, per controllare l'accesso a una risorsa specifica. Le policy basate su risorse stabiliscono quali operazioni uno specifico principale può eseguire, su quale risorsa e in quali condizioni. Non

si specifica la risorsa in una politica basata sulle risorse, ma è necessario specificare un principale, ad esempio account, utenti, ruoli, utenti federati o. Servizi AWS Le policy basate sulle risorse sono policy inline che si trovano nel servizio che gestisce la risorsa. Non è possibile utilizzare le policy AWS gestite di IAM, come la [policy AWSKeyManagementServicePowerUser gestita, in una policy basata sulle](#) risorse.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- Limiti delle autorizzazioni: impostano il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente IAM.
- Politiche di controllo del servizio (SCPs): specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa in. AWS Organizations Per ulteriori informazioni, consultare [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations .
- Politiche di controllo delle risorse (RCPs): imposta le autorizzazioni massime disponibili per le risorse nei tuoi account. Per ulteriori informazioni, consulta [Politiche di controllo delle risorse \(RCPs\)](#) nella Guida per l'AWS Organizations utente.
- Le policy di sessione sono policy avanzate che si passano come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare [Policy di sessione](#) nella Guida per l'utente IAM.

Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle policy](#) nella IAM User Guide.

AWS KMS risorse

Nel AWS KMS, la risorsa principale è un AWS KMS key. AWS KMS supporta anche un [alias](#), una risorsa indipendente che fornisce un nome descrittivo per una chiave KMS. Alcune AWS KMS operazioni consentono di utilizzare un alias per identificare una chiave KMS.

Ogni istanza di una chiave KMS o un alias ha un [Amazon Resource Name](#) (ARN) univoco con un formato standard. Nelle AWS KMS risorse, il nome del AWS servizio è. kms

- AWS KMS key

Formato ARN:

arn:AWS partition name:AWS service name:Regione AWS:Account AWS ID:key/key ID

Esempio di ARN:

`arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`

- Alias

Formato ARN:

arn:AWS partition name:AWS service name:Regione AWS:Account AWS ID:alias/alias name

Esempio di ARN:

`arn:aws:kms:us-west-2:111122223333:alias/example-alias`

AWS KMS fornisce una serie di operazioni API per lavorare con AWS KMS le tue risorse. Per ulteriori informazioni sull'identificazione delle chiavi KMS nelle operazioni Console di gestione AWS e sulle AWS KMS API, consulta [Identificatori chiave \(\) KeyId](#). Per un elenco delle AWS KMS operazioni, consulta l'[AWS Key Management Service API Reference](#).

Crea una chiave KMS

È possibile creare AWS KMS keys in o utilizzando l'[CreateKey](#) operazione o la [AWS::KMS::Key](#) [AWS CloudFormation risorsa](#). Console di gestione AWS Durante questo processo, imposti la politica chiave per la chiave KMS, che puoi modificare in qualsiasi momento. Seleziona anche i seguenti valori che definiscono il tipo di chiave KMS che crei. Una volta creata la chiave KMS, non è più possibile modificare queste proprietà.

Tipo di chiave KMS

Il tipo di chiave è una proprietà che determina il tipo di chiave crittografica creata. AWS KMS offre tre tipi di chiavi per proteggere i dati:

- chiavi simmetriche Advanced Encryption Standard (AES)

Chiavi a 256 bit utilizzate nella modalità Galois Counter Mode (GCM) di AES per fornire dati autenticati encryption/decryption di dimensioni inferiori a 4 KB. Questo è il tipo di chiave più comune e viene utilizzato per proteggere altre chiavi di crittografia dei dati utilizzate nelle applicazioni e quindi crittografare i dati per Servizi AWS conto dell'utente.

- Chiavi RSA, a curva ellittica o (solo per le regioni SM2 della Cina) asimmetriche

Queste chiavi sono disponibili in varie dimensioni e supportano molti algoritmi. Possono essere utilizzate per la crittografia e la decrittografia, la firma e la verifica o per derivare operazioni segrete condivise a seconda dell'algoritmo scelto.

- Chiavi simmetriche per eseguire operazioni con codici di autenticazione dei messaggi (HMAC) basati su hash

Queste chiavi sono chiavi a 256 bit utilizzate per le operazioni di firma e verifica.

Le chiavi KMS non possono essere esportate dal servizio in testo normale. Sono generate e possono essere utilizzate solo all'interno dei moduli di sicurezza hardware (HSMs) utilizzati dal servizio. Questa è la proprietà di sicurezza fondamentale di AWS KMS garantire che le chiavi non vengano compromesse.

Utilizzo delle chiavi

L'utilizzo della chiave è una proprietà che determina le operazioni di crittografia supportate dalla chiave. Le chiavi KMS possono utilizzare come chiave `ENCRYPT_DECRYPT`, `SIGN_VERIFY`, `GENERATE_VERIFY_MAC`, o `KEY_AGREEMENT`. Ogni chiave KMS può avere solo

un utilizzo. Ciò segue le migliori pratiche di utilizzo delle chiavi secondo la [pubblicazione speciale 800-57 del National Institute of Standards and Technology \(NIST\), Raccomandazioni per la gestione delle chiavi, Sezione 5.2](#), Utilizzo delle chiavi. L'utilizzo di una chiave KMS per più di un tipo di operazione rende il prodotto di entrambe le operazioni più vulnerabile agli attacchi.

Specifiche chiave

Key spec è una proprietà che rappresenta la configurazione crittografica della chiave. Il significato delle specifiche della chiave differisce dal tipo di chiave.

Per le chiavi KMS, la specifica chiave determina se la chiave KMS è simmetrica o asimmetrica. Determina anche il tipo di materiale della chiave e gli algoritmi supportati.

La specifica della chiave predefinita, [SYMMETRIC_DEFAULT](#), rappresenta una chiave crittografica simmetrica a 256 bit. Per una descrizione dettagliata di tutte le specifiche chiave supportate, consulta [Riferimento alle specifiche chiave](#)

Origine dei materiali chiave

L'origine del materiale chiave è una proprietà chiave KMS che identifica l'origine del materiale chiave nella chiave KMS. L'origine del materiale della chiave viene scelta in fase di creazione della chiave KMS e non può essere modificata in seguito. L'origine del materiale della chiave influisce sulle caratteristiche di sicurezza, durata, disponibilità, latenza e velocità di trasmissione effettiva della chiave KMS.

Ogni chiave KMS include un riferimento al relativo materiale della chiave nei suoi metadati. L'origine del materiale della chiave delle chiavi KMS di crittografia simmetrica può variare. È possibile utilizzare materiale chiave AWS KMS generato, materiale chiave generato in un [archivio chiavi personalizzato](#) o [importare materiale chiave personalizzato](#).

Per impostazione predefinita, ogni chiave KMS dispone di materiale chiave univoco. Tuttavia, è possibile creare un set di [chiavi per più regioni](#) con lo stesso materiale chiave.

Le chiavi KMS possono avere uno dei seguenti valori chiave di origine del materiale: AWS_KMS, EXTERNAL ([materiale chiave importato](#)), AWS_CLOUDHSM ([chiave KMS in un archivio AWS CloudHSM chiavi](#)) o EXTERNAL_KEY_STORE ([chiave KMS in un archivio chiavi esterno](#)).

Argomenti

- [Autorizzazioni per la creazione di chiavi KMS](#)
- [Scelta del tipo di chiave KMS da creare](#)

- [Creazione di una chiave KMS di crittografia simmetrica](#)
- [Creazione di una chiave KMS asimmetrica](#)
- [Creazione di una chiave KMS HMAC](#)
- [Crea chiavi primarie multiregionali](#)
- [Crea chiavi di replica multiregionali](#)
- [Crea una chiave KMS con materiale chiave importato](#)
- [Crea una chiave KMS in un AWS CloudHSM archivio di chiavi](#)
- [Crea una chiave KMS in archivi di chiavi esterni](#)

Autorizzazioni per la creazione di chiavi KMS

Per creare una chiave KMS nella console o utilizzando il APIs, devi disporre della seguente autorizzazione in una policy IAM. Quando possibile, utilizzare le [chiavi di condizione](#) per limitare le autorizzazioni. Ad esempio, puoi utilizzare la chiave [kms: KeySpec](#) condition in una policy IAM per consentire ai principali di creare solo chiavi di crittografia simmetriche.

Per un esempio di policy IAM per le entità principali che creano chiavi, vedere [Consentire a un utente di creare chiavi KMS](#).

Note

Presta attenzione quando concedi ai principali l'autorizzazione per gestire tag e alias. Modificando un tag o un alias puoi consentire o negare l'autorizzazione alla chiave gestita dal cliente. Per informazioni dettagliate, vedi [ABAC per AWS KMS](#).

- [kms:](#) è obbligatorio. `CreateKey`
- [kms: CreateAlias](#) è necessario per creare una chiave KMS nella console in cui è richiesto un alias per ogni nuova chiave KMS.
- [kms: TagResource](#) è necessario per aggiungere tag durante la creazione della chiave KMS.
- [iam: CreateServiceLinkedRole](#) è necessario per creare chiavi primarie multiregionali. Per informazioni dettagliate, vedi [Controlla l'accesso alle chiavi multiregionali](#).

Il [kms:](#) non è richiesta `PutKeyPolicy` l'autorizzazione per creare la chiave KMS. L'autorizzazione `kms:CreateKey` include l'autorizzazione per impostare la policy chiave iniziale. Tuttavia, è

necessario aggiungere questa autorizzazione alla policy chiave durante la creazione della chiave KMS per assicurarsi di poter controllare l'accesso alla chiave KMS. L'alternativa è utilizzare il [BypassLockoutSafetyCheck](#) parametro, che non è consigliato.

Le chiavi KMS appartengono all' AWS account in cui sono state create. L'utente IAM che crea una chiave KMS non è considerato il proprietario della chiave e non dispone automaticamente dell'autorizzazione a utilizzare o gestire la chiave KMS creata. Come qualsiasi altro principale, il creatore di chiavi deve ottenere l'autorizzazione tramite una policy chiave, una policy IAM o una concessione. Tuttavia, i principali che hanno l'autorizzazione `kms:CreateKey` possono impostare la policy chiave iniziale e concedersi l'autorizzazione all'utilizzo o alla gestione della chiave.

Scelta del tipo di chiave KMS da creare

Il tipo di chiave KMS che crei dipende in gran parte dal modo in cui intendi utilizzare la chiave KMS, dai requisiti di sicurezza e dai requisiti di autorizzazione. Il tipo di chiave e l'utilizzo della chiave KMS determinano le operazioni crittografiche che la chiave può eseguire. Ogni chiave KMS ha un solo utilizzo. L'utilizzo di una chiave KMS per più di un tipo di operazioni rende il prodotto di tutte le operazioni più vulnerabile agli attacchi.

Per consentire ai principali di creare chiavi KMS solo per un particolare utilizzo di chiavi, usa la chiave [kms:condition](#). Puoi inoltre utilizzare la `kms:KeyUsage` chiave di condizione per consentire ai principali di chiamare operazioni API per una chiave KMS in base al relativo utilizzo della chiave. Ad esempio, puoi consentire l'autorizzazione a disabilitare una chiave KMS solo se l'utilizzo della chiave è `SIGN_VERIFY`.

Utilizza le indicazioni che seguono per determinare quale tipo di chiave KMS è necessario in base al caso d'uso.

Crittografia e decrittografia dei dati

Utilizza una [chiave KMS simmetrica](#) per la maggior parte dei casi d'uso che richiedono la crittografia e la decrittazione dei dati. L'algoritmo di crittografia simmetrica utilizzato da AWS KMS è veloce, efficiente e garantisce la riservatezza e l'autenticità dei dati. Supporta la crittografia autenticata con dati autenticati aggiuntivi (AAD), definiti come [contesto di crittografia](#). Questo tipo di chiave KMS richiede che sia il mittente che il destinatario dei dati crittografati dispongano di credenziali valide per la chiamata. AWS KMS

Se il tuo caso d'uso richiede la crittografia al di fuori degli AWS utenti che non possono effettuare chiamate AWS KMS, le chiavi [KMS asimmetriche](#) sono una buona scelta. Puoi distribuire la parte

pubblica della chiave KMS asimmetrica per consentire a questi utenti di crittografare i dati. E le applicazioni che devono decriptare quei dati possono utilizzare la parte privata della chiave KMS asimmetrica all'interno di AWS KMS.

Firma dei messaggi e verifica delle firme

Per firmare i messaggi e verificare le firme, è necessario utilizzare una [chiave KMS asimmetrica](#). È possibile utilizzare una chiave KMS con una [specifica chiave](#) che rappresenti una coppia di chiavi RSA, una coppia di chiavi a curva ellittica (ECC), una coppia di chiavi ML-DSA o una coppia di chiavi (solo regioni della Cina). SM2 La specifica della chiave scelta è determinata dall'algoritmo di firma che desideri utilizzare. Gli algoritmi di firma ECDSA supportati dalle coppie di chiavi ECC sono consigliati rispetto agli algoritmi di firma RSA. Usa una coppia di chiavi ML-DSA per la migrazione da chiavi RSA o ECC a chiavi post-quantistiche. Tuttavia, potrebbe essere necessario utilizzare una determinata specifica della chiave e un algoritmo di firma per supportare gli utenti che verificano le firme all'esterno di AWS.

Crittografa con coppie di chiavi asimmetriche

[Per crittografare i dati con una coppia di chiavi asimmetrica, devi utilizzare una chiave KMS asimmetrica con una specifica chiave RSA o una specifica chiave \(solo regioni della Cina\). SM2](#)

Per crittografare i dati in AWS KMS con la chiave pubblica di una coppia di chiavi KMS, utilizza l'operazione [Encrypt](#) (crittografa). Puoi anche [scaricare la chiave pubblica e condividerla con le parti che](#) devono crittografare i dati all'esterno. AWS KMS

Quando scarichi la chiave pubblica di una chiave KMS asimmetrica, puoi utilizzarla all'esterno di AWS KMS. Ma non è più soggetto ai controlli di sicurezza che proteggono la chiave KMS. AWS KMS Ad esempio, non è possibile utilizzare politiche o concessioni AWS KMS chiave per controllare l'uso della chiave pubblica. Né è possibile controllare se la chiave viene utilizzata solo per la crittografia e la decrittografia utilizzando gli algoritmi di crittografia supportati. AWS KMS Per ulteriori dettagli, consulta la pagina sulle [considerazioni speciali per il download delle chiavi pubbliche](#).

[Per decrittografare i dati che sono stati crittografati con la chiave pubblica esterna a AWS KMS, chiamate l'operazione Decrypt.](#) L'Decryptoperazione ha esito negativo se i dati sono stati crittografati con una chiave pubblica da una chiave KMS con una chiave utilizzata di. SIGN_VERIFY Inoltre fallirà se è stata crittografata utilizzando un algoritmo che AWS KMS non supporta la specifica chiave selezionata. Per ulteriori informazioni sulle specifiche chiave e sugli algoritmi supportati, consulta. [Riferimento alle specifiche chiave](#)

Per evitare questi errori, chiunque utilizzi una chiave pubblica esterna a AWS KMS deve memorizzare la configurazione della chiave. La AWS KMS console e la [GetPublicKey](#) risposta forniscono le informazioni da includere quando si condivide la chiave pubblica.

Ricava segreti condivisi

Per ricavare segreti condivisi, usa una chiave KMS con [curva ellittica standard del NIST](#) o materiale chiave [SM2](#) (solo per le regioni della Cina). AWS KMS utilizza l'[Elliptic Curve Cryptography Cofactor Diffie-Hellman Primitive](#) (ECDH) per stabilire un accordo chiave tra due peer derivando un segreto condiviso dalle rispettive coppie di chiavi pubblico-private a curva ellittica. È possibile utilizzare il segreto condiviso non elaborato restituito dall'[DeriveSharedSecret](#) operazione per derivare una chiave simmetrica in grado di crittografare e decrittografare i dati inviati tra due parti oppure generare e verificare. HMACs AWS KMS consiglia di seguire le [raccomandazioni del NIST per la derivazione delle chiavi quando si utilizza il segreto condiviso non elaborato per derivare](#) una chiave simmetrica.

Generazione e verifica dei codici HMAC

Per generare e verificare i codici di autenticazione dei messaggi basati su hash, utilizza una chiave KMS HMAC. Quando si crea una chiave HMAC AWS KMS, AWS KMS crea e protegge il materiale chiave e si assicura di utilizzare gli algoritmi MAC corretti per la chiave. I codici HMAC possono essere utilizzati anche come numeri pseudo-casuali e, in alcuni scenari, per la firma simmetrica e la tokenizzazione.

Le chiavi KMS HMAC sono chiavi simmetriche. Quando crei una chiave KMS HMAC nella console AWS KMS, scegli il tipo di chiave `Symmetric`.

Utilizzo con i servizi AWS

Per creare una chiave KMS da utilizzare con un [AWS servizio integrato AWS KMS](#), consulta la documentazione del servizio. AWS i servizi che crittografano i dati richiedono una chiave KMS di crittografia [simmetrica](#).

Oltre a queste considerazioni, le operazioni crittografiche sulle chiavi KMS con specifiche della chiave diverse hanno prezzi e quote di richieste differenti. Per informazioni sui prezzi di AWS KMS, consulta la pagina dei [prezzi di AWS Key Management Service](#). Per informazioni sulle quote di richieste, consulta [Quote di richieste](#).

Creazione di una chiave KMS di crittografia simmetrica

Questo argomento spiega come creare la chiave KMS di base, una chiave KMS di [crittografia simmetrica per una singola regione con materiale chiave proveniente](#) da. AWS KMS Puoi utilizzare questa chiave KMS per proteggere le tue risorse in un Servizio AWS.

[È possibile creare chiavi KMS di crittografia simmetrica nella AWS KMS console, utilizzando l'CreateKeyAPI o utilizzando il modello. AWS::KMS::Key CloudFormation](#)

La specifica chiave predefinita, [SYMMETRIC_DEFAULT](#), è la [specificata chiave per le chiavi KMS di crittografia simmetrica](#). Quando si seleziona il tipo di chiave Symmetric e l'utilizzo della chiave di crittografia e decrittografia nella console, viene selezionata la specifica della chiave. AWS KMS SYMMETRIC_DEFAULT Nell'[CreateKey](#) operazione, se non si specifica un valore, viene selezionato SYMMETRIC_DEFAULT. KeySpec Se non hai motivo di utilizzare una specifica della chiave diversa, SYMMETRIC_DEFAULT è una scelta valida.

Per informazioni sulle quote applicabili alle chiavi KMS, consulta [Quote](#).

Utilizzo della console AWS KMS

Puoi usare Console di gestione AWS per creare AWS KMS keys (chiavi KMS).

Important

Non includere informazioni riservate o sensibili nell'alias, nella descrizione o nei tag. Questi campi possono apparire in testo semplice nei CloudTrail log e in altri output.

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
4. Scegliere Create key (Crea chiave).
5. Per creare una chiave KMS di crittografia simmetrica, in Key type (Tipo di chiave) scegli Symmetric (Simmetrica).
6. In Utilizzo della chiave, l'opzione Crittografia e decrittografia è selezionata per impostazione predefinita.

7. Scegli Next (Successivo).
8. Digita un alias per la chiave KMS. Un nome di alias non può iniziare con **aws/**. Il **aws/** prefisso è riservato da Amazon Web Services per essere rappresentato Chiavi gestite da AWS nel tuo account.

 Note

L'aggiunta, l'eliminazione o l'aggiornamento di un alias può consentire o negare l'autorizzazione alla chiave KMS. Per informazioni dettagliate, consulta [ABAC per AWS KMS](#) e [Usa gli alias per controllare l'accesso alle chiavi KMS](#).

Un alias è un nome visualizzato che può essere utilizzato per identificare la chiave KMS. È consigliabile scegliere un alias che indica il tipo di dati che desideri proteggere o l'applicazione che desideri utilizzare con la chiave KMS.

Gli alias sono obbligatori quando si crea una chiave KMS nella Console di gestione AWS. Sono opzionali quando si utilizza l'[CreateKey](#) operazione.

9. (Facoltativo) Digita una descrizione per la chiave KMS.

Puoi aggiungere una descrizione ora o aggiornarla in qualsiasi momento, a meno che lo [stato della chiave](#) non sia Pending Deletion o Pending Replica Deletion. Per aggiungere, modificare o eliminare la descrizione di una chiave gestita dal cliente esistente, modifica la descrizione nella pagina dei dettagli della chiave KMS inclusa nell'operazione Console di gestione AWS o utilizza l'[UpdateKeyDescription](#) operazione.

10. (Facoltativo) Digitare una chiave di tag e un valore di tag facoltativo. Per aggiungere più di un tag alla chiave KMS scegli Add tag (Aggiungi tag).

 Note

L'applicazione o l'eliminazione di un tag chiave KMS può consentire o negare l'autorizzazione alla chiave KMS. Per informazioni dettagliate, consulta [ABAC per AWS KMS](#) e [Usa i tag per controllare l'accesso alle chiavi KMS](#).

Quando aggiungi tag alle tue AWS risorse, AWS genera un rapporto sull'allocazione dei costi con utilizzo e costi aggregati per tag. I tag possono essere utilizzati anche per controllare

l'accesso a una chiave KMS. Per informazioni sull'assegnazione di tag delle chiavi KMS, consulta [Tag in AWS KMS](#) e [ABAC per AWS KMS](#).

11. Seleziona Next (Successivo).
12. Seleziona i ruoli e gli utenti IAM che possono gestire la chiave KMS.

Note

Questa politica chiave offre il Account AWS pieno controllo di questa chiave KMS. Consente agli amministratori dell'account di utilizzare policy IAM per concedere ad altri principali l'autorizzazione per la gestione della chiave KMS. Per informazioni dettagliate, vedi [the section called "Policy delle chiavi predefinita"](#).

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

La AWS KMS console aggiunge gli amministratori chiave alla politica chiave sotto l'identificatore dell'istruzione. "Allow access for Key Administrators" La modifica di questo identificatore di istruzione potrebbe influire sul modo in cui la console visualizza gli aggiornamenti apportati all'istruzione.

13. (Facoltativo) Per impedire ai ruoli e agli utenti IAM selezionati di eliminare questa chiave KMS, nella sezione Eliminazione chiave nella parte inferiore della pagina, deseleziona la casella di controllo Consenti agli amministratori delle chiavi di eliminare questa chiave.
14. Seleziona Next (Successivo).
15. Selezionare i ruoli e gli utenti IAM che possono utilizzare la chiave nelle [operazioni di crittografia](#).

Note

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

La AWS KMS console aggiunge gli utenti chiave alla politica chiave sotto gli "Allow use of the key" identificatori di dichiarazione e. "Allow attachment of persistent resources" La modifica di questi identificatori delle istruzioni potrebbe influire sul modo in cui la console visualizza gli aggiornamenti apportati all'istruzione.

16. (Facoltativo) È possibile consentire ad altri Account AWS di utilizzare questa chiave KMS per operazioni crittografiche. A tale scopo, nella Account AWS sezione Altro in fondo alla pagina, scegli Aggiungi un altro account Account AWS e inserisci il numero di Account AWS identificazione di un account esterno. Per aggiungere più account esterni, ripetere questo passaggio.

Note

Per consentire ai principali negli account esterni di utilizzare la chiave KMS, gli amministratori dell'account esterno devono creare policy IAM che forniscono tali autorizzazioni. Per ulteriori informazioni, consultare [Autorizzazione per gli utenti in altri account di utilizzare una chiave KMS](#).

17. Scegli Next (Successivo).
18. Consulta le principali dichiarazioni politiche per la chiave. Per apportare modifiche alla politica chiave, seleziona Modifica.
19. Scegli Next (Successivo).
20. Esaminare le principali impostazioni scelte. È comunque possibile tornare indietro e modificare tutte le impostazioni.
21. Scegli Termina per creare la chiave KMS.

Utilizzo dell' AWS KMS API

È possibile utilizzare l'[CreateKey](#) operazione per creare AWS KMS keys di tutti i tipi. Questi esempi utilizzano il [AWS Command Line Interface \(AWS CLI\)](#). Per esempi in più linguaggi di programmazione, consulta [Utilizzo CreateKey con un AWS SDK o una CLI](#).

Important

Non includere informazioni riservate o sensibili nei campi Description o Tags. Questi campi possono apparire in testo semplice nei CloudTrail log e in altri output.

La seguente operazione crea una chiave di crittografia simmetrica in una singola regione supportata da materiale chiave generato da. AWS KMS Questa operazione non include parametri obbligatori. È possibile anche utilizzare il parametro Policy per specificare una policy delle chiavi. È possibile modificare la politica della chiave ([PutKeyPolicy](#)) e aggiungere elementi opzionali, come una

[descrizione](#) e [tag](#) in qualsiasi momento. È inoltre possibile anche creare [chiavi asimmetriche](#), [chiavi multi-regione](#), chiavi con [materiale chiave importato](#) e chiavi in [archivi delle chiavi personalizzate](#). Per creare chiavi di dati per la crittografia lato client, utilizzate l'[GenerateDataKey](#) operazione.

L'[CreateKey](#) operazione non consente di specificare un alias, ma è possibile utilizzare l'[CreateAlias](#) operazione per creare un alias per la nuova chiave KMS.

Di seguito è riportato un esempio di una chiamata all'operazione `CreateKey` senza parametri. Questo comando utilizza tutti i valori predefiniti. Crea una chiave KMS di crittografia simmetrica per crittografare e decrittografare con materiale della chiave generato da AWS KMS.

```
$ aws kms create-key
{
  "KeyMetadata": {
    "Origin": "AWS_KMS",
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "Description": "",
    "KeyManager": "CUSTOMER",
    "Enabled": true,
    "KeySpec": "SYMMETRIC_DEFAULT",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "KeyState": "Enabled",
    "CreationDate": 1502910355.475,
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "MultiRegion": false
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
  },
}
```

Se non specifichi una policy delle chiavi per la nuova chiave KMS, la [policy delle chiavi predefinita](#) che `CreateKey` applica è diversa dalla policy delle chiavi predefinita che la console applica quando la utilizzi per creare una nuova chiave KMS.

Ad esempio, questa chiamata all'[GetKeyPolicy](#) operazione restituisce la politica chiave applicabile. `CreateKey` Fornisce l' Account AWS accesso alla chiave KMS e le consente di creare politiche AWS Identity and Access Management (IAM) per la chiave KMS. Per informazioni dettagliate sulle policy IAM e sulle policy delle chiavi KMS, consulta [Accesso e autorizzazioni alle chiavi KMS](#).

```
$ aws kms get-key-policy --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --policy-name default --output text
```

JSON

```
{
  "Version": "2012-10-17",
  "Id" : "key-default-1",
  "Statement" : [ {
    "Sid" : "EnableIAMUserPermissions",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : "kms:*",
    "Resource" : "*"
  } ]
}
```

Creazione di una chiave KMS asimmetrica

[Puoi creare chiavi KMS asimmetriche nella AWS KMS console, utilizzando l'API o utilizzando il CreateKeymodello. AWS::KMS::Key CloudFormation](#) Una chiave KMS asimmetrica rappresenta una coppia di chiavi pubblica e privata che può essere utilizzata per la crittografia, la firma o la derivazione di segreti condivisi. La chiave privata rimane all'interno. AWS KMS Per scaricare la chiave pubblica da utilizzare all'esterno AWS KMS, vedere [Scarica la chiave pubblica](#).

Quando crei una chiave KMS asimmetrica, devi selezionare una specifica chiave. Spesso la scelta delle specifiche chiave è determinata da requisiti normativi, di sicurezza o aziendali. Potrebbe essere influenzata anche dalla dimensione dei messaggi che è necessario crittografare o firmare. In generale, le chiavi di crittografia più lunghe sono più resistenti agli attacchi a forza bruta. Per una descrizione dettagliata di tutte le specifiche chiave supportate, consulta [Riferimento alle specifiche chiave](#)

AWS i servizi che si integrano con AWS KMS non supportano le chiavi KMS asimmetriche. Se desideri creare una chiave KMS che crittografa i dati archiviati o gestiti in un AWS servizio, [crea](#) una chiave KMS di crittografia simmetrica.

Per informazioni sulle autorizzazioni richieste per la creazione di chiavi KMS, consultare [Autorizzazioni per la creazione di chiavi KMS](#).

Utilizzo della console AWS KMS

È possibile utilizzare il Console di gestione AWS per creare elementi asimmetrici AWS KMS keys (chiavi KMS). Ogni chiave KMS asimmetrica rappresenta una coppia di chiavi pubbliche e private.

Important

Non includere informazioni riservate o sensibili nell'alias, nella descrizione o nei tag. Questi campi possono apparire in testo semplice nei CloudTrail log e in altri output.

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
4. Scegli Crea chiave.
5. Per creare una chiave KMS asimmetrica, in Key type (Tipo di chiave) scegli Asymmetric (Asimmetrica).
6. Per creare una chiave KMS asimmetrica per la crittografia a chiave pubblica, in Key usage (Utilizzo chiave) scegliere Encrypt and decrypt (Crittografia e decrittazione).

Per creare una chiave KMS asimmetrica per la firma dei messaggi e la verifica delle firme, in Utilizzo delle chiavi, scegli Firma e verifica.

Per creare una chiave KMS asimmetrica per ricavare segreti condivisi, in Utilizzo delle chiavi, scegli Contratto chiave.

Per informazioni sulla scelta di un valore di utilizzo chiave, consulta [Scelta del tipo di chiave KMS da creare](#).

7. Seleziona le specifiche (Key spec (Specifiche chiave)) per la tua chiave KMS asimmetrica.
8. Scegli Next (Successivo).
9. Digita un [alias](#) per la chiave KMS. Un nome di alias non può iniziare con **aws/**. Il prefisso **aws/** è riservato da Amazon Web Services per rappresentare le Chiavi gestite da AWS nel tuo account.

Un alias è un nome descrittivo che puoi usare per identificare la chiave KMS nella console e in alcune console. AWS KMS APIs. È consigliabile scegliere un alias che indica il tipo di dati che desideri proteggere o l'applicazione che desideri utilizzare con la chiave KMS.

Gli alias sono obbligatori quando si crea una chiave KMS nella Console di gestione AWS. Non è possibile specificare un alias quando si utilizza l'[CreateKey](#) operazione, ma è possibile utilizzare la console o l'[CreateAlias](#) operazione per creare un alias per una chiave KMS esistente. Per informazioni dettagliate, vedi [Alias in AWS KMS](#).

10. (Facoltativo) Digita una descrizione per la chiave KMS.

Inserire una descrizione che illustra il tipo di dati che si desidera proteggere o l'applicazione che si desidera utilizzare con la chiave KMS.

Puoi aggiungere una descrizione ora o aggiornarla in qualsiasi momento, a meno che lo [stato della chiave](#) non sia Pending Deletion o Pending Replica Deletion. Per aggiungere, modificare o eliminare la descrizione di una chiave gestita dal cliente esistente, modifica la descrizione nella pagina dei dettagli della chiave KMS inclusa nell'operazione Console di gestione AWS o utilizza l'operazione. [UpdateKeyDescription](#)

11. (Facoltativo) Digitare una chiave di tag e un valore di tag facoltativo. Per aggiungere più di un tag alla chiave KMS scegli Add tag (Aggiungi tag).

Quando aggiungi tag alle tue AWS risorse, AWS genera un rapporto sull'allocazione dei costi con utilizzo e costi aggregati per tag. I tag possono essere utilizzati anche per controllare l'accesso a una chiave KMS. Per informazioni sull'assegnazione di tag delle chiavi KMS, consulta [Tag in AWS KMS](#) e [ABAC per AWS KMS](#).

12. Seleziona Next (Successivo).
13. Seleziona i ruoli e gli utenti IAM che possono gestire la chiave KMS.

Note

Questa politica chiave offre il Account AWS pieno controllo di questa chiave KMS. Consente agli amministratori dell'account di utilizzare policy IAM per concedere ad altri principali l'autorizzazione per la gestione della chiave KMS. Per informazioni dettagliate, vedi [the section called “Policy delle chiavi predefinita”](#).
Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i

dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

La AWS KMS console aggiunge gli amministratori chiave alla politica chiave sotto l'identificatore dell'istruzione. "Allow access for Key Administrators" La modifica di questo identificatore di istruzione potrebbe influire sul modo in cui la console visualizza gli aggiornamenti apportati all'istruzione.

14. (Facoltativo) Per impedire ai ruoli e agli utenti IAM selezionati di eliminare questa chiave KMS, nella sezione Eliminazione chiave nella parte inferiore della pagina, deseleziona la casella di controllo Consenti agli amministratori delle chiavi di eliminare questa chiave.
15. Scegli Next (Successivo).
16. Seleziona i ruoli e gli utenti IAM che possono utilizzare la chiave KMS per [operazioni di crittografia](#).

Note

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

La AWS KMS console aggiunge gli utenti chiave alla politica chiave sotto gli "Allow use of the key" identificatori di dichiarazione e. "Allow attachment of persistent resources" La modifica di questi identificatori delle istruzioni potrebbe influire sul modo in cui la console visualizza gli aggiornamenti apportati all'istruzione.

17. (Facoltativo) È possibile consentire ad altri Account AWS di utilizzare questa chiave KMS per operazioni crittografiche. A questo proposito, nella sezione Altri Account AWS, nella parte inferiore della pagina, scegli Aggiungi un altro Account AWS e inserisci il numero di identificazione Account AWS di un account esterno. Per aggiungere più account esterni, ripetere questo passaggio.

Note

Per consentire ai principali negli account esterni di utilizzare la chiave KMS, gli amministratori dell'account esterno devono creare policy IAM che forniscono tali

autorizzazioni. Per ulteriori informazioni, consultare [Autorizzazione per gli utenti in altri account di utilizzare una chiave KMS](#).

18. Scegli Next (Successivo).
19. Consulta le principali dichiarazioni politiche relative alla chiave. Per apportare modifiche alla politica chiave, seleziona Modifica.
20. Scegli Next (Successivo).
21. Esaminare le principali impostazioni scelte. È comunque possibile tornare indietro e modificare tutte le impostazioni.
22. Scegli Termina per creare la chiave KMS.

Utilizzo dell' AWS KMS API

È possibile utilizzare l'[CreateKey](#) operazione per creare un'asimmetria AWS KMS key. Questi esempi utilizzano la [AWS Command Line Interface \(AWS CLI\)](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

Quando crei una chiave KMS asimmetrica devi specificare il parametro KeySpec che determina il tipo di chiavi create. Inoltre, è necessario specificare un KeyUsage valore di ENCRYPT_DECRYPT, SIGN_VERIFY o KEY_AGREEMENT. Una volta creata la chiave KMS, non è più possibile modificare queste proprietà.

L'[CreateKey](#) operazione non consente di specificare un alias, ma è possibile utilizzare l'[CreateAlias](#) operazione per creare un alias per la nuova chiave KMS.

Important

Non includere informazioni riservate o sensibili nei campi Description o Tags. Questi campi possono apparire in testo semplice nei CloudTrail log e in altri output.

Crea una coppia di chiavi KMS asimmetrica per la crittografia pubblica

Nell'esempio seguente viene utilizzata l'operazione CreateKey per creare una chiave KMS asimmetrica di chiavi RSA a 4096-bit progettata per la crittografia a chiave pubblica.

```
$ aws kms create-key --key-spec RSA_4096 --key-usage ENCRYPT_DECRYPT
{
```

```

    "KeyMetadata": {
      "KeyState": "Enabled",
      "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "KeyManager": "CUSTOMER",
      "Description": "",
      "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "CreationDate": 1569973196.214,
      "MultiRegion": false,
      "KeySpec": "RSA_4096",
      "CustomerMasterKeySpec": "RSA_4096",
      "KeyUsage": "ENCRYPT_DECRYPT",
      "EncryptionAlgorithms": [
        "RSAES_OAEP_SHA_1",
        "RSAES_OAEP_SHA_256"
      ],
      "AWSAccountId": "111122223333",
      "Origin": "AWS_KMS",
      "Enabled": true
    }
  }
}

```

Crea una coppia di chiavi KMS asimmetrica per la firma e la verifica

Il comando di esempio seguente crea una chiave KMS asimmetrica che rappresenta una coppia di chiavi ECC utilizzate per la firma e la verifica. Non è possibile creare una coppia di chiavi a curva ellittica per la crittografia e la decrittografia.

```

$ aws kms create-key --key-spec ECC_NIST_P521 --key-usage SIGN_VERIFY
{
  "KeyMetadata": {
    "KeyState": "Enabled",
    "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "CreationDate": 1570824817.837,
    "Origin": "AWS_KMS",
    "SigningAlgorithms": [
      "ECDSA_SHA_512"
    ],
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
    "AWSAccountId": "111122223333",
    "KeySpec": "ECC_NIST_P521",
    "CustomerMasterKeySpec": "ECC_NIST_P521",

```

```

    "KeyManager": "CUSTOMER",
    "Description": "",
    "Enabled": true,
    "MultiRegion": false,
    "KeyUsage": "SIGN_VERIFY"
  }
}

```

Crea una coppia di chiavi KMS asimmetrica per derivare segreti condivisi

Il comando di esempio seguente crea una chiave KMS asimmetrica che rappresenta una coppia di chiavi ECDH utilizzate per derivare segreti condivisi. Non è possibile creare una coppia di chiavi a curva ellittica per la crittografia e la decrittografia.

```

$ aws kms create-key --key-spec ECC_NIST_P256 --key-usage KEY_AGREEMENT
{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
    "CreationDate": "2023-12-27T19:10:15.063000+00:00",
    "Enabled": true,
    "Description": "",
    "KeyUsage": "KEY_AGREEMENT",
    "KeyState": "Enabled",
    "Origin": "AWS_KMS",
    "KeyManager": "CUSTOMER",
    "CustomerMasterKeySpec": "ECC_NIST_P256",
    "KeySpec": "ECC_NIST_P256",
    "KeyAgreementAlgorithms": [
      "ECDH"
    ],
    "MultiRegion": false
  }
}

```

Creazione di una chiave KMS HMAC

[Puoi creare chiavi KMS HMAC nella AWS KMS console, utilizzando l>CreateKeyAPI o utilizzando il modello. AWS::KMS::Key CloudFormation](#)

Quando crei una chiave KMS HMAC, devi selezionare una specifica chiave. AWS KMS supporta diverse [specifiche chiave per le chiavi HMAC](#) KMS. La scelta della specifica della chiave può essere determinata da requisiti normativi, di sicurezza o aziendali. In generale, le chiavi più lunghe sono più resistenti agli attacchi a forza bruta.

Per informazioni sulle autorizzazioni richieste per la creazione di chiavi KMS, consultare [Autorizzazioni per la creazione di chiavi KMS](#).

Utilizzo della console AWS KMS

È possibile utilizzare il Console di gestione AWS per creare chiavi KMS HMAC. Le chiavi KMS HMAC sono chiavi simmetriche con un utilizzo della chiave Generate and verify MAC (Genera e verifica MAC). È possibile anche creare chiavi HMAC multi-regione.

1. [Accedi a Console di gestione AWS e apri la console AWS Key Management Service \(AWS KMS\) in /kms. https://console.aws.amazon.com](#)
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
4. Scegli Crea chiave.
5. Alla voce Key type (Tipo di chiave), scegliere Symmetric (Simmetrica).

Le chiavi KMS HMAC sono simmetriche. Usa la stessa chiave per generare e verificare i tag HMAC.

6. Per Key usage (Utilizzo della chiave), scegli Generate and verify MAC (Genera e verifica MAC).

Generate and verify MAC (Genera e verifica MAC) è l'unico utilizzo valido per le chiavi KMS HMAC.

Note

Key usage (Utilizzo della chiave) viene visualizzato per le chiavi simmetriche solo quando le chiavi KMS HMAC sono supportate nella regione selezionata.

7. Seleziona un valore Key spec (Specifica della chiave) per la tua chiave KMS HMAC.

La scelta della specifica della chiave può essere determinata da requisiti normativi, di sicurezza o aziendali. In generale, le chiavi più lunghe sono più sicure.

8. Per creare una chiave HMAC [primaria](#) multi-regione, in Advanced options (Opzioni avanzate) scegli Multi-Region key (Chiave multi-regione). Le [proprietà condivise](#) che definisci per questa chiave KMS, come il tipo di chiave e l'utilizzo della chiave, saranno condivise con le relative chiavi di replica.

Non è possibile utilizzare questa procedura per creare una chiave di replica. Per creare una chiave HMAC di replica multi-regione, segui le [istruzioni per creare una chiave di replica](#).

9. Scegli Next (Successivo).
10. Inserisci un [alias](#) per la chiave KMS. Un nome di alias non può iniziare con **aws/**. Il prefisso **aws/** è riservato da Amazon Web Services per rappresentare le Chiavi gestite da AWS nel tuo account.

Ti consigliamo di utilizzare un alias che identifichi la chiave KMS come chiave HMAC, ad esempio HMAC/test-key. In questo modo sarà più facile identificare le chiavi HMAC nella AWS KMS console, dove è possibile ordinare e filtrare le chiavi in base a tag e alias, ma non in base alle specifiche o all'utilizzo delle chiavi.

Gli alias sono obbligatori quando si crea una chiave KMS nella Console di gestione AWS. Non è possibile specificare un alias quando si utilizza l'[CreateKey](#) operazione, ma è possibile utilizzare la console o l'[CreateAlias](#) operazione per creare un alias per una chiave KMS esistente. Per informazioni dettagliate, vedi [Alias in AWS KMS](#).

11. (Facoltativo) Inserisci una descrizione per la chiave KMS.

Inserire una descrizione che illustra il tipo di dati che si desidera proteggere o l'applicazione che si desidera utilizzare con la chiave KMS.

Puoi aggiungere una descrizione ora o aggiornarla in qualsiasi momento, a meno che lo [stato della chiave](#) non sia Pending Deletion o Pending Replica Deletion. Per aggiungere, modificare o eliminare la descrizione di una chiave gestita dal cliente esistente, modifica la descrizione nella pagina dei dettagli della chiave KMS Console di gestione AWS o utilizza Console di gestione AWS l'operazione. [UpdateKeyDescription](#)

12. (Facoltativo) Inserisci un tag della chiave e un valore facoltativo. Per aggiungere più di un tag alla chiave KMS scegli Add tag (Aggiungi tag).

Puoi anche aggiungere un tag che identifica la chiave come chiave HMAC, ad esempio Type=HMAC. In questo modo sarà più facile identificare le chiavi HMAC nella AWS KMS console, dove è possibile ordinare e filtrare le chiavi in base a tag e alias, ma non in base alle specifiche o all'utilizzo delle chiavi.

Quando aggiungi tag alle tue AWS risorse, AWS genera un rapporto sull'allocazione dei costi con utilizzo e costi aggregati per tag. I tag possono essere utilizzati anche per controllare l'accesso a una chiave KMS. Per informazioni sull'assegnazione di tag delle chiavi KMS, consulta [Tag in AWS KMS](#) e [ABAC per AWS KMS](#).

13. Seleziona Next (Successivo).
14. Seleziona i ruoli e gli utenti IAM che possono gestire la chiave KMS.

Note

Questa politica chiave offre il Account AWS pieno controllo di questa chiave KMS. Consente agli amministratori dell'account di utilizzare policy IAM per concedere ad altri principali l'autorizzazione per la gestione della chiave KMS. Per informazioni dettagliate, vedi [the section called "Policy delle chiavi predefinita"](#).

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

La AWS KMS console aggiunge gli amministratori chiave alla politica chiave sotto l'identificatore dell'istruzione. "Allow access for Key Administrators" La modifica di questo identificatore di istruzione potrebbe influire sul modo in cui la console visualizza gli aggiornamenti apportati all'istruzione.

15. (Facoltativo) Per impedire ai ruoli e agli utenti IAM selezionati di eliminare questa chiave KMS, nella sezione Eliminazione chiave nella parte inferiore della pagina, deseleziona la casella di controllo Consenti agli amministratori delle chiavi di eliminare questa chiave.
16. Scegli Next (Successivo).
17. Seleziona i ruoli e gli utenti IAM che possono utilizzare la chiave KMS per [operazioni di crittografia](#).

Note

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

La AWS KMS console aggiunge gli utenti chiave alla politica chiave sotto gli "Allow use of the key" identificatori di dichiarazione e. "Allow attachment of persistent resources" La modifica di questi identificatori delle istruzioni potrebbe influire sul modo in cui la console visualizza gli aggiornamenti apportati all'istruzione.

18. (Facoltativo) È possibile consentire ad altri Account AWS di utilizzare questa chiave KMS per operazioni crittografiche. A questo proposito, nella sezione Altri Account AWS, nella parte inferiore della pagina, scegli Aggiungi un altro Account AWS e inserisci il numero di identificazione Account AWS di un account esterno. Per aggiungere più account esterni, ripetere questo passaggio.

Note

Per consentire ai principali negli account esterni di utilizzare la chiave KMS, gli amministratori dell'account esterno devono creare policy IAM che forniscono tali autorizzazioni. Per ulteriori informazioni, consultare [Autorizzazione per gli utenti in altri account di utilizzare una chiave KMS](#).

19. Scegli Next (Successivo).
20. Consulta le principali dichiarazioni politiche relative alla chiave. Per apportare modifiche alla politica chiave, seleziona Modifica.
21. Scegli Next (Successivo).
22. Esaminare le principali impostazioni scelte. È comunque possibile tornare indietro e modificare tutte le impostazioni.
23. Scegli Finish (Termina) per creare la chiave KMS HMAC.

Utilizzo dell' AWS KMS API

È possibile utilizzare l'[CreateKey](#) operazione per creare una chiave KMS HMAC. Questi esempi utilizzano la [AWS Command Line Interface \(AWS CLI\)](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

Quando crei una chiave KMS HMAC, devi specificare il parametro KeySpec che determina il tipo di chiave KMS. Inoltre, devi specificare il valore KeyUsage di GENERATE_VERIFY_MAC, anche se è l'unico valore valido per l'utilizzo della chiave per le chiavi HMAC. Per creare una chiave KMS HMAC [multi-regione](#), aggiungi il parametro MultiRegion con un valore true. Una volta creata la chiave KMS, non è più possibile modificare queste proprietà.

L'CreateKeyoperazione non consente di specificare un alias, ma è possibile utilizzare l'[CreateAlias](#)operazione per creare un alias per la nuova chiave KMS. Ti consigliamo di utilizzare un alias che identifichi la chiave KMS come chiave HMAC, ad esempio HMAC/test-key. In questo modo sarà più facile identificare le chiavi HMAC nella AWS KMS console, dove è possibile ordinare e filtrare le chiavi per alias, ma non per specifica chiave o utilizzo della chiave.

Se si tenta di creare una chiave KMS HMAC in un ambiente Regione AWS in cui le chiavi HMAC non sono supportate, l'operazione restituisce un `CreateKey UnsupportedOperationException`

Gli esempi seguenti utilizzano l'operazione `CreateKey` per creare una chiave KMS HMAC a 512 bit.

```
$ aws kms create-key --key-spec HMAC_512 --key-usage GENERATE_VERIFY_MAC
{
  "KeyMetadata": {
    "KeyState": "Enabled",
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "Description": "",
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": 1669973196.214,
    "MultiRegion": false,
    "KeySpec": "HMAC_512",
    "CustomerMasterKeySpec": "HMAC_512",
    "KeyUsage": "GENERATE_VERIFY_MAC",
    "MacAlgorithms": [
      "HMAC_SHA_512"
    ],
    "AWSAccountId": "111122223333",
    "Origin": "AWS_KMS",
    "Enabled": true
  }
}
```

Crea chiavi primarie multiregionali

Puoi creare una [chiave primaria multiregionale](#) nella AWS KMS console o utilizzando l'API. AWS KMS È possibile creare la chiave primaria Regione AWS ovunque AWS KMS supporti le chiavi multiregionali.

Per creare una chiave primaria multiregionale, il principale necessita delle [stesse autorizzazioni](#) di cui ha bisogno per creare qualsiasi chiave KMS, inclusa l'CreateKeyautorizzazione [kms:](#) in una policy IAM. [Il preside necessita anche dell'autorizzazione iam:: CreateServiceLinkedRole](#) Puoi usare la chiave [kms: MultiRegionKeyType](#) condition per consentire o negare l'autorizzazione alla creazione di chiavi primarie multiregionali.

Note

Quando crei la tua chiave primaria multiregionale, considera attentamente gli utenti e i ruoli IAM che scegli per amministrare e utilizzare la chiave. Le policy IAM possono fornire ad altri ruoli e utenti IAM l'autorizzazione per gestire la chiave KMS.

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

Utilizzo della console AWS KMS

Per creare una chiave primaria multiregionale nella AWS KMS console, utilizza lo stesso processo che utilizzeresti per creare qualsiasi chiave KMS. Seleziona una chiave multiregione in Opzioni avanzate. Per istruzioni complete, consulta [Crea una chiave KMS](#).

Important

Non includere informazioni riservate o sensibili nell'alias, nella descrizione o nei tag. Questi campi possono apparire in testo semplice nei CloudTrail log e in altri output.

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
4. Scegli Crea chiave.
5. Selezionare un tipo di chiave [simmetrico o asimmetrico](#). Le chiavi simmetriche sono le chiavi di default.

È possibile creare chiavi simmetriche e asimmetriche multi-regione, incluse le chiavi KMS HMAC multi-regione, che sono simmetriche.

6. Seleziona l'utilizzo della chiave. Encrypt and decrypt (Crittografa e decrittografa) è l'utilizzo di default.

Per assistenza, consulta le sezioni [Crea una chiave KMS](#), [the section called “Creazione di una chiave KMS asimmetrica”](#) o [the section called “Creazione di una chiave KMS HMAC”](#).

7. Espandi Opzioni avanzate.
8. In Origine del materiale chiave, per AWS KMS generare il materiale chiave che condivideranno le chiavi principali e di replica, scegli KMS. Se [importi il materiale della chiave](#) nelle chiavi primarie e di replica, scegli External (Import key material) (Esterna (Importa materiale della chiave)).
9. In Regionalità, scegli la chiave multiregionale.

Non è possibile modificare questa impostazione dopo aver creato la chiave KMS.

10. Digita un [alias](#) per la chiave primaria.

Gli alias non sono una proprietà condivisa delle chiavi multiregione. Puoi assegnare alla tua chiave primaria multiregionale e alle sue repliche lo stesso alias o alias diversi. AWS KMS non sincronizza gli alias delle chiavi multiregionali.

Note

L'aggiunta, l'eliminazione o l'aggiornamento di un alias può consentire o negare l'autorizzazione alla chiave KMS. Per informazioni dettagliate, consulta [ABAC per AWS KMS](#) e [Usa gli alias per controllare l'accesso alle chiavi KMS](#).

11. (Facoltativo) Digita una descrizione della chiave primaria.

Le descrizioni non sono una proprietà condivisa delle chiavi multiregione. È possibile assegnare alla chiave primaria multiregione e alle relative repliche la stessa descrizione o descrizioni diverse. AWS KMS non sincronizza le descrizioni dei tasti delle chiavi multiregionali.

12. (Facoltativo) Digita tag per una chiave di un valore di tag facoltativo. Per assegnare più di un tag alla chiave primaria, scegli Aggiungi tag.

I tag non sono una proprietà condivisa delle chiavi multiregione. È possibile assegnare alla chiave primaria multiregione e alle relative repliche gli stessi tag o tag diversi. AWS KMS non

sincronizza i tag delle chiavi multiregione. Puoi modificare i tag nelle chiaviKMS in qualsiasi momento.

 Note

L'applicazione o l'eliminazione di un tag chiave KMS può consentire o negare l'autorizzazione alla chiave KMS. Per informazioni dettagliate, consulta [ABAC per AWS KMS](#) e [Usa i tag per controllare l'accesso alle chiavi KMS](#).

13. Seleziona i ruoli e gli utenti IAM che possono gestire la chiave primaria.

 Note

- Questo passaggio avvia il processo di creazione di una [policy chiave](#) per la chiave primaria. Le policy chiave non sono una proprietà condivisa delle chiavi multiregione. È possibile assegnare alla chiave primaria multiregionale e alle relative repliche la stessa politica chiave o politiche chiave diverse. AWS KMS non sincronizza le politiche chiave delle chiavi multiregionali. È possibile modificare la policy chiave di una chiave KMS in qualsiasi momento.
- Quando crei una chiave primaria multiregionale, prendi in considerazione l'utilizzo della [policy di chiave predefinita](#) generata dalla console. Se modifichi questa politica, la console non fornirà i passaggi per selezionare gli amministratori e gli utenti chiave durante la creazione delle chiavi di replica, né aggiungerà le dichiarazioni politiche corrispondenti. Di conseguenza, dovrai aggiungerle manualmente.
- La AWS KMS console aggiunge gli amministratori chiave alla policy chiave sotto l'identificatore "Allow access for Key Administrators" dell'istruzione. La modifica di questo identificatore di istruzione potrebbe influire sul modo in cui la console visualizza gli aggiornamenti apportati all'istruzione.

14. (Facoltativo) Per impedire ai ruoli e agli utenti IAM selezionati di eliminare questa chiave KMS, nella sezione Eliminazione chiave nella parte inferiore della pagina, deselecta la casella di controllo Consenti agli amministratori delle chiavi di eliminare questa chiave.

15. Scegli Next (Successivo).

16. Seleziona i ruoli e gli utenti IAM che possono utilizzare la chiave KMS per [operazioni di crittografia](#).

 Note

La AWS KMS console aggiunge gli utenti chiave alla politica chiave sotto gli "Allow use of the key" identificatori di dichiarazione e. "Allow attachment of persistent resources" La modifica di questi identificatori delle istruzioni potrebbe influire sul modo in cui la console visualizza gli aggiornamenti apportati all'istruzione.

17. (Facoltativo) È possibile consentire ad altri Account AWS di utilizzare questa chiave KMS per operazioni crittografiche. A questo proposito, nella sezione Altri Account AWS, nella parte inferiore della pagina, scegli Aggiungi un altro Account AWS e inserisci il numero di identificazione Account AWS di un account esterno. Per aggiungere più account esterni, ripetere questo passaggio.

 Note

Per consentire ai principali negli account esterni di utilizzare la chiave KMS, gli amministratori dell'account esterno devono creare policy IAM che forniscono tali autorizzazioni. Per ulteriori informazioni, consultare [Autorizzazione per gli utenti in altri account di utilizzare una chiave KMS](#).

18. Scegli Next (Successivo).
19. Consulta le principali dichiarazioni politiche relative alla chiave. Per apportare modifiche alla politica chiave, seleziona Modifica.
20. Scegli Next (Successivo).
21. Esaminare le principali impostazioni scelte. È comunque possibile tornare indietro e modificare tutte le impostazioni.
22. Scegli Fine per creare la chiave primaria multiregionale.

Utilizzo dell'API AWS KMS

Per creare una chiave primaria multiregionale, usa l'[CreateKey](#) operazione. Usa il parametro `MultiRegion` con valore `True`.

Ad esempio, il comando seguente crea una chiave primaria multiregionale in quella del chiamante (Regione AWS us-east-1). Accetta valori predefiniti per tutte le altre proprietà, inclusa la policy chiave. I valori predefiniti per le chiavi primarie multiregione sono gli stessi dei valori predefiniti per

tutte le altre chiavi KMS, inclusa la proprietà [policy chiave predefinita](#). Questa procedura crea una chiave di crittografia simmetrica, la chiave KMS di default.

La risposta include l'elemento `MultiRegion` e l'elemento `MultiRegionConfiguration` con sottoelementi e valori tipici per una chiave primaria multiregione senza chiavi di replica. L'[ID chiave](#) di una chiave multiregione inizia sempre con `mrk-`.

 Important

Non includere informazioni riservate o sensibili nei campi `Description` o `Tags`. Questi campi possono apparire in testo semplice nei CloudTrail log e in altri output.

```
$ aws kms create-key --multi-region
{
  "KeyMetadata": {
    "Origin": "AWS_KMS",
    "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
    "Description": "",
    "KeyManager": "CUSTOMER",
    "Enabled": true,
    "KeySpec": "SYMMETRIC_DEFAULT",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "KeyState": "Enabled",
    "CreationDate": 1606329032.475,
    "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
    "AWSAccountId": "111122223333",
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "MultiRegion": true,
    "MultiRegionConfiguration": {
      "MultiRegionKeyType": "PRIMARY",
      "PrimaryKey": {
        "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "Region": "us-east-1"
      },
      "ReplicaKeys": [ ]
    }
  }
}
```

```
}  
}
```

Crea chiavi di replica multiregionali

[È possibile creare una chiave di replica multiregionale nella AWS KMS console, utilizzando l'ReplicateKeyoperazione o utilizzando un modello. AWS::KMS::ReplicaKey CloudFormation](#) Non è possibile utilizzare l'[CreateKey](#)operazione per creare una chiave di replica.

È possibile utilizzare queste procedure per replicare qualsiasi chiave primaria multi-regione, incluse le [chiavi KMS di crittografia simmetrica](#), le [chiavi KMS asimmetriche](#) e le [chiavi KMS HMAC](#).

Al termine di questa operazione, la nuova chiave di replica presenta uno [stato chiave](#) `Creating`. Lo stato della chiave cambia in `Enabled` (o `PendingImport` se si crea una chiave multiregionale con [materiale chiave importato](#)) dopo alcuni secondi, quando il processo di creazione della nuova chiave di replica è completo. Quando lo stato della chiave è `Creating`, puoi visualizzare e gestire la chiave, ma non utilizzarla per operazioni di crittografia. Se state creando e utilizzando la chiave di replica a livello di codice, riprova `KMSInvalidStateException` o richiama [DescribeKey](#) per verificarne il valore prima di utilizzarla. `KeyState`

Se si elimina accidentalmente una chiave di replica, è possibile utilizzare questa procedura per ricrearla. Se si replica la stessa chiave primaria nella stessa regione, la nuova chiave di replica creata avrà le stesse [proprietà condivise](#) della chiave di replica originale.

Important

Non includere informazioni riservate o sensibili nell'alias, nella descrizione o nei tag. Questi campi possono apparire in testo semplice nei CloudTrail log e in altri output.

Per utilizzare un AWS CloudFormation modello per creare una chiave di replica, consulta [AWS::KMS::ReplicaKey](#) la Guida per l'AWS CloudFormation utente.

Passaggio 1: Scegli le regioni di replica

In genere si sceglie di replicare una chiave multiregionale in una in Regione AWS base al modello di business e ai requisiti normativi. Ad esempio, puoi replicare una chiave nelle Regioni in cui conservi le tue risorse. In alternativa, per soddisfare i requisiti di ripristino di emergenza, è possibile replicare una chiave in Regioni geograficamente distanti.

Di seguito sono riportati i AWS KMS requisiti per le regioni di replica. Se la Regione scelta non è conforme a questi requisiti, i tentativi di replicare una chiave hanno esito negativo.

- Una chiave multiregione correlata per Regione: non è possibile creare una chiave di replica nella stessa Regione della chiave primaria o nella stessa Regione di un'altra replica della chiave primaria.

Se si prova a replicare una chiave primaria in una regione che ha già una replica di quella chiave primaria, il tentativo avrà esito negativo. Se la chiave di replica corrente nella regione si trova nello [stato delle chiavi PendingDeletion](#), è possibile [annullare l'eliminazione della chiave di replica](#) oppure attendere fino a quando la chiave di replica non viene eliminata.

- Più chiavi multiregione non correlate nella stessa Regione — È possibile avere più chiavi multiregione non correlate nella stessa Regione. Ad esempio, è possibile avere due chiavi primarie multiregione nella Regione us-east-1. Ciascuna delle chiavi primarie può avere una chiave di replica nella Regione us-west-2.
- Regioni nella stessa partizione — La Regione della chiave di replica deve trovarsi nella stessa [partizione AWS](#) della Regione della chiave primaria.
- La Regione deve essere abilitata — Se una regione è [disabilitata per impostazione predefinita](#), non è possibile creare alcuna risorsa in tale Regione finché non viene abilitata per il tuo Account AWS.

Fase 2: Creare chiavi di replica

Note

Quando crei le chiavi di replica, considera attentamente gli utenti e i ruoli IAM che scegli per amministrare e utilizzare la chiave di replica. Le policy IAM possono fornire ad altri ruoli e utenti IAM l'autorizzazione per gestire la chiave KMS.

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

Utilizzo della console AWS KMS

Nella AWS KMS console, è possibile creare una o più repliche di una chiave primaria multiregionale con la stessa operazione.

Questa procedura è simile alla creazione di una chiave KMS standard per singola Regione nella console. Tuttavia, poiché una chiave di replica è basata sulla chiave primaria, non è possibile selezionare i valori per le [proprietà condivise](#), ad esempio la specifica della chiave (simmetrica o asimmetrica), l'utilizzo della chiave o l'origine della chiave.

È possibile specificare proprietà non condivise, tra cui un alias, tag, una descrizione e una policy chiave. Per comodità, la console visualizza i valori delle proprietà correnti della chiave primaria, ma è possibile modificarli. Anche se si mantengono i valori della chiave primaria, questi valori AWS KMS non vengono mantenuti sincronizzati.

 Important

Non includere informazioni riservate o sensibili nell'alias, nella descrizione o nei tag. Questi campi possono apparire in testo semplice nei CloudTrail log e in altri output.

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
4. Seleziona l'alias o l'ID chiave di una [chiave primaria multiregione](#). In questo modo si apre la pagina dei dettagli delle chiavi per la chiave KMS.

Per identificare una chiave primaria multiregione, utilizza l'icona dello strumento nell'angolo in alto a destra per aggiungere la colonna Regionalità nella tabella.

5. Seleziona la tab Regionalità.
6. Nella sezione Chiavi multiregione correlate, scegli Crea nuove chiavi di replica.

Le chiavi multiregione correlate mostrano la Regione della chiave primaria e le relative chiavi di replica. È possibile utilizzare questa visualizzazione per scegliere la Regione per la nuova chiave di replica.

7. Scegli una o più Regioni AWS. Questa procedura crea una chiave di replica in ciascuna delle Regioni selezionate.

Il menu include solo le regioni nella stessa AWS partizione della chiave primaria. Le Regioni che dispongono già di una chiave multiregione correlata vengono visualizzate, ma non sono

selezionabili. È possibile che non si disponga dell'autorizzazione per replicare una chiave in tutte le Regioni del menu.

Quando hai finito di scegliere Regioni, chiudi il menu. Vengono visualizzate le Regioni selezionate. Per annullare la replica in una Regione, scegli la casella di controllo X accanto al nome della Regione.

8. Digita un [alias](#) per la chiave di replica.

La console visualizza uno degli alias correnti della chiave primaria, ma è possibile modificarlo. È possibile assegnare alla chiave primaria multiregione e alle relative repliche gli stessi alias o alias diversi. Gli alias non sono una [proprietà condivisa delle chiavi](#) multiregionali. AWS KMS non sincronizza gli alias delle chiavi multiregionali.

L'aggiunta, l'eliminazione o l'aggiornamento di un alias può consentire o negare l'autorizzazione alla chiave KMS. Per informazioni dettagliate, consulta [ABAC per AWS KMS](#) e [Usa gli alias per controllare l'accesso alle chiavi KMS](#).

9. (Facoltativo) Digita una descrizione della chiave di replica.

La console visualizza la descrizione corrente della chiave primaria, ma è possibile modificarla. Le descrizioni non sono una proprietà condivisa delle chiavi multiregione. È possibile assegnare alla chiave primaria multiregione e alle relative repliche la stessa descrizione o descrizioni diverse. AWS KMS non sincronizza le descrizioni dei tasti delle chiavi multiregionali.

10. (Facoltativo) Digita tag per una chiave di un valore di tag facoltativo. Per assegnare più di un tag alla chiave di replica, scegli Aggiungi tag.

Nella console vengono visualizzati i tag attualmente collegati alla chiave primaria, ma è possibile modificarli. I tag non sono una proprietà condivisa delle chiavi multiRegione. È possibile assegnare alla chiave primaria multiregione e alle relative repliche gli stessi tag o tag diversi. AWS KMS non sincronizza i tag delle chiavi multiregionali.

L'applicazione o l'eliminazione di un tag chiave KMS può consentire o negare l'autorizzazione alla chiave KMS. Per informazioni dettagliate, consulta [ABAC per AWS KMS](#) e [Usa i tag per controllare l'accesso alle chiavi KMS](#).

11. Seleziona i ruoli e gli utenti IAM che possono gestire la chiave di replica.

 Note

- Se hai modificato la politica delle chiavi predefinita durante la creazione della chiave primaria multiregionale, la console non ti chiederà di selezionare gli amministratori o gli utenti chiave (passaggi 11-15) durante la creazione della chiave di replica. In questo caso, dovrai aggiungere manualmente le autorizzazioni necessarie per gli amministratori e gli utenti principali alla politica chiave selezionando Modifica nel passaggio Modifica la politica chiave (Passaggio 17).
- Questo passaggio avvia il processo di creazione di una [policy chiave](#) per la chiave di replica. Nella console vengono visualizzate le policy chiave correnti della chiave primaria, ma è possibile modificarle. Le policy chiave non sono una proprietà condivisa delle chiavi multiregione. È possibile assegnare alla chiave primaria multiregione e alle relative repliche le stesse policy chiave o policy chiave diverse. AWS KMS non sincronizza le policy chiave. È possibile modificare la policy chiave delle chiavi KMS in qualsiasi momento.
- La AWS KMS console aggiunge gli amministratori chiave alla politica chiave sotto l'identificatore dell'istruzione. "Allow access for Key Administrators"
La modifica di questo identificatore di istruzione potrebbe influire sul modo in cui la console visualizza gli aggiornamenti apportati all'istruzione.

12. (Facoltativo) Per impedire ai ruoli e agli utenti IAM selezionati di eliminare questa chiave KMS, nella sezione Eliminazione chiave nella parte inferiore della pagina, deseleziona la casella di controllo Consenti agli amministratori delle chiavi di eliminare questa chiave.
13. Scegli Next (Successivo).
14. Seleziona i ruoli e gli utenti IAM che possono utilizzare la chiave KMS per [operazioni di crittografia](#).

 Nota

La AWS KMS console aggiunge gli utenti chiave alla politica chiave sotto gli "Allow use of the key" identificatori di dichiarazione e "Allow attachment of persistent resources" La modifica di questi identificatori delle istruzioni potrebbe influire sul modo in cui la console visualizza gli aggiornamenti apportati all'istruzione.

15. (Facoltativo) È possibile consentire ad altri Account AWS di utilizzare questa chiave KMS per operazioni crittografiche. A questo proposito, nella sezione Altri Account AWS, nella parte inferiore della pagina, scegli Aggiungi un altro Account AWS e inserisci il numero di identificazione Account AWS di un account esterno. Per aggiungere più account esterni, ripetere questo passaggio.

Note

Per consentire ai principali negli account esterni di utilizzare la chiave KMS, gli amministratori dell'account esterno devono creare policy IAM che forniscono tali autorizzazioni. Per ulteriori informazioni, consultare [Autorizzazione per gli utenti in altri account di utilizzare una chiave KMS](#).

16. Scegli Next (Successivo).
17. Consulta le principali dichiarazioni politiche relative alla chiave. Per apportare modifiche alla politica chiave, seleziona Modifica.
18. Scegli Next (Successivo).
19. Esaminare le principali impostazioni scelte. È comunque possibile tornare indietro e modificare tutte le impostazioni.
20. Scegli Fine per creare la chiave di replica multiregionale.

Utilizzo dell'API AWS KMS

Per creare una chiave di replica multiregionale, utilizzare l'[ReplicateKey](#) operazione. Non è possibile utilizzare l'[CreateKey](#) operazione per creare una chiave di replica. Questa operazione crea una chiave di replica per volta. La regione specificata deve essere conforme ai [Requisiti per le Regioni](#) per le chiavi di replica.

Quando si utilizza l'operazione `ReplicateKey`, non specificare valori per le [proprietà condivise](#) delle chiavi multiregione. I valori delle proprietà condivise vengono copiati dalla chiave primaria e mantenuti sincronizzati. Tuttavia, è possibile specificare valori per proprietà non condivise. Altrimenti, AWS KMS applica i valori predefiniti standard per le chiavi KMS, non i valori della chiave primaria.

 Note

Se non specifichi valori per i Tags parametri `Description`, `oKeyPolicy`, AWS KMS crea la chiave di replica con una descrizione di stringa vuota, la [politica di chiave predefinita](#) e nessun tag.

Non includere informazioni riservate o sensibili nei campi `Description` o `Tags`. Questi campi possono apparire in testo semplice nei CloudTrail log e in altri output.

Ad esempio, il seguente comando crea una chiave di replica multiregione nella Regione Asia Pacifico (Sydney) (ap-southeast-2). Questa chiave di replica è modellata sulla chiave primaria nella Regione Stati Uniti orientali (Virginia settentrionale) (us-east-1), identificata dal valore del parametro `KeyId`. Questo esempio accetta valori predefiniti per tutte le altre proprietà, inclusa la policy chiave.

La risposta descrive la nuova chiave di replica. Include i campi per le proprietà condivise, ad esempio `KeyId`, `KeySpec`, `KeyUsage` e l'origine del materiale chiave (`Origin`). Include anche proprietà indipendenti dalla chiave primaria, come la `Description`, la policy chiave (`ReplicaKeyPolicy`), e i tag (`ReplicaTags`).

La risposta include anche l'ARN chiave e la Regione della chiave primaria e tutte le relative chiavi di replica, inclusa quella appena creata nella Regione ap-southeast-2. In questo esempio, l'elemento `ReplicaKey` mostra che questa chiave primaria è già stata replicata nella Regione Europa (Irlanda) (eu-west-1).

```
$ aws kms replicate-key \
  --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \
  --replica-region ap-southeast-2
{
  "ReplicaKeyMetadata": {
    "MultiRegion": true,
    "MultiRegionConfiguration": {
      "MultiRegionKeyType": "REPLICA",
      "PrimaryKey": {
        "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "Region": "us-east-1"
      },
      "ReplicaKeys": [
        {
```

```

        "Arn": "arn:aws:kms:ap-southeast-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "Region": "ap-southeast-2"
    },
    {
        "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "Region": "eu-west-1"
    }
]
},
"AWSAccountId": "111122223333",
"Arn": "arn:aws:kms:ap-southeast-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
"CreationDate": 1607472987.918,
"Description": "",
"Enabled": true,
"KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
"KeyManager": "CUSTOMER",
"KeySpec": "SYMMETRIC_DEFAULT",
"KeyState": "Enabled",
"KeyUsage": "ENCRYPT_DECRYPT",
"Origin": "AWS_KMS",
"CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
"EncryptionAlgorithms": [
    "SYMMETRIC_DEFAULT"
]
},
"ReplicaKeyPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Id\" : \"key-
default-1\",...,
  \"ReplicaTags\": []
}

```

Crea una chiave KMS con materiale chiave importato

Il materiale chiave importato consente di proteggere le AWS risorse con le chiavi crittografiche generate dall'utente. La seguente panoramica illustra come importare il materiale della chiave in AWS KMS. Per ulteriori dettagli su ogni fase del processo, consultate gli argomenti corrispondenti.

1. [Crea una chiave KMS senza materiale della chiave](#): l'origine deve essere EXTERNAL. Un'origine chiave di EXTERNAL indica che la chiave è progettata per il materiale chiave importato e AWS

KMS impedisce la generazione di materiale chiave per la chiave KMS. In una fase successiva importerai il tuo materiale della chiave in questa chiave KMS.

Il materiale chiave che importate deve essere compatibile con le specifiche chiave della chiave associata AWS KMS. Per ulteriori informazioni sulla compatibilità, vedere [the section called "Requisiti per il materiale della chiave importato"](#).

2. [Scarica la chiave pubblica di wrapping e il token di importazione](#): dopo aver completato la fase 1, scarica una chiave pubblica di wrapping e un token di importazione. Questi articoli proteggono il materiale chiave durante l'importazione AWS KMS.

In questa fase, scegli il tipo ("specifica chiave") della chiave di wrapping RSA e l'algoritmo di wrapping che utilizzerai per la crittografia dei dati in transito in AWS KMS. Puoi scegliere una specifica della chiave di wrapping e un algoritmo della chiave di wrapping diversi ogni volta che importi o reimporti lo stesso materiale della chiave.

3. [Decifra il materiale della chiave](#): usa la chiave pubblica di wrapping che hai scaricato nella fase 2 per crittografare il materiale della chiave che hai creato sul tuo sistema.
4. [Importa il materiale chiave](#) – Carica il materiale della chiave crittografato che hai creato nella fase 3 e il token di importazione che hai scaricato nella fase 2.

In questa fase, puoi [impostare una scadenza facoltativa](#). Quando il materiale chiave importato scade, lo AWS KMS elimina e la chiave KMS diventa inutilizzabile. Per continuare a utilizzare la chiave KMS, devi importare nuovamente lo stesso materiale della chiave.

Quando l'operazione di importazione viene completata correttamente, lo stato della chiave della chiave KMS cambia da PendingImport a Enabled. ora, puoi utilizzare la chiave KMS nelle operazioni di crittografia.

AWS KMS [registra una voce nel AWS CloudTrail registro quando si crea la chiave KMS, si scarica la chiave pubblica di wrapping e si importa il token e si importa il materiale chiave](#). AWS KMS registra anche una voce quando si elimina materiale chiave importato o quando si AWS KMS [elimina materiale chiave scaduto](#).

Autorizzazioni per l'importazione del materiale della chiave

Per creare e gestire le chiave KMS con materiale della chiave importato, l'utente deve avere l'autorizzazione per le operazioni di questo processo. Puoi fornire le autorizzazioni `kms:GetParametersForImport`, `kms:ImportKeyMaterial`, e

`kms:DeleteImportedKeyMaterial` nella policy delle chiavi quando crei la chiave KMS. Nella AWS KMS console, queste autorizzazioni vengono aggiunte automaticamente per gli amministratori chiave quando si crea una chiave con un'origine materiale esterna.

Per creare chiavi KMS con materiale della chiave importato, il principale richiede le seguenti autorizzazioni.

- [kms: CreateKey \(politica IAM\)](#)
 - Per limitare questa autorizzazione alle chiavi KMS con materiale chiave importato, utilizza la condizione [kms: KeyOrigin](#) policy con un valore di. EXTERNAL

```
{
  "Sid": "CreateKMSKeysWithoutKeyMaterial",
  "Effect": "Allow",
  "Resource": "*",
  "Action": "kms:CreateKey",
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "EXTERNAL"
    }
  }
}
```

- [kms: GetParametersForImport](#) (Politica chiave o politica IAM)
 - [Per limitare questa autorizzazione alle richieste che utilizzano un particolare algoritmo di wrapping e una specifica chiave di wrapping, utilizza le condizioni delle policy kms: WrappingAlgorithm e kms: WrappingKeySpec](#)
- [kms: ImportKeyMaterial](#) (Politica chiave o politica IAM)
 - Per consentire o vietare la scadenza del materiale chiave e controllare la data di scadenza, utilizza le condizioni delle politiche [kms: ExpirationModel](#) e [kms: ValidTo](#)

[Per reimportare il materiale chiave importato, il principale necessita delle autorizzazioni kms: e kms: GetParametersForImport ImportKeyMaterial](#)

[Per eliminare il materiale chiave importato, il principale necessita dell'autorizzazione kms: DeleteImportedKeyMaterial](#)

Ad esempio, per dare a `KMSAdminRole` di esempio l'autorizzazione per gestire tutti gli aspetti di una chiave KMS con materiale chiave importato, includi una dichiarazione di policy delle chiavi come la seguente nella policy della chiave KMS.

```
{
  "Sid": "Manage KMS keys with imported key material",
  "Effect": "Allow",
  "Resource": "*",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/KMSAdminRole"
  },
  "Action": [
    "kms:GetParametersForImport",
    "kms:ImportKeyMaterial",
    "kms:DeleteImportedKeyMaterial"
  ]
}
```

Requisiti per il materiale della chiave importato

Il materiale della chiave importato deve essere compatibile con le [specifiche chiave](#) della chiave KMS associata. Per le coppie di chiavi asimmetriche, importa solo la chiave privata della coppia. AWS KMS ricava la chiave pubblica dalla chiave privata.

AWS KMS supporta le seguenti specifiche chiave per le chiavi KMS con materiale chiave importato.

- Chiavi di crittografia simmetrica
 - Specifiche chiave:
 - SYMMETRIC_DEFAULT.
 - Requisiti:
 - 256 bit (32 byte) di dati binari.
 - Nelle regioni cinesi, devono essere dati binari a 128 bit (16 byte).
- Chiavi HMAC
 - Specifiche chiave:
 - HMAC_224
 - HMAC_256
 - HMAC_384

- HMAC_512
- Requisiti:
 - Il materiale della chiave HMAC deve essere conforme alla [RFC 2104](#).
 - La lunghezza della chiave deve essere almeno la stessa lunghezza specificata dalle specifiche della chiave. La lunghezza massima della chiave è di 1024 bit.
 - Se il materiale della chiave supera i 1024 bit, è possibile eseguire l'hash del materiale chiave e importare l'output dell'hash. L'algoritmo di hashing deve corrispondere alle specifiche chiave della chiave HMAC KMS che stai creando.
- Esempio:
 - Per importare 2048 bit di materiale chiave in una chiave HMAC_256, calcola prima l'hash SHA-256 del materiale chiave a 2048 bit, quindi importa l'output hash a 256 bit risultante nella chiave KMS.
- Lunghezze di chiave valide:
 - HMAC_224:224—1024 bit
 - HMAC_256:256-1024 bit
 - HMAC_384:384—1024 bit
 - HMAC_512:512—1024 bit
- Chiave privata asimmetrica RSA
 - Specifiche chiave:
 - RSA_2048
 - RSA_3072
 - RSA_4096
 - Requisiti:
 - La chiave privata asimmetrica RSA importata deve far parte di una coppia di chiavi conforme alla [RFC 3447](#).
 - Modulo: 2048 bit, 3072 bit o 4096 bit
 - Numero di numeri primi: 2 (le chiavi RSA con più numeri primi non sono supportate)
 - [Il materiale a chiave asimmetrica deve essere codificato in BER o DER nel formato Public-Key Cryptography Standards \(PKCS\) #8 conforme alla RFC 5208.](#)
- Chiave privata asimmetrica a curva ellittica
 - Specifiche chiave:

- ECC_NIST_P256 (secp256r1)
- ECC_NIST_P384 (secp384r1)
- ECC_NIST_P521 (secp521r1)
- ECC_SECG_P256K1 (secp256k1)
- EDWARDS25519 ECC_NIST_ (ed25519)
- Requisiti:
 - La chiave privata asimmetrica RSA importata deve far parte di una coppia di chiavi conforme alla [RFC 5915](#).
 - Curva: NIST P-256, NIST P-384, NIST P-521, SecP256k1, NIST Ed25519.
 - Parametri: solo curve con nome (le chiavi ECC con parametri espliciti vengono rifiutate).
 - Coordinate pubbliche dei punti: possono essere compresse, non compresse o proiettive.
 - [Il materiale a chiave asimmetrica deve essere codificato in BER o DER nel formato Public-Key Cryptography Standards \(PKCS\) #8 conforme alla RFC 5208.](#)
- Chiave ML-DSA
 - Specifiche chiave:
 - ML_DSA_44
 - ML_DSA_65
 - ML_DSA_87

 Important

L'importazione di chiavi ML-DSA non è supportata.

- SM2 chiave privata asimmetrica (solo regioni della Cina)
- Requisiti:
 - La chiave privata SM2 asimmetrica importata deve far parte di una coppia di chiavi conforme a 0003. GM/T
 - SM2Curva:.
 - Parametri: solo curva con nome (SM2 le chiavi con parametri espliciti vengono rifiutate).
 - Coordinate pubbliche dei punti: possono essere compresse, non compresse o proiettive.
 - [Il materiale a chiave asimmetrica deve essere codificato in BER o DER nel formato Public-Key Cryptography Standards \(PKCS\) #8 conforme alla RFC 5208.](#)

Fase AWS KMS key 1: Creare un materiale senza chiave

Per impostazione predefinita, AWS KMS crea automaticamente il materiale chiave quando crei una chiave KMS. Per importare invece il materiale della propria chiave, avvia creando una chiave KMS senza materiale chiave. Quindi, importare il materiale chiave. Per creare una chiave KMS senza materiale chiave, usa la AWS KMS console o l'[CreateKey](#) operazione.

Per creare una chiave senza materiale della chiave, specifica un'[origine](#) EXTERNAL. La proprietà dell'origine di una chiave KMS è immutabile. Una volta creata, non è possibile convertire una chiave KMS progettata per il materiale chiave importato in una chiave KMS con materiale chiave proveniente da AWS KMS o da qualsiasi altra fonte.

Lo [stato della chiave](#) di una chiave KMS con un origine EXTERNAL e nessun materiale chiave è PendingImport. Una chiave KMS può rimanere in uno stato PendingImport indefinitamente. Tuttavia, non puoi utilizzare una chiave KMS in stato PendingImport in operazioni crittografiche. Quando importi il materiale della chiave, lo stato della chiave KMS diventa Enabled e puoi utilizzarla in operazioni crittografiche.

AWS KMS registra un evento nel AWS CloudTrail registro quando si [crea la chiave KMS, si scarica la chiave pubblica e si importa il token e si importa il](#) materiale chiave. AWS KMS registra anche un CloudTrail evento quando si [elimina materiale chiave importato o quando si AWS KMS elimina materiale chiave scaduto](#).

Argomenti

- [Creazione di una chiave KMS senza materiale della chiave \(console\)](#)
- [Creazione di una chiave KMS senza materiale chiave \(API\)](#) **AWS KMS**

Creazione di una chiave KMS senza materiale della chiave (console)

Devi solo creare una sola volta una chiave KMS per il materiale della chiave importato. Puoi importare e reimportare quando vuoi lo stesso materiale della chiave nella chiave KMS, ma non puoi importare materiale della chiave diverso in una chiave KMS. Per informazioni dettagliate, vedi [Fase 2: download della chiave pubblica di wrapping e del token di importazione](#).

Per trovare le chiavi KMS esistenti con materiale della chiave importato nella tabella Customer managed keys (Chiavi gestite dal cliente), utilizza l'icona a forma di ingranaggio nell'angolo in alto a destra per mostrare la colonna Origin (Origine) nell'elenco delle chiavi KMS. Il valore di Origine per le chiavi importate è Esterna (Importa il materiale della chiave).

Per creare una chiave KMS con materiale chiave importato, inizia seguendo le [istruzioni per creare una chiave KMS del tipo di chiave preferito](#), con la seguente eccezione.

Una volta scelto l'utilizzo della chiave, effettua le seguenti operazioni:

1. Espandi Opzioni avanzate.
2. In Key material origin (Origine del materiale della chiave), seleziona External (Import key material) (Esterna (Importa materiale della chiave)).
3. Scegli la casella di controllo accanto a Comprendo le implicazioni in termini di sicurezza e durabilità derivanti dall'utilizzo di una chiave importata) per confermare di aver compreso le implicazioni dell'utilizzo del materiale della chiave importato. Per leggere queste implicazioni, consulta [Protezione del materiale della chiave importato](#).
4. Facoltativo: per creare una chiave [KMS multiregionale con materiale chiave](#) importato, in Regionalità seleziona Chiave multiregionale.
5. Torna alle istruzioni basilari. Le fasi rimanenti della procedura basilare sono identici per tutte le chiavi KMS di tale tipo.

Quando scegli Fine, hai creato una chiave KMS senza materiale della chiave con lo stato ([stato chiave](#)) Importazione in attesa.

Tuttavia, invece di tornare alla tabella delle Chiavi gestite dal cliente, la console visualizza una pagina in cui puoi scaricare la chiave pubblica e importare il token necessario per l'importazione del materiale della chiave. A questo punto, puoi continuare con la fase di download o scegliere Annulla per fermarti a questo punto. Puoi tornare a questa fase di download in qualunque momento.

Successivo: [Fase 2: download della chiave pubblica di wrapping e del token di importazione](#).

Creazione di una chiave KMS senza materiale chiave (API)AWS KMS

Per utilizzare l'[AWS KMS API](#) per creare una chiave KMS di crittografia simmetrica senza materiale chiave, invia una [CreateKey](#) richiesta con il `Origin` parametro impostato su `EXTERNAL`. L'esempio seguente mostra come eseguire questa operazione con l'[AWS Command Line Interface \(AWS CLI\)](#).

```
$ aws kms create-key --origin EXTERNAL
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente. La AWS KMS chiave `Origin` è `EXTERNAL` e la sua è `KeyState PendingImport`

 Tip

Se l'esito del comando non è positivo, potresti visualizzare un'`KMSInvalidStateException` o un'`NotFoundException`. Puoi ritentare la richiesta.

```
{
  "KeyMetadata": {
    "Origin": "EXTERNAL",
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "Description": "",
    "Enabled": false,
    "MultiRegion": false,
    "KeyUsage": "ENCRYPT_DECRYPT",
    "KeyState": "PendingImport",
    "CreationDate": 1568289600.0,
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ]
  }
}
```

Copia il valore `KeyId` dall'output del comando per utilizzarlo in un secondo momento, quindi passa a [Fase 2: download della chiave pubblica di wrapping e del token di importazione](#).

 Note

Questo comando crea una chiave KMS di crittografia simmetrica con `KeySpec SYMMETRIC_DEFAULT` e `KeyUsage ENCRYPT_DECRYPT`. Puoi utilizzare i parametri opzionali `--key-spec` e `--key-usage` per creare una chiave KMS HMAC o asimmetrica. Per maggiori informazioni, vedi l'operazione [CreateKey](#).

Fase 2: download della chiave pubblica di wrapping e del token di importazione

Dopo aver [creato un materiale AWS KMS key senza chiave](#), scarica una chiave pubblica di wrapping e un token di importazione per quella chiave KMS utilizzando la AWS KMS console o l'API. [GetParametersForImport](#) La chiave pubblica di wrapping e il token di importazione costituiscono due elementi di un set indivisibile che devono essere usati assieme.

Utilizzerai la chiave pubblica di wrapping per [crittografare il materiale della chiave](#) per il trasporto. [Prima di scaricare una coppia di chiavi di wrapping RSA, seleziona la lunghezza \(specifica chiave\) della coppia di chiavi di wrapping RSA e l'algoritmo di wrapping che utilizzerai per crittografare il materiale chiave importato per il trasporto nella fase 3.](#) AWS KMS supporta anche le specifiche della chiave di SM2 avvolgimento (solo regioni della Cina).

Ogni set di chiave pubblica di wrapping e token di importazione è valido per 24 ore. Se non lo utilizzi per importare il materiale della chiave entro 24 ore dal download, devi scaricare un nuovo set. Puoi scaricare set con una nuova chiave pubblica di wrapping e token di importazione in qualunque momento. Ciò consente di modificare la lunghezza della chiave di wrapping RSA ("specifica chiave") o di sostituire un set perso.

Puoi importare un set con chiave pubblica di wrapping e token di importazione anche per [importare nuovamente lo stesso materiale della chiave](#) in una chiave KMS. Puoi eseguire questa operazione per importare o modificare la data di scadenza del materiale della chiave o per ripristinare materiale della chiave scaduto o eliminato. È necessario scaricare e crittografare nuovamente il materiale chiave ogni volta che lo si importa in AWS KMS.

Utilizzo della chiave pubblica di wrapping

Il download include una chiave pubblica unica per te Account AWS, chiamata anche chiave pubblica di wrapping.

Prima di importare il materiale chiave, crittografate il materiale chiave con la chiave di wrapping pubblica, quindi caricate il materiale della chiave crittografata su AWS KMS. Quando AWS KMS riceve il materiale della chiave crittografata, lo decripta con la chiave privata corrispondente, quindi cripta nuovamente il materiale chiave con una chiave simmetrica AES, il tutto all'interno di un modulo di sicurezza hardware (HSM). AWS KMS

Utilizzo dei token di importazione

Il download include un token di importazione con i metadati che assicura che il materiale della chiave sia stato importato correttamente. Quando carichi il materiale della tua chiave crittografata su AWS KMS, devi caricare lo stesso token di importazione scaricato in questa fase.

Selezione di una specifica della chiave pubblica di wrapping

Per proteggere il materiale chiave durante l'importazione, lo crittografate utilizzando la chiave pubblica di wrapping da AWS KMS cui scaricate e un algoritmo di [wrapping](#) supportato. Seleziona una specifica chiave prima di scaricare la chiave pubblica di wrapping e il token di importazione. Tutte le coppie di chiavi di wrapping vengono generate in moduli di sicurezza AWS KMS hardware (HSMs). La chiave non esce mai dal modulo HSM in testo semplice.

Specifiche principali del wrapping RSA

Le specifiche chiave della chiave pubblica di wrapping determinano la lunghezza delle chiavi nella coppia di chiavi RSA che protegge il materiale della chiave durante il trasporto in AWS KMS. In generale, consigliamo di utilizzare la chiave pubblica di wrapping più lunga possibile tale che sia pratica. Offriamo diverse specifiche di wrapping public key per supportare una varietà di HSMs gestori chiave.

AWS KMS supporta le seguenti specifiche chiave per le chiavi di wrapping RSA utilizzate per importare materiale chiave di tutti i tipi, ad eccezione di quanto indicato.

- RSA_4096 (preferito)
- RSA_3072
- RSA_2048

Note

La seguente combinazione NON è supportata: materiale della chiave ECC_NIST_P521, specifica della chiave di wrapping pubblica RSA_2048 e algoritmo di wrapping RSAES_OAEP_SHA_*.

Non puoi eseguire il wrapping del materiale della chiave ECC_NIST_P521 direttamente con una chiave di wrapping pubblica RSA_2048. Usa una chiave di wrapping più grande o un algoritmo di wrapping RSA_AES_KEY_WRAP_SHA_*.

SM2 specifiche della chiave di avvolgimento (solo regioni della Cina)

AWS KMS supporta le seguenti specifiche chiave per le chiavi di SM2 avvolgimento utilizzate per importare materiale chiave asimmetrico.

- SM2

Selezione di un algoritmo di wrapping

Per proteggere il materiale della chiave durante l'importazione, crittografalo utilizzando la chiave pubblica di wrapping scaricata e un algoritmo di wrapping supportato.

AWS KMS supporta diversi algoritmi di wrapping RSA standard e un algoritmo di wrapping ibrido in due fasi. In generale, consigliamo di utilizzare l'algoritmo di wrapping più sicuro che sia compatibile con il materiale della chiave importato e con le [specifiche della chiave di wrapping](#). Di solito, puoi scegliere un algoritmo che è supportato dal modulo di sicurezza hardware (HSM) o del sistema di gestione delle chiavi che protegge il materiale della chiave.

La tabella seguente mostra gli algoritmi di wrapping supportati per ogni tipo di chiave KMS e materiale della chiave. Gli algoritmi sono elencati in ordine di preferenza.

Materiale chiave	Specifiche e algoritmo di wrapping supportati
Chiave di crittografia simmetrica	Algoritmi di wrapping:
Chiave AES a 256-bit	RSAES_OAEP_SHA_256
SM4 Chiave a 128 bit (solo regioni della Cina)	RSAES_OAEP_SHA_1
	Algoritmi di wrapping obsoleti:
	PKCS1RSAES_ _V1
	 Note Al 10 ottobre 2023, non AWS KMS supporta l'algoritmo di wrapping RSAES_ _V1_5. PKCS1

Materiale chiave	Specifiche e algoritmo di wrapping supportati
	Specifiche della chiave di wrapping: RSA_2048 RSA_3072 RSA_4096
Chiave privata RSA asimmetrica	Algoritmi di wrapping: RSA_AES_KEY_WRAP_SHA_256 RSA_AES_KEY_WRAP_SHA_1 SM2PKE (solo regioni della Cina) Specifiche della chiave di wrapping: RSA_2048 RSA_3072 RSA_4096 SM2 (Solo regioni della Cina)

Materiale chiave	Specifiche e algoritmo di wrapping supportati
Chiave privata a curva ellittica asimmetrica (ECC) Non puoi utilizzare gli algoritmi di wrapping RSAES_OAEP_SHA_* con la specifica della chiave di wrapping RSA_2048 per eseguire il wrapping del materiale della chiave ECC_NIST_P521.	Algoritmi di wrapping: RSA_AES_KEY_WRAP_SHA_256 RSA_AES_KEY_WRAP_SHA_1 RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 SM2PKE (solo regioni della Cina) Specifiche della chiave di wrapping: RSA_2048 RSA_3072 RSA_4096 SM2 (Solo regioni della Cina)
Chiave SM2 privata asimmetrica (solo regioni della Cina)	Algoritmi di wrapping: RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 SM2PKE (solo regioni della Cina) Specifiche della chiave di wrapping: RSA_2048 RSA_3072 RSA_4096 SM2 (Solo regioni della Cina)

Materiale chiave	Specifiche e algoritmo di wrapping supportati
Chiave HMAC	Algoritmi di wrapping: RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 Specifiche della chiave di wrapping: RSA_2048 RSA_3072 RSA_4096

 Note

Gli algoritmi RSA_AES_KEY_WRAP_SHA_256 and RSA_AES_KEY_WRAP_SHA_1 wrapping non sono supportati nelle regioni della Cina.

- RSA_AES_KEY_WRAP_SHA_256: algoritmo di wrapping ibrido in due fasi che combina la crittografia del materiale della chiave con una chiave simmetrica AES generata da te, quindi la crittografia della chiave simmetrica AES con la chiave di wrapping pubblica RSA scaricata e l'algoritmo di wrapping RSAES_OAEP_SHA_256.

È necessario un algoritmo di RSA_AES_KEY_WRAP_SHA_* wrapping per il wrapping del materiale a chiave privata RSA, tranne nelle regioni della Cina, dove è necessario utilizzare l'SM2PKE algoritmo di wrapping.

- RSA_AES_KEY_WRAP_SHA_1: algoritmo di wrapping ibrido in due fasi che combina la crittografia del materiale della chiave con una chiave simmetrica AES generata da te, quindi la crittografia della chiave simmetrica AES con la chiave pubblica di wrapping RSA scaricata e l'algoritmo di wrapping RSAES_OAEP_SHA_1.

È necessario un algoritmo di RSA_AES_KEY_WRAP_SHA_* wrapping per il wrapping del materiale a chiave privata RSA, tranne nelle regioni della Cina, dove è necessario utilizzare l'SM2PKE algoritmo di wrapping.

- **RSAES_OAEP_SHA_256**: l'algoritmo di crittografia RSA con Optimal Asymmetric Encryption Padding (OAEP) con la funzione hash SHA-256.
- **RSAES_OAEP_SHA_1**: l'algoritmo di crittografia RSA con Optimal Asymmetric Encryption Padding (OAEP) con la funzione hash SHA-1.
- **RSAES_PKCS1_V1_5**(Obsoleto; al 10 ottobre 2023, AWS KMS non supporta l'algoritmo di wrapping **RSAES_PKCS1_V1_5**) — L'algoritmo di crittografia RSA con il formato di riempimento definito in PKCS #1 versione 1.5.
- **SM2PKE**(Solo regioni della Cina) — Un algoritmo di crittografia basato sulla curva ellittica definito dall'OSCCA nel 0003.4-2012. GM/T

Argomenti

- [Download della chiave pubblica di wrapping e del token di importazione \(console\)](#)
- [Scaricamento della chiave pubblica di wrapping e del token di importazione \(API\)](#)**AWS KMS**

Download della chiave pubblica di wrapping e del token di importazione (console)

È possibile utilizzare la AWS KMS console per scaricare la chiave pubblica di wrapping e importare il token.

1. Se hai appena completato la procedura per [creare una chiave KMS senza materiale chiave](#) e ti trovi sulla pagina Scarica la chiave di wrapping e il token di importazione, passa a [Step 9](#).
2. Accedi Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) su <https://console.aws.amazon.com/kms>.
3. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
4. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.

Tip

Puoi importare il materiale della chiave solo in una chiave KMS con Origine Esterna (Importa il materiale della chiave). Ciò indica che la chiave KMS è stata creata senza materiale chiave. Per aggiungere una colonna Origin (Origine) alla tabella, nell'angolo in alto a destra della pagina scegliere l'icona delle impostazioni



Attivare l'origine in Origine (Origine), quindi scegliere Confirm (Conferma).

5. Scegli l'alias o l'ID chiave della chiave KMS che è in attesa di importazione.
6. Scegli la tab Configurazione crittografica e visualizzane i valori. Le tab si trovano nella sezione Configurazione generale.

Puoi importare il materiale della chiave solo in chiavi KMS con Origine Esterna (Importa il materiale della chiave). Per ulteriori informazioni sulla creazione di chiavi KMS con materiale chiave importato, consulta [Importazione di materiale chiave per le AWS KMS chiavi](#).

7. Scegli la scheda Materiale della chiave, quindi scegli Importa il materiale della chiave.

La scheda Materiale della chiave viene visualizzata solo per le chiavi KMS simmetriche il cui valore Origine è Esterna (Importa il materiale della chiave).

8. Per Seleziona le specifiche della chiave di wrapping, scegli la configurazione per la tua chiave KMS. Dopo aver creato questa chiave, non puoi modificare le specifiche della chiave.
9. In Select wrapping algorithm (Seleziona algoritmo di wrapping), scegliere l'opzione da utilizzare per crittografare il materiale della chiave. Per ulteriori informazioni sulle opzioni, consulta [Selezione di un algoritmo di wrapping](#).
10. Scegli Scarica la chiave pubblica di wrapping e il token di importazione), quindi salva il file.

Se è presente un'opzione Next (Successivo), per continuare il processo ora scegliere Next (Successivo). Per continuare in un secondo momento, scegliere Cancel (Annulla).

11. Decomprimere il file .zip salvato nella fase precedente (Import_Parameters_<key_id>_<timestamp>).

La cartella contiene i file seguenti:

- Una chiave pubblica che racchiude un file denominato WrappingPublicKey.bin
- Un token di importazione in un file denominato ImportToken.bin.
- Un file di testo denominato README.txt. Questo file contiene informazioni sulla chiave pubblica di wrapping, l'algoritmo di wrapping da utilizzare per crittografare il materiale della chiave e la data e l'ora in cui di scadenza della chiave pubblica di wrapping e del token di importazione.

12. Per continuare il processo, [crittografare il materiale della chiave](#).

Scaricamento della chiave pubblica di wrapping e del token di importazione (API)AWS KMS

Per scaricare la chiave pubblica e il token di importazione, utilizza l'[GetParametersForImport](#) API. Specifica la chiave KMS che verrà associata al materiale della chiave importato. Questa chiave KMS deve avere un valore di [origine](#) pari a EXTERNAL.

Note

Non è possibile importare materiale chiave per le chiavi KMS ML-DSA.

Questo esempio specifica l'algoritmo di wrapping RSA_AES_KEY_WRAP_SHA_256, la specifica della chiave pubblica di wrapping RSA_3072 e un esempio di ID chiave. Sostituisci questi valori di esempio con valori validi per il download. Per l'ID della chiave puoi utilizzare l'[ID della chiave](#) o l'[ARN della chiave](#), ma non puoi utilizzare un [nome alias](#) o l'[ARN alias](#) in questa operazione.

```
$ aws kms get-parameters-for-import \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --wrapping-algorithm RSA_AES_KEY_WRAP_SHA_256 \
  --wrapping-key-spec RSA_3072
```

Se il comando viene eseguito correttamente, verrà visualizzato un output simile al seguente:

```
{
  "ParametersValidTo": 1568290320.0,
  "PublicKey": "public key (base64 encoded)",
  "KeyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "ImportToken": "import token (base64 encoded)"
}
```

Per preparare i dati per il passaggio successivo, base64 decodifica la chiave pubblica e il token di importazione e salva i valori decodificati nei file.

Perché base64 decodifichi la chiave pubblica e importi il token:

1. Copiate la chiave pubblica codificata in base64 (rappresentata **public key (base64 encoded)** nell'output di esempio), incollatela in un nuovo file e salvate il file. Assegna al file un nome descrittivo, ad esempio `PublicKey.b64`.

2. Utilizza [OpenSSL](#) per decodificare su base64 il contenuto del file e salvare i dati decodificati in un nuovo file. L'esempio seguente decodifica i dati nel file salvato nel passaggio precedente (PublicKey.b64) e salva l'output in un nuovo file con nome WrappingPublicKey.bin.

```
$ openssl enc -d -base64 -A -in PublicKey.b64 -out WrappingPublicKey.bin
```

3. Copiate il token di importazione codificato in base64 (rappresentato *import token (base64 encoded)* nell'output di esempio), incollatelo in un nuovo file e salvate il file. Assegna al file un nome descrittivo, ad esempio importtoken.b64.
4. Utilizza [OpenSSL](#) per decodificare su base64 il contenuto del file e salvare i dati decodificati in un nuovo file. L'esempio seguente decodifica i dati nel file salvato nel passaggio precedente (ImportToken.b64) e salva l'output in un nuovo file con nome ImportToken.bin.

```
$ openssl enc -d -base64 -A -in importtoken.b64 -out ImportToken.bin
```

Passa a [Fase 3: crittografare il materiale delle chiavi](#).

Fase 3: crittografare il materiale delle chiavi

Dopo aver [scaricato la chiave pubblica e il token di importazione](#), esegue la crittografia del materiale della chiave utilizzando la chiave pubblica scaricata e l'algoritmo di wrapping specificato. Se devi sostituire la chiave pubblica o il token di importazione oppure modificare l'algoritmo di wrapping, devi scaricare una nuova chiave pubblica e un nuovo token di importazione. Per informazioni sulle chiavi pubbliche e sugli algoritmi di wrapping AWS KMS supportati, consulta e. [Selezione di una specifica della chiave pubblica di wrapping](#) [Selezione di un algoritmo di wrapping](#)

Il materiale delle chiavi deve essere in formato binario. Per informazioni dettagliate, consulta [Requisiti per il materiale della chiave importato](#).

Note

Per le coppie di chiavi asimmetriche, crittografa e importa solo la chiave privata. AWS KMS ricava la chiave pubblica dalla chiave privata.

La seguente combinazione NON è supportata: materiale della chiave ECC_NIST_P521, specifica della chiave di wrapping pubblica RSA_2048 e algoritmo di wrapping RSAES_OAEP_SHA_*.

Non puoi eseguire il wrapping del materiale della chiave ECC_NIST_P521 direttamente con una chiave di wrapping pubblica RSA_2048. Usa una chiave di wrapping più grande o un algoritmo di wrapping RSA_AES_KEY_WRAP_SHA_*.

Gli algoritmi di wrapping RSA_AES_KEY_WRAP_SHA_256 e RSA_AES_KEY_WRAP_SHA_1 non sono supportati nelle regioni della Cina.

Di solito, è possibile crittografare il materiale delle chiavi quando viene esportato dal modulo di sicurezza hardware (HSM) o dal sistema di gestione delle chiavi. Per informazioni su come esportare il materiale delle chiavi in formato binario, consulta la documentazione per il tuo HSM o per il sistema di gestione delle chiavi. È anche possibile consultare la sezione seguente che fornisce un proof of concept sull'utilizzo di OpenSSL.

Quando crittografi il materiale della chiave, usa lo stesso algoritmo di wrapping specificato quando hai [scaricato la chiave pubblica e il token di importazione](#). Per trovare l'algoritmo di wrapping specificato, consulta l'evento di registro per la richiesta associata. CloudTrail [GetParametersForImport](#)

Genera il materiale della chiave per i test

I comandi OpenSSL indicati di seguito generano il materiale della chiave di ogni tipo supportato per i test. Questi esempi vengono forniti solo a scopo di test e proof-of-concept dimostrazioni. Per i sistemi di produzione, usa un metodo più sicuro per generare e memorizzare il materiale della chiave, ad esempio un modulo di sicurezza hardware o un sistema di gestione delle chiavi.

Per convertire le chiavi private delle coppie di chiavi asimmetriche in formato con codifica DER, reindirizza il comando di generazione del materiale della chiave sul comando `openssl pkcs8` seguente. Il parametro `topk8` indica a OpenSSL di accettare una chiave privata come input e restituire una chiave in formato PKCS #8. (Il comportamento predefinito è l'opposto.)

```
openssl pkcs8 -topk8 -outform der -nocrypt
```

I comandi indicati di seguito generano il materiale della chiave per ogni tipo di chiave supportato.

- Chiavi di crittografia simmetrica (32 byte)

Questo comando genera una chiave simmetrica a 256 bit (stringa casuale di 32 byte) e la salva nel file `PlaintextKeyMaterial.bin`. Non è necessario codificare questo materiale della chiave.

```
openssl rand -out PlaintextKeyMaterial.bin 32
```

Solo nelle regioni cinesi, devi generare una chiave simmetrica a 128 bit (stringa casuale di 16 byte).

```
openssl rand -out PlaintextKeyMaterial.bin 16
```

- Chiavi HMAC

Questo comando genera una stringa di byte casuale della dimensione specificata. Non è necessario codificare questo materiale della chiave.

La lunghezza della chiave HMAC deve corrispondere alla lunghezza definita dalle specifiche della chiave KMS. Ad esempio, se la chiave KMS è HMAC_384, devi importare una chiave a 384 bit (48 byte).

```
openssl rand -out HMAC_224_PlaintextKey.bin 28
```

```
openssl rand -out HMAC_256_PlaintextKey.bin 32
```

```
openssl rand -out HMAC_384_PlaintextKey.bin 48
```

```
openssl rand -out HMAC_512_PlaintextKey.bin 64
```

- Chiavi private RSA

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:2048 | openssl pkcs8 -topk8 -outform der -nocrypt > RSA_2048_PrivateKey.der
```

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:3072 | openssl pkcs8 -topk8 -outform der -nocrypt > RSA_3072_PrivateKey.der
```

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 | openssl pkcs8 -topk8 -outform der -nocrypt > RSA_4096_PrivateKey.der
```

- Chiavi private EEC

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:P-256 | openssl pkcs8 -topk8 -outform der -nocrypt > ECC_NIST_P256_PrivateKey.der
```

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:P-384 | openssl pkcs8 -topk8 -outform der -nocrypt > ECC_NIST_P384_PrivateKey.der
```

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:P-521 | openssl pkcs8 -topk8  
-outform der -nocrypt > ECC_NIST_P521_PrivateKey.der
```

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:secp256k1 | openssl pkcs8 -  
topk8 -outform der -nocrypt > ECC_SECG_P256K1_PrivateKey.der
```

- SM2 chiavi private (solo regioni della Cina)

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:sm2 | openssl pkcs8 -topk8 -  
outform der -nocrypt > SM2_PrivateKey.der
```

Esempio di crittografia del materiale della chiave con OpenSSL

Gli esempi seguenti descrivono l'utilizzo di [OpenSSL](#) per crittografare il materiale della chiave con la chiave pubblica scaricata. Per crittografare il materiale chiave utilizzando una chiave SM2 pubblica (solo regioni della Cina), usa la [SM2OfflineOperationHelper](#) classe. Per ulteriori informazioni sui principali tipi di materiali supportati da ciascun algoritmo di wrapping, consulta [the section called “Selezione di un algoritmo di wrapping”](#)

Important

Questo esempio è solo una dimostrazione di proof of concept. Per i sistemi di produzione, utilizzare un metodo più sicuro (ad esempio un sistema di gestione delle chiavi commerciale o HSM) per generare e memorizzare il materiale delle chiavi.

La seguente combinazione NON è supportata: materiale della chiave ECC_NIST_P521, specifica della chiave di wrapping pubblica RSA_2048 e algoritmo di wrapping RSAES_OAEP_SHA_*.

Non puoi eseguire il wrapping del materiale della chiave ECC_NIST_P521 direttamente con una chiave di wrapping pubblica RSA_2048. Usa una chiave di wrapping più grande o un algoritmo di wrapping RSA_AES_KEY_WRAP_SHA_*.

RSAES_OAEP_SHA_1

AWS KMS supporta RSAES_OAEP_SHA_1 per chiavi di crittografia simmetriche (SYMMETRIC_DEFAULT), chiavi private a curva ellittica (ECC), chiavi private e chiavi HMAC. SM2

RSAES_OAEP_SHA_1 non è supportato per le chiavi private RSA. Inoltre, non puoi utilizzare una chiave di wrapping pubblica RSA_2048 con un algoritmo di wrapping RSAES_OAEP_SHA_* per eseguire il wrapping di una chiave privata ECC_NIST_P521 (secp521r1). Devi utilizzare una chiave di wrapping più grande o un algoritmo di wrapping RSA_AES_KEY_WRAP.

Nell'esempio seguente il materiale della chiave viene criptato con la [chiave pubblica scaricata](#) e l'algoritmo di wrapping RSAES_OAEP_SHA_1, quindi viene salvato nel file `EncryptedKeyMaterial.bin`.

In questo esempio:

- *WrappingPublicKey.bin* è il file che contiene la chiave pubblica di wrapping scaricata.
- *PlaintextKeyMaterial.bin* è il file che contiene il materiale della chiave da crittografare, ad esempio `PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin` o `ECC_NIST_P521_PrivateKey.der`.

```
$ openssl pkeyutl \  
  -encrypt \  
  -in PlaintextKeyMaterial.bin \  
  -out EncryptedKeyMaterial.bin \  
  -inkey WrappingPublicKey.bin \  
  -keyform DER \  
  -pubin \  
  -pkeyopt rsa_padding_mode:oaep \  
  -pkeyopt rsa_oaep_md:sha1
```

RSAES_OAEP_SHA_256

AWS KMS supporta RSAES_OAEP_SHA_256 per chiavi di crittografia simmetriche (SYMMETRIC_DEFAULT), chiavi private a curva ellittica (ECC), chiavi private e chiavi HMAC. SM2

RSAES_OAEP_SHA_256 non è supportato per le chiavi private RSA. Inoltre, non puoi utilizzare una chiave di wrapping pubblica RSA_2048 con un algoritmo di wrapping RSAES_OAEP_SHA_* per eseguire il wrapping di una chiave privata ECC_NIST_P521 (secp521r1). Devi utilizzare una chiave pubblica più grande o un algoritmo di wrapping RSA_AES_KEY_WRAP.

Nell'esempio seguente, il materiale della chiave viene criptato con la [chiave pubblica scaricata](#) e l'algoritmo di wrapping RSAES_OAEP_SHA_256, quindi viene salvato nel file `EncryptedKeyMaterial.bin`.

In questo esempio:

- *WrappingPublicKey.bin* è il file che contiene la chiave di wrapping pubblica scaricata. Se hai scaricato la chiave pubblica dalla console, questo file è denominato *wrappingKey_KMS_key_key_ID_timestamp* (ad esempio *wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909*).
- *PlaintextKeyMaterial.bin* è il file che contiene il materiale della chiave da crittografare, ad esempio *PlaintextKeyMaterial.bin*, *HMAC_384_PlaintextKey.bin* o *ECC_NIST_P521_PrivateKey.der*.

```
$ openssl pkeyutl \  
-encrypt \  
-in PlaintextKeyMaterial.bin \  
-out EncryptedKeyMaterial.bin \  
-inkey WrappingPublicKey.bin \  
-keyform DER \  
-pubin \  
-pkeyopt rsa_padding_mode:oaep \  
-pkeyopt rsa_oaep_md:sha256 \  
-pkeyopt rsa_mgf1_md:sha256
```

RSA_AES_KEY_WRAP_SHA_1

L'algoritmo di wrapping RSA_AES_KEY_WRAP_SHA_1 prevede due operazioni di crittografia.

1. Esegui la crittografia del materiale della chiave con una chiave simmetrica AES generata da te e un algoritmo di crittografia simmetrica AES.
2. Esegui la crittografia della chiave simmetrica AES che hai usato con la chiave pubblica che hai scaricato e l'algoritmo di wrapping RSAES_OAEP_SHA_1.

L'algoritmo di wrapping RSA_AES_KEY_WRAP_SHA_1 richiede OpenSSL versione 3.x o versione successiva.

1. Genera una chiave di crittografia simmetrica AES a 256-bit

Questo comando genera una chiave di crittografia simmetrica AES composta da 256 bit casuali e la salva nel file *aes-key.bin*

```
# Generate a 32-byte AES symmetric encryption key
```

```
$ openssl rand -out aes-key.bin 32
```

2. Esegui la crittografia del materiale della chiave con la chiave di crittografia simmetrica AES

Questo comando esegue la crittografia del materiale chiave con la chiave di crittografia simmetrica AES e salva il materiale della chiave crittografato nel file `key-material-wrapped.bin`.

In questo esempio di comando:

- *PlaintextKeyMaterial.bin* è il file che contiene il materiale della chiave da importare, ad esempio `PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin`, `RSA_3072_PrivateKey.der` o `ECC_NIST_P521_PrivateKey.der`.
- *aes-key.bin* è il file che contiene la chiave di crittografia simmetrica AES a 256 bit generata nel comando precedente.

```
# Encrypt your key material with the AES symmetric encryption key
$ openssl enc -id-aes256-wrap-pad \
    -K "$(xxd -p < aes-key.bin | tr -d '\n')" \
    -iv A65959A6 \
    -in PlaintextKeyMaterial.bin \
    -out key-material-wrapped.bin
```

3. Esegui la crittografia della chiave di crittografia simmetrica AES con la chiave pubblica

Questo comando esegue la crittografia della chiave di crittografia simmetrica AES con la chiave pubblica scaricata e l'algoritmo di wrapping `RSAES_OAEP_SHA_1`, applica la codifica DER e la salva nel file `aes-key-wrapped.bin`.

In questo esempio di comando:

- *WrappingPublicKey.bin* è il file che contiene la chiave di wrapping pubblica scaricata. Se hai scaricato la chiave pubblica dalla console, questo file è denominato `wrappingKey_KMS_key_key_ID_timestamp` (ad esempio `wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909`).
- *aes-key.bin* è il file che contiene la chiave di crittografia simmetrica AES a 256 bit generata nel primo comando in questa sequenza di esempi.

```
# Encrypt your AES symmetric encryption key with the downloaded public key
$ openssl pkeyutl \
  -encrypt \
  -in aes-key.bin \
  -out aes-key-wrapped.bin \
  -inkey WrappingPublicKey.bin \
  -keyform DER \
  -pubin \
  -pkeyopt rsa_padding_mode:oaep \
  -pkeyopt rsa_oaep_md:sha1 \
  -pkeyopt rsa_mgf1_md:sha1
```

4. Generare il file da importare

Concatena il file con il materiale della chiave crittografato e il file con la chiave AES crittografata. Salvali nel file `EncryptedKeyMaterial.bin`, che è il file che importerai in [Fase 4: importare il materiale delle chiavi](#).

In questo esempio di comando:

- *key-material-wrapped.bin* è il file che contiene il materiale della chiave crittografato.
- *aes-key-wrapped.bin* è il file che contiene la chiave di crittografia AES crittografata.

```
# Combine the encrypted AES key and encrypted key material in a file
$ cat aes-key-wrapped.bin key-material-wrapped.bin > EncryptedKeyMaterial.bin
```

RSA_AES_KEY_WRAP_SHA_256

L'algoritmo di wrapping `RSA_AES_KEY_WRAP_SHA_256` prevede due operazioni di crittografia.

1. Esegui la crittografia del materiale della chiave con una chiave simmetrica AES generata da te e un algoritmo di crittografia simmetrica AES.
2. Esegui la crittografia della chiave simmetrica AES che hai usato con la chiave pubblica che hai scaricato e l'algoritmo di wrapping `RSAES_OAEP_SHA_256`.

L'algoritmo di wrapping `RSA_AES_KEY_WRAP_SHA_256` richiede OpenSSL versione 3.x o versione successiva.

1. Genera una chiave di crittografia simmetrica AES a 256-bit

Questo comando genera una chiave di crittografia simmetrica AES composta da 256 bit casuali e la salva nel file `aes-key.bin`

```
# Generate a 32-byte AES symmetric encryption key
$ openssl rand -out aes-key.bin 32
```

2. Esegui la crittografia del materiale della chiave con la chiave di crittografia simmetrica AES

Questo comando esegue la crittografia del materiale chiave con la chiave di crittografia simmetrica AES e salva il materiale della chiave crittografato nel file `key-material-wrapped.bin`.

In questo esempio di comando:

- *PlaintextKeyMaterial.bin* è il file che contiene il materiale della chiave da importare, ad esempio `PlaintextKeyMaterial.bin`, `HMAC_384_PlaintextKey.bin`, `RSA_3072_PrivateKey.der` o `ECC_NIST_P521_PrivateKey.der`.
- *aes-key.bin* è il file che contiene la chiave di crittografia simmetrica AES a 256 bit generata nel comando precedente.

```
# Encrypt your key material with the AES symmetric encryption key
$ openssl enc -id-aes256-wrap-pad \
    -K "$(xxd -p < aes-key.bin | tr -d '\n')" \
    -iv A65959A6 \
    -in PlaintextKeyMaterial.bin \
    -out key-material-wrapped.bin
```

3. Esegui la crittografia della chiave di crittografia simmetrica AES con la chiave pubblica

Questo comando esegue la crittografia della chiave di crittografia simmetrica AES con la chiave pubblica scaricata e l'algoritmo di wrapping `RSAES_OAEP_SHA_256`, applica la codifica DER e la salva nel file `aes-key-wrapped.bin`.

In questo esempio di comando:

- *WrappingPublicKey.bin* è il file che contiene la chiave di wrapping pubblica scaricata. Se hai scaricato la chiave pubblica dalla console, questo file è denominato

wrappingKey_*KMS key_key_ID_timestamp* (ad esempio wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909).

- *aes-key.bin* è il file che contiene la chiave di crittografia simmetrica AES a 256 bit generata nel primo comando in questa sequenza di esempi.

```
# Encrypt your AES symmetric encryption key with the downloaded public key
$ openssl pkeyutl \
  -encrypt \
  -in aes-key.bin \
  -out aes-key-wrapped.bin \
  -inkey WrappingPublicKey.bin \
  -keyform DER \
  -pubin \
  -pkeyopt rsa_padding_mode:oaep \
  -pkeyopt rsa_oaep_md:sha256 \
  -pkeyopt rsa_mgf1_md:sha256
```

4. Generare il file da importare

Concatena il file con il materiale della chiave crittografato e il file con la chiave AES crittografata. Salvali nel file EncryptedKeyMaterial.bin, che è il file che importerai in [Fase 4: importare il materiale delle chiavi](#).

In questo esempio di comando:

- *key-material-wrapped.bin* è il file che contiene il materiale della chiave crittografato.
- *aes-key-wrapped.bin* è il file che contiene la chiave di crittografia AES crittografata.

```
# Combine the encrypted AES key and encrypted key material in a file
$ cat aes-key-wrapped.bin key-material-wrapped.bin > EncryptedKeyMaterial.bin
```

Passa a [Fase 4: importare il materiale delle chiavi](#).

Fase 4: importare il materiale delle chiavi

Dopo aver [crittografato il materiale della chiave](#), puoi importarlo per utilizzarlo con una AWS KMS key. Per importare il materiale della chiave, è possibile caricare il materiale della chiave crittografato da [Fase 3: crittografare il materiale delle chiavi](#) e il token di importazione scaricato in [Fase 2:](#)

[download della chiave pubblica di wrapping e del token di importazione](#). È necessario importare il materiale nella stessa chiave KMS che hai specificato quando hai [scaricato la chiave pubblica e il token di importazione](#). Quando il materiale della chiave viene importato correttamente, lo [stato chiave](#) della chiave KMS diventa `Enabled`, per cui puoi utilizzare la chiave KMS in operazioni crittografiche.

Quando importi il materiale della chiave, puoi [impostare un'ora di scadenza facoltativa](#) per il materiale della chiave. Quando il materiale della chiave scade, AWS KMS elimina tale materiale e la chiave KMS diventa inutilizzabile. Dopo aver importato il materiale della chiave, non potrai impostare, modificare o annullare la data di scadenza dell'importazione corrente. Per modificare questi valori, devi [reimportare](#) lo stesso materiale chiave.

Considerazioni chiave sull'importazione di materiali per tipo di chiave

Chiavi di crittografia simmetrica

Per tutte le chiavi KMS con `EXTERNAL` origine, il primo materiale chiave importato diventa corrente e ad esso associato permanentemente. Le chiavi di crittografia simmetriche con `EXTERNAL` origine supportano la rotazione su richiesta. È possibile associare più materiali chiave a chiavi importate che supportano la rotazione su richiesta. È necessario impostare il `importType` parametro su `NEW_KEY_MATERIAL` con l'[ImportKeyMaterial](#) operazione per associare un nuovo materiale chiave a una chiave KMS. Questo materiale chiave non viene associato in modo permanente alla chiave finché non si esegue l'[RotateKeyOnDemand](#) operazione. Fino ad allora, questo materiale chiave è `PENDING_ROTATION` disponibile. Il valore predefinito del `ImportType` parametro opzionale è `EXISTING_KEY_MATERIAL`. Quando omettete il `ImportType` parametro o lo specificate come `EXISTING_KEY_MATERIAL`, dovete importare un materiale chiave precedentemente associato alla chiave KMS.

Chiavi asimmetriche e HMAC

Per le chiavi KMS asimmetriche o HMAC con `EXTERNAL` origine, è possibile associare un solo materiale chiave alla chiave. AWS KMS rifiuterà le richieste API con il parametro [ImportKeyMaterial](#). `ImportType`

Chiavi multiregionali

È possibile importare nuovi materiali chiave in chiavi simmetriche multiregionali. A tale scopo, è necessario creare le chiavi `Region` primarie e di replica. Quindi è necessario importare il nuovo materiale chiave nella chiave regionale principale. Non è possibile importare direttamente nuovi materiali chiave nelle chiavi regionali di replica. Dopo aver importato il nuovo materiale chiave nella chiave di regione principale, è possibile importare lo stesso materiale chiave nelle chiavi di regione di replica.

Se un materiale chiave in una chiave di regione principale o di replica viene eliminato o scade, viene interessata solo quella chiave specifica. Le chiavi regionali di replica a cui sono stati importati tutti i materiali chiave associati in modo permanente possono essere utilizzate nelle operazioni crittografiche.

È possibile impostare o modificare la descrizione del materiale chiave per la chiave regionale principale solo con chiavi multiregionali.

Stati chiave durante l'importazione di materiale chiave

Quando tutti i materiali chiave associati in modo permanente a una chiave KMS vengono importati, la chiave KMS è disponibile per l'uso nelle operazioni crittografiche. Se uno qualsiasi di questi materiali chiave viene eliminato o lasciato scadere, lo stato della chiave KMS diventa `PendingImport` e la chiave è inutilizzabile per le operazioni crittografiche.

Per importare materiale chiave, puoi utilizzare la [AWS KMS console](#) o l'API. [ImportKeyMaterial](#) È possibile utilizzare l'API direttamente effettuando richieste HTTP oppure utilizzando un [AWS SDKs](#), [AWS Command Line Interface](#) o [AWS Strumenti per PowerShell](#).

Quando importate il materiale chiave, viene aggiunta una [ImportKeyMaterial](#) voce al AWS CloudTrail registro per registrare l'ImportKeyMaterial operazione. La CloudTrail voce è la stessa sia che si utilizzi la AWS KMS console che l'AWS KMS API.

Impostazione di una data di scadenza (facoltativo)

Quando importi il materiale della chiave per la chiave KMS, puoi impostare facoltativamente una data e un'ora di scadenza per il materiale della chiave fino a 365 giorni dalla data di importazione. Quando il materiale chiave importato scade, lo AWS KMS elimina. Questa azione modifica lo [stato chiave](#) della chiave KMS in `PendingImport`, impedendone l'utilizzo in qualsiasi operazione di crittografia. Per utilizzare la chiave KMS, devi [reimportare una copia del materiale della chiave originale](#).

Garantire che il materiale della chiave importato scada frequentemente può aiutarti a soddisfare i requisiti normativi, ma comporta un ulteriore rischio per i dati crittografati con la chiave KMS. Fino a quando non reimporti una copia del materiale della chiave originale, la chiave KMS con il materiale della chiave scaduto è inutilizzabile e tutti i dati crittografati con essa sono inaccessibili. Se per qualsiasi motivo non riesci a reimportare il materiale della chiave o se perdi il materiale della chiave originale, la chiave KMS è definitivamente inutilizzabile e i dati crittografati con essa non sono recuperabili.

Per mitigare questo rischio, assicuratevi che la copia del materiale chiave importato sia accessibile e progettate un sistema per eliminare e reimportare il materiale chiave prima che scada e interrompa il carico di lavoro. AWS Ti consigliamo di [configurare un allarme](#) per la scadenza del materiale della chiave importato, in modo da avere tutto il tempo necessario per reimportarlo prima che scada. [È inoltre possibile utilizzare CloudTrail i registri per controllare le operazioni di importazione \(e reimportazione\) del materiale chiave e l'eliminazione del materiale chiave importato, nonché l'operazione per eliminare il AWS KMS materiale chiave scaduto.](#)

AWS KMS non è possibile ripristinare, recuperare o riprodurre il materiale chiave eliminato. Invece di impostare una scadenza, puoi periodicamente [eliminare](#) e [reimportare](#) a livello di programmazione il materiale della chiave importato, tuttavia i requisiti per conservare una copia del materiale della chiave originale sono gli stessi.

Puoi determinare l'eventuale scadenza del materiale della chiave importato durante la sua importazione. Tuttavia è possibile attivare e disattivare la scadenza o impostare una nuova data di scadenza reimportando il materiale chiave. Utilizzate il `ExpirationModel` parametro di [ImportKeyMaterial](#) per attivare (`KEY_MATERIAL_EXPIRES`) e disattivare la scadenza (`KEY_MATERIAL_DOES_NOT_EXPIRE`) e il `ValidTo` parametro per impostare l'ora di scadenza. Il tempo massimo è di 365 giorni dall'importazione dei dati. Sebbene non sia prevista una data minima, l'ora di scadenza deve essere successiva alla data corrente.

Imposta la descrizione del materiale chiave

Alle chiavi di crittografia simmetriche con `EXTERNAL` origine possono essere associati più materiali chiave. È possibile specificare una descrizione facoltativa del materiale chiave quando si importa il materiale chiave in tali chiavi. La descrizione può essere utilizzata per tenere traccia di dove il materiale chiave corrispondente viene conservato all'esterno in modo duraturo. AWS KMS

È possibile impostare o modificare la descrizione del materiale chiave per la chiave regionale principale solo con chiavi multiregionali.

Reimporta il materiale chiave

Se gestisci una chiave KMS con materiale chiave importato, potresti dover reimportare il materiale della chiave. Puoi reimportare il materiale della chiave per sostituire il materiale della chiave in scadenza o eliminato oppure per modificare il modello di scadenza o la data di scadenza del materiale.

Puoi reimportare il materiale della chiave in qualsiasi momento e secondo qualsiasi pianificazione che soddisfi i requisiti di sicurezza. Non è necessario attendere che il materiale della chiave sia scaduto o prossimo alla scadenza.

La procedura per reimportare il materiale chiave è la stessa procedura utilizzata per importare il materiale chiave per la prima volta, con le seguenti eccezioni.

- Utilizzare una chiave KMS del servizio di gestione delle chiavi esistenti anziché creare una nuova chiave KMS del servizio di gestione delle chiavi. Puoi saltare la [fase 1](#) della procedura di importazione.
- Quando reimportate il materiale chiave, potete modificare il modello di scadenza, la data di scadenza e la descrizione del materiale chiave.

È possibile impostare o modificare la descrizione del materiale chiave per la chiave regionale principale solo con chiavi multiregionali.

Ogni volta che importi materiale della chiave in una chiave KMS, devi [scaricare e utilizzare una nuova chiave di wrapping e un nuovo token di importazione](#) per la chiave KMS. La procedura di wrapping non influisce sul contenuto del materiale della chiave, per cui puoi utilizzare chiavi pubbliche di wrapping diverse e algoritmi di wrapping diversi per importare lo stesso materiale della chiave.

Importa nuovo materiale chiave

Per eseguire la rotazione su richiesta su una chiave KMS di crittografia simmetrica con materiale chiave importato, devi prima importare nuovo materiale chiave, non precedentemente associato alla chiave.

- Chiavi a regione singola
 - Utilizzate l'[ImportKeyMaterial](#) operazione con il `ImportType` parametro impostato `NEW_KEY_MATERIAL` per eseguire questa operazione. Il materiale chiave importato in questo modo rimarrà `PENDING_ROTATION` invariato finché non si esegue l'[RotateKeyOnDemand](#) operazione o si ruota la chiave nel. Console di gestione AWS Una chiave KMS può avere al massimo un materiale chiave attivo in `PENDING_ROTATION` qualsiasi momento.
- Chiavi multi-regione
 - Per importare materiale chiave in una chiave multiregionale, devi prima importare il nuovo materiale chiave nella chiave regionale principale. Non è possibile importare direttamente nuovi

materiali chiave nelle chiavi regionali di replica. Dopo aver importato il nuovo materiale chiave nella chiave di regione principale, è possibile importare gli stessi materiali chiave nelle chiavi di regione di replica.

- Utilizzate l'[ImportKeyMaterial](#) operazione con il `ImportType` parametro impostato su **NEW_KEY_MATERIAL** per la chiave Region principale per eseguire questa operazione. Per la chiave Region di replica, utilizzare il **EXISTING_KEY_MATERIAL** parametro `ImportType` per l'`ImportKeyMaterial` operazione.
- Il materiale chiave per le chiavi multiregionali deve essere importato in tutte le chiavi di regione di replica e le chiavi di regione primarie prima che lo stato del materiale chiave cambi in stato. `PENDING_ROTATION` Fino ad allora, lo stato del nuovo materiale chiave è. `PENDING_MULTI_REGION_IMPORT_AND_ROTATION`
Una chiave KMS può avere al massimo un materiale chiave `PENDING_ROTATION` o `PENDING_MULTI_REGION_IMPORT_AND_ROTATION` uno stato alla volta.

Importazione del materiale della chiave (console)

È possibile utilizzare il Console di gestione AWS per importare materiale chiave.

1. Se ti trovi nella pagina Carica il materiale della chiave di wrapping, passa a [Step 8](#).
2. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
3. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
4. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
5. Scegli l'alias o l'ID chiave della chiave KMS per la quale hai scaricato il token di importazione e la chiave pubblica.
6. Scegli la tab Configurazione crittografica e visualizzane i valori. Le tab si trovano nella pagina dei dettagli di una chiave KMS sotto la sezione Configurazione generale.

Puoi importare il materiale della chiave solo in chiavi KMS con Origine Esterna (Importa materiale chiave). Per ulteriori informazioni sulla creazione di chiavi KMS con materiale della chiave importato, consulta [Importazione di materiale chiave per le AWS KMS chiavi](#).

7. Per le chiavi asimmetriche e HMAC, scegliete la scheda Materiale chiave, quindi scegliete Importa materiale chiave. Per le chiavi di crittografia simmetriche, scegli la scheda Materiale chiave e rotazioni. Quindi, scegli Importa materiale chiave iniziale o Importa nuovo materiale

chiave o Reimporta materiale chiave. L'opzione Reimporta il materiale chiave è disponibile nel **Actions** menu della tabella dei materiali chiave.

Se hai scaricato il materiale della chiave, il token di importazione e hai crittografato il materiale della chiave, scegli **Avanti**.

 Note

Per le chiavi multiregionali, è necessario prima importare il nuovo materiale chiave nella chiave Region principale. Quindi è possibile importare lo stesso materiale chiave nelle chiavi regionali di replica.

8. Nella sezione **Materiale della chiave crittografato e token di importazione**, effettua le operazioni seguenti.
 - a. In **Materiale della chiave di wrapping**, scegli **Scegli file**. Quindi caricare il file contenente il materiale delle chiavi (crittografato) sottoposto a wrapping.
 - b. In **Token di importazione**, scegli **Scegli file**. Caricare il file contenente il token di importazione [scaricato](#).
9. Nella sezione **Choose an expiration option** (Scegli un'opzione di scadenza) stabilire se il materiale della chiave scade. Per impostare una data e un'ora di scadenza, scegliere **Key material expires** (Il materiale chiave scade) e utilizzare il calendario per selezionare una data e un'ora. Puoi specificare una data fino a 365 giorni dalla data e dall'ora corrente.
10. Per le chiavi di crittografia simmetriche, è possibile specificare facoltativamente una descrizione per il materiale chiave da importare.
11. Scegliete **Importa materiale chiave**.

Importa materiale chiave (AWS KMS API)

Per importare materiale chiave, utilizzate l'[ImportKeyMaterial](#) operazione. Gli esempi seguenti utilizzano la [AWS CLI](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

Per utilizzare questo esempio:

È possibile impostare o modificare la descrizione del materiale chiave per la chiave Region principale solo con chiavi multiregionali.

1. Sostituisci `1234abcd-12ab-34cd-56ef-1234567890ab` con l'ID chiave della chiave KMS specificata per il download della chiave pubblica e del token di importazione. Per identificare la chiave KMS utilizza [l'ID chiave](#) o [l'ARN di chiave](#). Per questa operazione, non puoi utilizzare un [nome alias](#) o un [ARN alias](#).
2. Sostituire `EncryptedKeyMaterial.bin` con il nome del file che contiene il materiale della chiave crittografato.
3. Sostituire `ImportToken.bin` con il nome del file che contiene il token di importazione.
4. Se desideri che il materiale della chiave importato abbia una scadenza, imposta il valore del parametro `expiration-model` sul valore predefinito, `KEY_MATERIAL_EXPIRES`, oppure ometti il parametro `expiration-model`. Quindi, sostituisci il valore del parametro `valid-to` con la data e l'ora in cui desideri che il materiale della chiave scada. La data e l'ora possono arrivare fino a 365 giorni dal momento della richiesta.

```
$ aws kms import-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --encrypted-key-material fileb://EncryptedKeyMaterial.bin \
  --import-token fileb://ImportToken.bin \
  --expiration-model KEY_MATERIAL_EXPIRES \
  --valid-to 2023-06-17T12:00:00-08:00
```

Se desideri che il materiale della chiave importato abbia una scadenza, imposta il valore del parametro `expiration-model` su `KEY_MATERIAL_DOES_NOT_EXPIRE` e ometti il parametro `valid-to` dal comando.

```
$ aws kms import-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --encrypted-key-material fileb://EncryptedKeyMaterial.bin \
  --import-token fileb://ImportToken.bin \
  --expiration-model KEY_MATERIAL_DOES_NOT_EXPIRE
```

5. Se desideri importare nuovo materiale chiave, non precedentemente associato alla chiave KMS, imposta il `ImportType` parametro su `NEW_KEY_MATERIAL`. Questa opzione può essere utilizzata solo con chiavi di crittografia simmetriche. Per queste chiavi, puoi anche utilizzare il `KeyMaterialDescription` parametro opzionale per impostare una descrizione del materiale chiave importato nel seguente esempio da riga di comando:

```
$ aws kms import-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --encrypted-key-material fileb://EncryptedKeyMaterial.bin \
  --import-token fileb://ImportToken.bin \
  --expiration-model KEY_MATERIAL_EXPIRES \
```

```
--valid-to 2023-06-17T12:00:00-08:00 \  
--import-type NEW_KEY_MATERIAL \  
--key-material-description "Q2 2025 Rotation"
```

 Tip

Se l'esito del comando non è positivo, potresti visualizzare un'`KMSInvalidStateException` o un'`NotFoundException`. Puoi ritentare la richiesta.

Crea una chiave KMS in un AWS CloudHSM archivio di chiavi

Dopo aver creato un archivio di AWS CloudHSM chiavi, puoi crearlo AWS KMS keys nel tuo archivio di chiavi. Devono essere [chiavi KMS con crittografia simmetrica](#) con materiale chiave generato. AWS KMS Non è possibile creare [chiavi KMS asimmetriche](#), [chiavi KMS HMAC](#) o chiavi KMS con [materiale della chiave importato](#) in un archivio delle chiavi personalizzate. Inoltre, non è possibile utilizzare chiavi KMS di crittografia simmetrica in un archivio delle chiavi personalizzate per generare coppie di chiavi di dati asimmetriche. KMS non può comunicare con gli archivi di chiavi. IPv6 AWS CloudHSM

Per creare una chiave KMS in un AWS CloudHSM key store, l'archivio AWS CloudHSM chiavi deve essere [connesso al AWS CloudHSM cluster associato e il cluster](#) deve contenere almeno due chiavi attive HSMs in diverse zone di disponibilità. Per trovare lo stato e il numero di connessioni HSMs, visualizza la [pagina degli archivi di AWS CloudHSM chiavi](#) in. Console di gestione AWS Quando si utilizzano le operazioni API, utilizzare l'[DescribeCustomKeyStores](#) operazione per verificare che l'archivio delle AWS CloudHSM chiavi sia connesso. Per verificare il numero di persone attive HSMs nel cluster e le relative zone di disponibilità, utilizzate l' AWS CloudHSM [DescribeClusters](#) operazione.

Quando crei una chiave KMS nel tuo archivio AWS CloudHSM chiavi, AWS KMS crea la chiave KMS in. AWS KMS Tuttavia, crea il materiale chiave per la chiave KMS nel cluster associato. AWS CloudHSM In particolare, AWS KMS accede al cluster come [kmsuserCU che hai](#) creato. Crea quindi una chiave simmetrica AES (Advanced Encryption Standard) a 256 bit non estraibile e persistente nel cluster. AWS KMS imposta il valore [dell'attributo dell'etichetta di chiave](#), visibile solo nel cluster, sull'Amazon Resource Name (ARN) della chiave KMS.

Quando il comando riesce, lo [stato di chiave](#) della nuova chiave KMS è `Enabled` e la relativa origine è `AWS_CLOUDHSM`. Non puoi modificare l'origine di una chiave KMS dopo averla creata. Quando si

visualizza una chiave KMS in un AWS CloudHSM key store della AWS KMS console o utilizzando l'[DescribeKey](#) operazione, è possibile visualizzare le proprietà tipiche, come l'ID della chiave, lo stato della chiave e la data di creazione. Ma puoi anche visualizzare l'ID store chiavi personalizzate ed eventualmente l'ID del cluster AWS CloudHSM .

Se il tentativo di creare una chiave KMS nell'archivio delle AWS CloudHSM chiavi fallisce, utilizza il messaggio di errore per determinarne la causa. Potrebbe indicare che l'archivio AWS CloudHSM chiavi non è connesso (`CustomKeyStoreInvalidStateException`) o HSMS che il AWS CloudHSM cluster associato non ha le due chiavi attive necessarie per questa operazione (`CloudHsmClusterInvalidConfigurationException`). Per assistenza, consulta [Risoluzione di problemi relativi a store delle chiavi personalizzate](#).

Per un esempio del AWS CloudTrail registro dell'operazione che crea una chiave KMS in un archivio di AWS CloudHSM chiavi, vedi [CreateKey](#).

Crea una nuova chiave KMS nel tuo key store CloudHSM

Puoi creare una chiave KMS di crittografia simmetrica nel tuo archivio di AWS CloudHSM chiavi nella AWS KMS console o utilizzando l'operazione. [CreateKey](#)

Utilizzo della console AWS KMS

Utilizzare la procedura seguente per creare una chiave KMS di crittografia simmetrica in un AWS CloudHSM archivio di chiavi.

Note

Non includere informazioni riservate o sensibili nell'alias, nella descrizione o nei tag. Questi campi possono apparire in testo semplice nei CloudTrail log e in altri output.

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
4. Scegli Crea chiave.

5. Scegliere Symmetric (Simmetrica).
6. In Key usage (Utilizzo della chiave), l'opzione Encrypt and decrypt (Crittografa e decrittografa) è selezionata per default. Non modificarla.
7. Scegliere Advanced options (Opzioni avanzate).
8. Per Origine del materiale della chiave, scegli Archivio di chiavi AWS CloudHSM .

Non è possibile creare una chiave multiregionale in un archivio di chiavi. AWS CloudHSM

9. Scegli Next (Successivo).
10. Seleziona un archivio di AWS CloudHSM chiavi per la tua nuova chiave KMS. Per creare un nuovo archivio AWS CloudHSM chiavi, scegli Crea archivio chiavi personalizzato.

Lo stato dell'archivio AWS CloudHSM chiavi selezionato deve avere lo stato Connesso. Il AWS CloudHSM cluster associato deve essere attivo e contenerne almeno due attivi HSMs in diverse zone di disponibilità.

Per informazioni sulla connessione di un archivio di AWS CloudHSM chiavi, consulta [Disconnetti un archivio AWS CloudHSM chiavi](#). Per informazioni sull'aggiunta HSMs, consulta [Aggiungere un HSM](#) nella Guida per l'AWS CloudHSM utente.

11. Scegli Next (Successivo).
12. Digita un alias ed eventualmente una descrizione per la chiave KMS.
13. (Facoltativo). Nella pagina Add Tags (Aggiungi tag), aggiungi i tag che identificano o categorizzano la chiave KMS.

Quando aggiungi tag alle tue AWS risorse, AWS genera un rapporto sull'allocazione dei costi con utilizzo e costi aggregati per tag. I tag possono essere utilizzati anche per controllare l'accesso a una chiave KMS. Per informazioni sull'assegnazione di tag delle chiavi KMS, consulta [Tag in AWS KMS](#) e [ABAC per AWS KMS](#).

14. Scegli Next (Successivo).
15. Nella sezione amministratori delle chiavi, seleziona utenti IAM e ruoli IAM che possono gestire la chiave KMS. Per ulteriori informazioni, consulta [Consente agli amministratori delle chiavi di amministrare la chiave KMS](#).

Note

Le policy IAM possono fornire ad altri ruoli e utenti IAM l'autorizzazione per utilizzare la chiave KMS.

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

La AWS KMS console aggiunge gli amministratori chiave alla politica chiave sotto l'identificatore dell'istruzione. "Allow access for Key Administrators" La modifica di questo identificatore di istruzione potrebbe influire sul modo in cui la console visualizza gli aggiornamenti apportati all'istruzione.

16. (Facoltativo) Per impedire a questi amministratori delle chiavi di eliminare questa chiave KMS, nella parte inferiore della pagina deseleziona Allow key administrators to delete this key. (Consenti agli amministratori delle chiavi di eliminare questa chiave).
17. Scegli Next (Successivo).
18. [Nella sezione Questo account, seleziona gli utenti e i ruoli IAM Account AWS che possono utilizzare la chiave KMS nelle operazioni crittografiche.](#) Per ulteriori informazioni, consulta [Consente agli utenti della chiave di utilizzare la chiave KMS.](#)

Note

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

La AWS KMS console aggiunge gli utenti chiave alla politica chiave sotto gli "Allow use of the key" identificatori di dichiarazione e. "Allow attachment of persistent resources" La modifica di questi identificatori delle istruzioni potrebbe influire sul modo in cui la console visualizza gli aggiornamenti apportati all'istruzione.

19. (Facoltativo) È possibile consentire ad altri Account AWS di utilizzare questa chiave KMS per operazioni crittografiche. A tale scopo, nella Account AWS sezione Altro in fondo alla pagina, scegli Aggiungi un altro account Account AWS e inserisci l' Account AWS ID di un account esterno. Per aggiungere più account esterni, ripetere questo passaggio.

 Note

Gli amministratori dell'altro Account AWS devono inoltre consentire l'accesso alla chiave KMS creando policy IAM per i propri utenti. Per ulteriori informazioni, consulta [Autorizzazione per gli utenti in altri account di utilizzare una chiave KMS](#).

20. Scegli Next (Successivo).
21. Consulta le principali dichiarazioni politiche relative alla chiave. Per apportare modifiche alla politica chiave, seleziona Modifica.
22. Scegli Next (Successivo).
23. Esaminare le principali impostazioni scelte. È comunque possibile tornare indietro e modificare tutte le impostazioni.
24. Al termine, scegli Crea filtro.

Quando la procedura ha esito positivo, il display mostra la nuova chiave KMS nell'archivio delle AWS CloudHSM chiavi che hai scelto. Quando scegli il nome o l'alias della nuova chiave KMS, la scheda Configurazione crittografica nella relativa pagina dei dettagli mostra l'origine della chiave KMS (AWS CloudHSM), il nome, l'ID e il tipo dell'archivio chiavi personalizzato e l'ID del cluster. AWS CloudHSM Se la procedura ha esito negativo, viene visualizzato un messaggio di errore che descrive l'errore.

 Tip

Per agevolare l'identificazione delle chiavi KMS in uno store delle chiavi personalizzate, nella pagina Chiavi gestite cliente, aggiungi la colonna ID dell'archivio delle chiavi personalizzate alla visualizzazione. Fai clic sull'icona che raffigura un ingranaggio in alto a destra e seleziona Custom key store ID (ID store chiavi personalizzate). Per informazioni dettagliate, vedi [Personalizza la visualizzazione della console](#).

AWS KMS Utilizzo dell'API

Per creare una nuova AWS KMS key (chiave KMS) nel tuo AWS CloudHSM key store, usa l'[CreateKey](#) operazione. Utilizza il parametro CustomKeyId per identificare lo store e specifica AWS_CLOUDHSM per Origin.

Potresti anche voler utilizzare il parametro `Policy` per specificare una policy delle chiavi. Puoi modificare la politica chiave ([PutKeyPolicy](#)) e aggiungere elementi opzionali, come una [descrizione](#) e dei [tag](#) in qualsiasi momento.

Gli esempi in questa sezione utilizzano [AWS Command Line Interface \(AWS CLI\)](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

L'esempio seguente inizia con una chiamata all'[DescribeCustomKeyStores](#) operazione per verificare che l'archivio AWS CloudHSM chiavi sia connesso al AWS CloudHSM cluster associato. Per impostazione predefinita, questa operazione restituisce tutti gli store delle chiavi personalizzate presenti nel tuo account e nella regione. Per descrivere solo un particolare archivio di AWS CloudHSM chiavi, utilizzate il relativo `CustomKeyName` parametro `CustomKeyId` o (ma non entrambi).

Prima di eseguire questo comando, sostituisci l'ID store chiavi personalizzate di esempio con un ID valido.

Note

Non includere informazioni riservate o sensibili nei campi `Description` o `Tags`. Questi campi possono apparire in testo semplice nei CloudTrail log e in altri output.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
  "CustomKeyStores": [
    {
      "CustomKeyId": "cks-1234567890abcdef0",
      "CustomKeyName": "ExampleKeyStore",
      "CustomKeyType": "AWS CloudHSM key store",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "CONNECTED"
    }
  ],
}
```

Il comando di esempio successivo utilizza l'[DescribeClusters](#) operazione per verificare che il AWS CloudHSM cluster associato a `ExampleKeyStore` (`cluster-1a23b4cdefg`) ne abbia almeno due attivi. `HSMS` Se il cluster ne ha meno di due, l'operazione ha esito negativo. `HSMS CreateKey`

```
$ aws cloudhsmv2 describe-clusters
{
  "Clusters": [
    {
      "SubnetMapping": {
        ...
      },
      "CreateTimestamp": 1507133412.351,
      "ClusterId": "cluster-1a23b4cdefg",
      "SecurityGroup": "sg-865af2fb",
      "HsmType": "hsm1.medium",
      "VpcId": "vpc-1a2b3c4d",
      "BackupPolicy": "DEFAULT",
      "Certificates": {
        "ClusterCertificate": "-----BEGIN CERTIFICATE-----\...\n-----END
CERTIFICATE-----\n"
      },
      "Hsms": [
        {
          "AvailabilityZone": "us-west-2a",
          "EniIp": "10.0.1.11",
          "ClusterId": "cluster-1a23b4cdefg",
          "EniId": "eni-ea8647e1",
          "StateMessage": "HSM created.",
          "SubnetId": "subnet-a6b10bd1",
          "HsmId": "hsm-abcdefghijkl",
          "State": "ACTIVE"
        },
        {
          "AvailabilityZone": "us-west-2b",
          "EniIp": "10.0.0.2",
          "ClusterId": "cluster-1a23b4cdefg",
          "EniId": "eni-ea8647e1",
          "StateMessage": "HSM created.",
          "SubnetId": "subnet-b6b10bd2",
          "HsmId": "hsm-zyxwvutsrq",
          "State": "ACTIVE"
        }
      ],
      "State": "ACTIVE"
    }
  ]
}
```

Questo comando di esempio utilizza l'[CreateKey](#) operazione per creare una chiave KMS in un archivio di AWS CloudHSM chiavi. Per creare una chiave KMS in un archivio AWS CloudHSM chiavi, è necessario fornire l'ID dell'archivio chiavi personalizzato dell'archivio AWS CloudHSM chiavi e specificare il `Origin` valore di `AWS_CLOUDHSM`.

La risposta include l'archivio IDs di chiavi personalizzato e il AWS CloudHSM cluster.

Prima di eseguire questo comando, sostituisci l'ID store chiavi personalizzate di esempio con un ID valido.

```
$ aws kms create-key --origin AWS_CLOUDHSM --custom-key-store-id cks-1234567890abcdef0
{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": 1.499288695918E9,
    "Description": "Example key",
    "Enabled": true,
    "MultiRegion": false,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "Origin": "AWS_CLOUDHSM",
    "CloudHsmClusterId": "cluster-1a23b4cdefg",
    "CustomKeyId": "cks-1234567890abcdef0",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ]
  }
}
```

Crea una chiave KMS in archivi di chiavi esterni

Dopo aver [creato](#) e [collegato](#) l'archivio chiavi esterno, puoi creare AWS KMS keys nel tuo archivio chiavi. Devono essere [chiavi KMS con crittografia simmetrica](#) con un valore di origine impostato su External key store (Archivio delle chiavi esterne) (`EXTERNAL_KEY_STORE`). Non è possibile creare [chiavi KMS asimmetriche](#), [chiavi KMS HMAC](#) o chiavi KMS con [materiale della chiave](#)

[importato](#) in un archivio delle chiavi personalizzate. Inoltre, non è possibile utilizzare chiavi KMS di crittografia simmetrica in un archivio delle chiavi personalizzate per generare coppie di chiavi di dati asimmetriche.

Una chiave KMS in un archivio delle chiavi esterne potrebbe avere una latenza, una durata e una disponibilità inferiori rispetto a una chiave KMS standard, perché dipende da componenti situati al di fuori di AWS. Prima di creare o utilizzare una chiave KMS in un archivio delle chiavi esterne, verifica che sia necessaria una chiave con proprietà di archivio delle chiavi esterne.

Note

Alcuni gestori delle chiavi esterne offrono un metodo più semplice per creare chiavi KMS in un archivio delle chiavi esterne. Per ulteriori informazioni, consulta la documentazione del gestore delle chiavi esterne.

Per creare una chiave KMS nell'archivio delle chiavi esterne, devi specificare quanto segue:

- L'ID dell'archivio delle chiavi esterne.
- Un'[origine del materiale della chiave](#) con valore External key store (Archivio delle chiavi esterne) (EXTERNAL_KEY_STORE).
- L'ID di una [chiave esterna](#) esistente nel [gestore delle chiavi esterne](#) associato all'archivio delle chiavi esterne. Questa chiave esterna funge da materiale della chiave per la chiave KMS. Non puoi modificare l'ID della chiave esterna dopo aver creato la chiave KMS.

AWS KMS fornisce l'ID della chiave esterna al proxy dell'archivio chiavi esterno nelle richieste di operazioni di crittografia e decrittografia. AWS KMS non può accedere direttamente al gestore di chiavi esterno o a nessuna delle sue chiavi crittografiche.

Oltre alla chiave esterna, una chiave KMS in un archivio di chiavi esterno contiene anche materiale AWS KMS chiave. Tutti i dati crittografati con la chiave KMS vengono prima crittografati AWS KMS utilizzando il materiale chiave della AWS KMS chiave e poi dal gestore delle chiavi esterno utilizzando la chiave esterna. Questo processo di [doppia crittografia](#) garantisce che il testo criptato protetto da una chiave KMS in un archivio delle chiavi esterne sia almeno altrettanto sicuro del testo criptato protetto solo da AWS KMS. Per informazioni dettagliate, vedi [Funzionamento degli archivi delle chiavi esterne](#).

Quando l'operazione `CreateKey` ha esito positivo, lo [stato chiave](#) della nuova chiave KMS è `Enabled`. Quando [visualizzi una chiave KMS in un archivio delle chiavi esterne](#) puoi vedere proprietà tipiche, come l'ID, le [specifiche della chiave](#), nonché [l'utilizzo della chiave](#), [lo stato della chiave](#) e la data di creazione. Puoi inoltre visualizzare l'ID e lo [stato di connessione](#) dell'archivio delle chiavi esterne e l'ID della chiave esterna.

Se il tentativo di creare una chiave KMS nell'archivio delle chiavi esterne ha esito negativo, utilizza il messaggio di errore per determinarne la causa. Potrebbe indicare che l'archivio delle chiavi esterne non è connesso (`CustomKeyStoreInvalidStateException`), che il proxy dell'archivio delle chiavi esterne non è in grado di trovare una chiave esterna con l'ID della chiave esterna specificato (`XksKeyNotFoundException`) o che la chiave esterna è già associata a una chiave KMS nello stesso archivio con l'eccezione `XksKeyAlreadyInUseException`.

Per un esempio del AWS CloudTrail registro dell'operazione che crea una chiave KMS in un archivio di chiavi esterno, vedi. [CreateKey](#)

Argomenti

- [Requisiti per una chiave KMS in un archivio delle chiavi esterne](#)
- [Crea una nuova chiave KMS nel tuo archivio di chiavi esterno](#)

Requisiti per una chiave KMS in un archivio delle chiavi esterne

Per creare una chiave KMS in un archivio delle chiavi esterne, sono necessarie le seguenti proprietà dell'archivio delle chiavi esterne, della chiave KMS e della chiave esterna che funge da materiale della chiave crittografica esterna per la chiave KMS.

Requisiti dell'archivio delle chiavi esterne

- Deve essere collegato al relativo proxy dell'archivio delle chiavi esterne.

Per visualizzare lo [stato di connessione](#) dell'archivio delle chiavi esterne, consulta [Visualizza gli archivi di chiavi esterni](#). Per connettere l'archivio delle chiavi esterne, consulta [Connect e disconnetti gli archivi di chiavi esterni](#).

Requisiti della chiave KMS

Dopo aver creato la chiave KMS, non puoi più modificare queste proprietà.

- Specifica della chiave `SYMMETRIC_DEFAULT`

- Utilizzo della chiave: ENCRYPT_DECRYPT
- Origine del materiale della chiave: EXTERNAL_KEY_STORE
- Multi regione: FALSE

Requisiti della chiave esterna

- Chiave crittografica AES a 256 bit (256 bit casuali). Il valore di KeySpec per la chiave esterna deve essere AES_256.
- Attivata e disponibile per l'uso. Il valore di Status per la chiave esterna deve essere ENABLED.
- Configurata per la crittografia e la decrittografia. Il valore di KeyUsage per la chiave esterna deve includere ENCRYPT e DECRYPT.
- Utilizzata solo con questa chiave KMS. Ciascuna KMS key in un archivio delle chiavi esterne deve essere associata a una chiave esterna diversa.

AWS KMS consiglia inoltre di utilizzare la chiave esterna esclusivamente per l'archivio di chiavi esterno. Questa restrizione semplifica l'identificazione e la risoluzione dei problemi relativi alla chiave.

- Accessibile dal [proxy dell'archivio delle chiavi esterne](#) per l'archivio delle chiavi esterne.

Se il proxy dell'archivio delle chiavi esterne non riesce a trovare la chiave utilizzando l'ID della chiave esterna specificato, l'operazione CreateKey ha esito negativo.

- È in grado di gestire il traffico previsto Servizi AWS generato dall'utilizzo. AWS KMS consiglia di predisporre chiavi esterne per gestire fino a 1800 richieste al secondo.

Crea una nuova chiave KMS nel tuo archivio di chiavi esterno

Puoi creare una nuova chiave KMS nel tuo archivio chiavi esterno nella AWS KMS console o utilizzando l'[CreateKey](#) operazione.

Utilizzo della console AWS KMS

Esistono due modi per creare una chiave KMS in un archivio delle chiavi esterne.

- Metodo 1 (consigliato): scegli un archivio di chiavi esterno e crea una chiave KMS all'interno dell'archivio di chiavi esterno.
- Metodo 2: crea una chiave KMS e indica che si trova in un archivio di chiavi esterno.

Se utilizzi il Metodo 1, in cui scegli il tuo archivio di chiavi esterno prima di creare la chiave, AWS KMS sceglie automaticamente tutte le proprietà della chiave KMS richieste e inserisce l'ID del tuo archivio di chiavi esterno. Questo metodo evita errori che potrebbero verificarsi durante la creazione della chiave KMS.

Note

Non includere informazioni riservate o sensibili nell'alias, nella descrizione o nei tag. Questi campi possono apparire in testo semplice nei CloudTrail log e in altri output.

Metodo 1 (consigliato): inizia dall'archivio di chiavi esterno

Per utilizzare questo metodo, scegli l'archivio delle chiavi esterne, quindi crea una chiave KMS. La AWS KMS console sceglie per te tutte le proprietà richieste e inserisce l'ID del tuo archivio di chiavi esterno. Questo metodo evita molti errori che potrebbero verificarsi durante la creazione della chiave KMS.

1. [Accedi a Console di gestione AWS e apri la console AWS Key Management Service \(AWS KMS\) in /kms. https://console.aws.amazon.com](https://console.aws.amazon.com/kms)
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel pannello di navigazione, scegli Custom key stores (Archivi delle chiavi personalizzate), External key stores (Archivi delle chiavi esterne).
4. Scegli il nome dell'archivio delle chiavi esterne.
5. Nell'angolo in alto a destra, scegli Create a KMS key in this key store (Crea una chiave KMS in questo archivio delle chiavi).

Se l'archivio delle chiavi esterne non è connesso, ti verrà richiesto di collegarlo. Se il tentativo di connessione fallisce, è necessario risolvere il problema e connettere l'archivio delle chiavi esterne prima di poter creare una nuova chiave KMS al suo interno.

Se l'archivio delle chiavi esterne è connesso, verrai reindirizzato alla pagina Customer managed keys (Chiavi gestite dal cliente) per creare una chiave. I valori di Key configuration (Configurazione della chiave) richiesti sono già stati selezionati automaticamente. Inoltre, è già stato inserito l'ID dell'archivio delle chiavi personalizzate per l'archivio delle chiavi esterne, sebbene sia possibile modificarlo.

6. Inserisci l'ID chiave di una [chiave esterna](#) nel [gestore delle chiavi esterne](#). Questa chiave esterna deve [soddisfare i requisiti](#) per l'uso con una chiave KMS. Non puoi modificare questo valore dopo la creazione della chiave.

Se la chiave esterna è multipla IDs, inserisci l'ID della chiave utilizzato dal proxy dell'archivio chiavi esterno per identificare la chiave esterna.

7. Conferma che intendi creare una chiave KMS nell'archivio delle chiavi esterne specificato.
8. Scegli Next (Successivo).

Il resto di questa procedura è uguale alla [creazione di una chiave KMS standard](#).

9. Digita un alias (obbligatorio) e, facoltativamente, una descrizione della chiave KMS.
10. (Facoltativo). Nella pagina Add Tags (Aggiungi tag), aggiungi i tag che identificano o categorizzano la chiave KMS.

Quando aggiungi tag alle tue AWS risorse, AWS genera un rapporto sull'allocazione dei costi con utilizzo e costi aggregati per tag. I tag possono essere utilizzati anche per controllare l'accesso a una chiave KMS. Per informazioni sull'assegnazione di tag delle chiavi KMS, consulta [Tag in AWS KMS](#) e [ABAC per AWS KMS](#).

11. Scegli Next (Successivo).
12. Nella sezione amministratori delle chiavi, seleziona utenti IAM e ruoli IAM che possono gestire la chiave KMS. Per ulteriori informazioni, consulta [Consente agli amministratori delle chiavi di amministrare la chiave KMS](#).

Note

Le policy IAM possono fornire ad altri ruoli e utenti IAM l'autorizzazione per utilizzare la chiave KMS.

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

13. (Facoltativo) Per impedire a questi amministratori delle chiavi di eliminare tale chiave KMS, deseleziona la casella di controllo Allow key administrators to delete this key. (Consenti agli amministratori delle chiavi di eliminare questa chiave).

L'eliminazione di una chiave KMS è un'operazione distruttiva e irreversibile che può rendere il testo criptato irrecuperabile. Non puoi ricreare una chiave KMS simmetrica in un archivio delle chiavi esterne, anche se disponi del materiale della chiave. L'eliminazione di una chiave KMS non ha alcun effetto sulla chiave esterna associata. Per informazioni sull'eliminazione di una chiave KMS da un archivio di chiavi esterno, consulta [Considerazioni speciali](#) per l'eliminazione delle chiavi.

14. Scegli Next (Successivo).
15. [Nella sezione Questo account, seleziona gli utenti e i ruoli IAM Account AWS che possono utilizzare la chiave KMS nelle operazioni crittografiche.](#) Per ulteriori informazioni, consulta [Consente agli utenti della chiave di utilizzare la chiave KMS.](#)

 Note

Le policy IAM possono fornire ad altri ruoli e utenti IAM l'autorizzazione per utilizzare la chiave KMS.

Le best practice di IAM disincentivano l'uso di utenti IAM con credenziali a lungo termine. Quando possibile, utilizza i ruoli IAM che forniscono credenziali temporanee. Per i dettagli, consulta la sezione [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

16. (Facoltativo) Puoi consentire ad altri di Account AWS utilizzare questa chiave KMS per operazioni crittografiche. A tale scopo, nella Account AWS sezione Altro in fondo alla pagina, scegli Aggiungi un altro account Account AWS e inserisci l' Account AWS ID di un account esterno. Per aggiungere più account esterni, ripetere questo passaggio.

 Note

Gli amministratori dell'altro Account AWS devono inoltre consentire l'accesso alla chiave KMS creando policy IAM per i propri utenti. Per ulteriori informazioni, consulta [Autorizzazione per gli utenti in altri account di utilizzare una chiave KMS.](#)

17. Seleziona Next (Successivo).
18. Esaminare le principali impostazioni scelte. È comunque possibile tornare indietro e modificare tutte le impostazioni.
19. Al termine, scegli Crea filtro.

Metodo 2: inizia dalle chiavi gestite dal cliente

Questa procedura è la stessa di quella per creare una chiave di crittografia simmetrica con materiale chiave. AWS KMS Tuttavia, in questa procedura puoi specificare l'ID dell'archivio delle chiavi personalizzate dell'archivio delle chiavi esterne e l'ID chiave della chiave esterna. Puoi inoltre specificare i [valori delle proprietà richiesti](#) per una chiave KMS in un archivio delle chiavi esterne, come le specifiche e l'utilizzo della chiave.

1. [Accedi a Console di gestione AWS e apri la console AWS Key Management Service \(AWS KMS\) su https://console.aws.amazon.com /kms.](https://console.aws.amazon.com/kms)
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
4. Scegli Crea chiave.
5. Scegliere Symmetric (Simmetrica).
6. In Key usage (Utilizzo della chiave), l'opzione Encrypt and decrypt (Crittografa e decrittografa) è selezionata per default. Non modificarla.
7. Scegliere Advanced options (Opzioni avanzate).
8. In Key material origin (Origine del materiale della chiave), scegli External key store (Archivio delle chiavi esterne).
9. Conferma che intendi creare una chiave KMS nell'archivio delle chiavi esterne specificato.
10. Scegli Next (Successivo).
11. Scegli la riga che rappresenta l'archivio delle chiavi esterne per la nuova chiave KMS.

Non puoi scegliere un archivio delle chiavi esterne disconnesso. Per connettere un archivio delle chiavi disconnesso, scegli il nome dell'archivio, quindi in Key store actions (Operazioni per l'archivio delle chiavi), scegli Connect (Connetti). Per informazioni dettagliate, vedi [Utilizzo della console AWS KMS](#).

12. Inserisci l'ID chiave di una [chiave esterna](#) nel [gestore delle chiavi esterne](#). Questa chiave esterna deve [soddisfare i requisiti](#) per l'uso con una chiave KMS. Non puoi modificare questo valore dopo la creazione della chiave.

Se la chiave esterna è multipla IDs, inserisci l'ID della chiave utilizzato dal proxy dell'archivio chiavi esterno per identificare la chiave esterna.

13. Scegli Next (Successivo).

Il resto di questa procedura è uguale alla [creazione di una chiave KMS standard](#).

14. Digita un alias ed eventualmente una descrizione per la chiave KMS.
15. (Facoltativo). Nella pagina Add Tags (Aggiungi tag), aggiungi i tag che identificano o categorizzano la chiave KMS.

Quando aggiungi tag alle tue AWS risorse, AWS genera un rapporto sull'allocazione dei costi con utilizzo e costi aggregati per tag. I tag possono essere utilizzati anche per controllare l'accesso a una chiave KMS. Per informazioni sull'assegnazione di tag delle chiavi KMS, consulta [Tag in AWS KMS](#) e [ABAC per AWS KMS](#).

16. Scegli Next (Successivo).
17. Nella sezione amministratori delle chiavi, seleziona utenti IAM e ruoli IAM che possono gestire la chiave KMS. Per ulteriori informazioni, consulta [Consente agli amministratori delle chiavi di amministrare la chiave KMS](#).

 Note

Le policy IAM possono fornire ad altri ruoli e utenti IAM l'autorizzazione per utilizzare la chiave KMS.

18. (Facoltativo) Per impedire a questi amministratori delle chiavi di eliminare tale chiave KMS, deseleziona la casella di controllo Allow key administrators to delete this key. (Consenti agli amministratori delle chiavi di eliminare questa chiave).

L'eliminazione di una chiave KMS è un'operazione distruttiva e irreversibile che può rendere il testo criptato irrecuperabile. Non puoi ricreare una chiave KMS simmetrica in un archivio delle chiavi esterne, anche se disponi del materiale della chiave. L'eliminazione di una chiave KMS non ha alcun effetto sulla chiave esterna associata. Per informazioni sull'eliminazione di una chiave KMS da un archivio delle chiavi esterne, consulta [Eliminare un AWS KMS key](#).

19. Scegli Next (Successivo).
20. [Nella sezione Questo account, seleziona gli utenti e i ruoli IAM in grado di utilizzare la chiave KMS nelle operazioni crittografiche. Account AWS](#) Per ulteriori informazioni, consulta [Consente agli utenti della chiave di utilizzare la chiave KMS](#).

 Note

Le policy IAM possono fornire ad altri ruoli e utenti IAM l'autorizzazione per utilizzare la chiave KMS.

21. (Facoltativo) Puoi consentire ad altri di Account AWS utilizzare questa chiave KMS per operazioni crittografiche. A tale scopo, nella Account AWS sezione Altro in fondo alla pagina, scegli Aggiungi un altro account Account AWS e inserisci l' Account AWS ID di un account esterno. Per aggiungere più account esterni, ripetere questo passaggio.

 Note

Gli amministratori dell'altro Account AWS devono inoltre consentire l'accesso alla chiave KMS creando policy IAM per i propri utenti. Per ulteriori informazioni, consulta [Autorizzazione per gli utenti in altri account di utilizzare una chiave KMS](#).

22. Seleziona Next (Successivo).
23. Esaminare le principali impostazioni scelte. È comunque possibile tornare indietro e modificare tutte le impostazioni.
24. Al termine, scegli Crea filtro.

Quando la procedura riesce, la visualizzazione mostra la nuova chiave KMS nell'archivio delle chiavi esterne che hai scelto. Quando scegli il nome o l'alias della nuova chiave KMS, la scheda Cryptographic configuration (Configurazione crittografica) nella relativa pagina dei dettagli visualizza l'origine della chiave KMS (External key store [Archivio delle chiavi esterne]), il nome, l'ID e il tipo dell'archivio delle chiavi personalizzate, nonché l'ID, l'utilizzo della chiave e lo stato della chiave esterna. Se la procedura ha esito negativo, viene visualizzato un messaggio di errore che descrive l'errore. Per , consulta [Risoluzione dei problemi relativi all'archivio delle chiavi esterne](#).

 Tip

Per agevolare l'identificazione delle chiavi KMS in un archivio delle chiavi personalizzate, nella pagina Customer managed keys (Chiavi gestite dal cliente), aggiungi il campo Origin (Origine) e la colonna Custom key store ID (ID dell'archivio chiavi personalizzate) alla visualizzazione. Per modificare i campi della tabella, scegli l'icona a forma di ingranaggio

nell'angolo in alto a destra della pagina. Per informazioni dettagliate, vedi [Personalizza la visualizzazione della console](#).

Utilizzo dell'API AWS KMS

Per creare una nuova chiave KMS in un archivio di chiavi esterno, usa l'[CreateKey](#) operazione. I parametri seguenti sono obbligatori:

- Il valore `Origin` deve essere `EXTERNAL_KEY_STORE`.
- Il parametro `CustomKeyStoreId` identifica l'archivio delle chiavi esterne. Il valore di [ConnectionState](#) per l'archivio delle chiavi esterne specificato deve essere `CONNECTED`. Per trovare `CustomKeyStoreId` e `ConnectionState`, usa l'operazione `DescribeCustomKeyStores`.
- Il parametro `XksKeyId` identifica la chiave esterna. Tale chiave deve [soddisfare i requisiti](#) per l'associazione a una chiave KMS.

Puoi inoltre utilizzare uno qualsiasi dei parametri facoltativi dell'operazione `CreateKey`, ad esempio `Policy` o i parametri [Tags](#) (Tag).

Note

Non includere informazioni riservate o sensibili nei campi `Description` o `Tags`. Questi campi possono apparire in testo semplice nei CloudTrail log e in altri output.

Gli esempi in questa sezione utilizzano [AWS Command Line Interface \(AWS CLI\)](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

Questo comando di esempio utilizza l'[CreateKey](#) operazione per creare una chiave KMS in un archivio di chiavi esterno. La risposta include le proprietà delle chiavi KMS, l'ID dell'archivio delle chiavi esterne e l'ID, l'utilizzo e lo stato della chiave esterna.

Prima di eseguire questo comando, sostituisci l'ID store chiavi personalizzate di esempio con un ID valido.

```
$ aws kms create-key --origin EXTERNAL_KEY_STORE --custom-key-store-id cks-1234567890abcdef0 --xks-key-id bb8562717f809024
```

```
{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2022-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "CustomKeyStoreId": "cks-1234567890abcdef0",
    "Description": "",
    "Enabled": true,
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "EXTERNAL_KEY_STORE",
    "XksKeyConfiguration": {
      "Id": "bb8562717f809024"
    }
  }
}
```

Identifica e visualizza le chiavi

Puoi utilizzare [Console di gestione AWS](#) o l'[API AWS Key Management Service \(AWS KMS\)](#) per visualizzare AWS KMS keys in ogni account e regione, comprese le chiavi KMS che gestisci e le chiavi KMS gestite da AWS.

Argomenti

- [Trova l'ID della chiave e l'ARN della chiave](#)
- [Accedi ed elenca i dettagli chiave KMS](#)
- [Identifica diversi tipi di chiave](#)
- [Personalizza la visualizzazione della console](#)
- [Trova le chiavi KMS e il materiale chiave in un archivio di AWS CloudHSM chiavi](#)

Trova l'ID della chiave e l'ARN della chiave

Per identificare un AWS KMS key, puoi utilizzare l'[ID della chiave](#) o l'Amazon Resource Name ([chiave ARN](#)). Nelle [operazioni di crittografia](#), puoi anche utilizzare il [nome alias](#) o l'[ARN dell'alias](#).

Puoi utilizzare la [AWS KMS console](#) o l'[ListKeys](#) operazione per identificare l'ID chiave e l'ARN della chiave di ogni chiave KMS nel tuo account e nella tua regione.

Per informazioni dettagliate sugli identificatori di chiave KMS supportati da, consulta [AWS KMS Identificatori chiave \(\) KeyId](#). Per informazioni sulla ricerca di un nome alias e di un alias ARN, consulta [Trova il nome dell'alias e l'alias ARN per una chiave KMS](#).

Utilizzo della console AWS KMS

1. Apri la AWS KMS console in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Per visualizzare le chiavi nell'account creato e gestito dall'utente, nel riquadro di navigazione, seleziona Chiavi gestite dal cliente. Per visualizzare le chiavi dell'account che AWS crea e gestisce per te, nel riquadro di navigazione, scegli chiavi gestite AWS.
4. Per trovare l'[ID chiave](#) per una chiave KMS, esamina la riga che inizia con l'alias della chiave KMS.

La colonna Key ID (ID chiave) viene visualizzata nelle tabelle per impostazione predefinita. Se la colonna ID chiave non viene visualizzata nella tabella, utilizzare la procedura descritta in [the section called “Personalizza la visualizzazione della console”](#) per ripristinarla. È inoltre possibile visualizzare l'ID chiave di una chiave KMS nella relativa pagina dei dettagli.

Customer managed keys					Key actions ▼	Create key
					< 1 >	⚙
☐	Aliases ▲	Key ID ▼	Status	Creation date		
☐	key-test	1234abcd-12ab-34cd-56ef-1234567890ab	Enabled	Oct 19, 2018 12:43 PDT		

5. Per trovare l'Amazon Resource Name (ARN) della chiave KMS, scegli l'alias o l'ID chiave. L'[ARN della chiave](#) viene visualizzato nella sezione General Configuration (Configurazione generale).

General configuration		
Aliases key-test	Status Enabled	ARN arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
Description -	Creation date Nov 06, 2018 15:11 PST	

Utilizzo dell' AWS KMS API

Per trovare l'[ID della chiave](#) e l'[ARN](#) della chiave di un AWS KMS key, utilizzare l'[ListKeys](#) operazione.

L'[ListKeys](#) operazione restituisce l'ID della chiave e l'Amazon Resource Name (ARN) di tutte le chiavi KMS nell'account e nella regione del chiamante.

Ad esempio, questa chiamata all'operazione `ListKeys` restituisce l'ID e l'ARN di ciascuna chiave KMS presente in questo account fittizio. Per esempi in più linguaggi di programmazione, consulta [Utilizzo ListKeys con un AWS SDK o una CLI](#).

```
$ aws kms list-keys
{
  "Keys": [
    {
```

```
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyArn": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  {
    "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "KeyArn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
  }
]
```

Accedi ed elenca i dettagli chiave KMS

Puoi utilizzare la [AWS KMS console](#) o l'[DescribeKey](#) operazione per accedere ed elencare informazioni dettagliate sulle chiavi KMS nell'account e nella regione.

Le seguenti procedure mostrano come accedere ai dettagli delle chiavi KMS, come l'ID della chiave, le specifiche della chiave, l'utilizzo delle chiavi e altro ancora.

Utilizzo della console AWS KMS

La pagina dei dettagli di ogni chiave KMS visualizza le proprietà della chiave KMS. Differisce leggermente per i diversi tipi di chiavi KMS.

Per visualizzare informazioni dettagliate su una chiave KMS, nella pagina Chiavi gestite da AWS o Chiavi gestite dal cliente, scegli l'alias o l'ID della chiave KMS.

La pagina dei dettagli per una chiave KMS include una Configurazione generale in cui vengono visualizzate le proprietà di base della chiave KMS. Include anche schede in cui è possibile visualizzare e modificare le proprietà della chiave KMS, come la politica delle chiavi, la configurazione crittografica, i tag, il materiale chiave e le rotazioni (per le chiavi KMS che supportano la rotazione automatica o su richiesta), Regionalità (per le chiavi multiregione) e la chiave pubblica (per le chiavi KMS asimmetriche).

Note

[La AWS KMS console mostra le chiavi KMS che sei autorizzato a visualizzare nel tuo account e nella tua regione.](#) Le chiavi KMS di altri utenti Account AWS non vengono visualizzate nella

console, anche se sei autorizzato a visualizzarle, gestirle e utilizzarle. Per visualizzare le chiavi KMS in altri account, usa l'[DescribeKey](#) operazione.

Per passare alla pagina dei dettagli delle chiavi per una chiave KMS.

1. Accedi Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Per visualizzare le chiavi nell'account creato e gestito dall'utente, nel riquadro di navigazione, seleziona Chiavi gestite dal cliente. Per visualizzare le chiavi dell'account che AWS crea e gestisce per te, nel riquadro di navigazione, scegli chiavi gestite.AWS
4. Per aprire la pagina dei dettagli delle chiavi, nella tabella delle chiavi, scegli l'ID chiave o l'alias della chiave KMS.

Se la chiave KMS dispone di più alias, viene visualizzato un riepilogo degli alias (+n più) viene visualizzato accanto al nome di uno degli alias. La scelta del riepilogo alias consente di accedere direttamente alla sezione Alias nella pagina dei dettagli delle chiavi.

 [KMS](#) > [Customer managed keys](#) > Key ID: 6f84178a-007d-4501-8229-abc6af9e609c  

6f84178a-007d-4501-8229-abc6af9e609c Key actions ▼ Edit

General configuration

Alias kms-test	Status Enabled	Creation date May 26, 2025 01:49 PDT
ARN  arn:aws:kms:us-east-1:849743133991:key/6f84178a-007d-4501-8229-abc6af9e609c	Description -	Regionality Single Region
Current key material ID f2e38520f5b8424d0ccd2dc64530a25e830926738224b0cb97d8e62302549f81		

Key policy

Cryptographic configuration

Tags

Key material and rotations

Aliases

Cryptographic configuration

Key Type Symmetric	Origin AWS KMS	Key Spec ⓘ SYMMETRIC_DEFAULT	Key Usage Encrypt and decrypt

Nell'elenco seguente vengono descritti i campi nella visualizzazione dettagliata, incluso il campo nelle schede. Alcuni di questi campi sono disponibili anche come colonne nella visualizzazione della tabella.

Alias

Dove: scheda degli Alias

Specifica un nome intuitivo per la chiave KMS. Puoi utilizzare un alias per identificare la chiave KMS nella console e in alcune altre. AWS KMS APIs Per informazioni dettagliate, vedi [Alias in AWS KMS](#).

La scheda Alias mostra tutti gli alias associati alla chiave KMS nella regione and. Account AWS ARN

Dove: sezione Configurazione generale

Il nome della risorsa Amazon (ARN) della chiave KMS. Questo valore identifica singolarmente la chiave KMS. Puoi utilizzarlo per identificare la chiave KMS nelle operazioni API AWS KMS .

Stato connessione

Indica se un [archivio delle chiavi personalizzate](#) è collegato al relativo archivio del materiale della chiave. Questo campo viene visualizzato solo quando la chiave KMS viene creata in un archivio delle chiavi personalizzate.

Per informazioni sui valori in questo campo, consulta l'AWS KMS API [ConnectionState](#)Reference.

Data di creazione

Dove: sezione Configurazione generale

La data e l'ora di creazione della chiave KMS. Questo valore viene visualizzato nell'ora locale per il dispositivo. Il fuso orario non dipende dalla regione.

A differenza della Expiration (Scadenza), la creazione si riferisce solo alla chiave KMS, non al materiale della chiave.

ID cluster CloudHSM

Dove: scheda Configurazione crittografica

L'ID del AWS CloudHSM cluster che contiene il materiale chiave per la chiave KMS. Questo campo viene visualizzato solo quando la chiave KMS viene creata in un [archivio delle chiavi personalizzate](#).

Se scegli l'ID del cluster CloudHSM, viene aperta la pagina Cluster nella console. AWS CloudHSM

Materiale chiave attuale

Dove: sezione Configurazione generale

Le chiavi di crittografia simmetriche con AWS_KMS origine supportano sia la rotazione automatica che quella su richiesta. Le chiavi di crittografia simmetriche con EXTERNAL origine supportano la rotazione su richiesta. Queste chiavi possono avere più materiali chiave associati alla chiave. Il materiale chiave ruotato più di recente può essere utilizzato sia per la crittografia che per la decrittografia. Questo materiale chiave è identificato come il materiale chiave attuale. Gli altri materiali chiave possono essere utilizzati solo per la decrittografia. La rotazione automatica o su richiesta di una chiave KMS modifica il materiale chiave corrente.

ID dello store delle chiavi personalizzate

Dove: scheda Configurazione crittografica

L'ID [dell'archivio delle chiavi personalizzate](#) che contiene la chiave KMS. Questo campo viene visualizzato solo quando la chiave KMS viene creata in un archivio delle chiavi personalizzate.

Se scegli l'ID dell'archivio chiavi personalizzato, viene aperta la pagina Custom key store nella AWS KMS console.

Il nome dello store delle chiavi personalizzate

Dove: scheda Configurazione crittografica

Il nome [dell'archivio delle chiavi personalizzate](#) che contiene la chiave KMS. Questo campo viene visualizzato solo quando la chiave KMS viene creata in un archivio delle chiavi personalizzate.

Tipo di archivio delle chiavi personalizzate

Dove: scheda Configurazione crittografica

Indica se l'archivio delle chiavi personalizzate è un [archivio delle chiavi di AWS CloudHSM](#) o un [archivio delle chiavi esterne](#). Questo campo viene visualizzato solo quando la chiave KMS viene creata in un [archivio delle chiavi personalizzate](#).

Description

Dove: sezione Configurazione generale

Una breve descrizione facoltativa della chiave KMS che puoi scrivere e modificare. Per aggiungere o aggiornare la descrizione di una chiave gestita dal cliente, sopra Configurazione generale seleziona Modifica.

Algoritmi di crittografia

Dove: scheda Configurazione crittografica

Elenca gli algoritmi di crittografia che possono essere utilizzati con la chiave KMS in AWS KMS. Questo campo viene visualizzato solo quando Key type (Tipo di chiave) è Asymmetric (Asimmetrico) e Key usage (Utilizzo della chiave) è Encrypt and decrypt (Crittografa e decripta). Per informazioni sugli algoritmi di crittografia AWS KMS supportati, consulta [Specifica della chiave SYMMETRIC_DEFAULT](#) e [Specifiche della chiave RSA per la crittografia e la decrittografia](#).

Data di scadenza

Dove: scheda Materiale della chiave

Data e ora in cui scade il materiale della chiave per la chiave KMS. Questo campo viene visualizzato solo per le chiavi KMS con [materiale della chiave importato](#), ovvero quando Origin (Origine) è External (Esterna) e la chiave KMS ha materiale della chiave in scadenza. Alle chiavi di crittografia simmetriche possono essere associati più materiali chiave. Per tali chiavi, questo campo indica la prima data e ora in cui scade uno dei materiali chiave associati.

ID chiave esterna

Dove: scheda Configurazione crittografica

L'ID della [chiave esterna](#) associata a una chiave KMS in un [archivio delle chiavi esterne](#). Questo campo viene visualizzato solo per le chiavi KMS in un archivio delle chiavi esterne.

Stato della chiave esterna

Dove: scheda Configurazione crittografica

Lo stato più recente riportato dal [proxy dell'archivio delle chiavi esterne](#) per la [chiave esterna](#) associata alla chiave KMS. Questo campo viene visualizzato solo per le chiavi KMS in un archivio delle chiavi esterne.

Utilizzo della chiave esterna

Dove: scheda Configurazione crittografica

Le operazioni di crittografia abilitate sulla [chiave esterna](#) associata alla chiave KMS. Questo campo viene visualizzato solo per le chiavi KMS in un archivio delle chiavi esterne.

Policy della chiave

Dove: scheda Policy delle chiavi

Controlla l'accesso alla chiave KMS e alle [policy IAM](#) e alle [concessioni](#). Ogni chiave KMS ha una policy delle chiavi. È l'unico elemento di autorizzazione obbligatorio. Per modificare la policy delle chiavi di una chiave gestita dal cliente, nella scheda Policy delle chiavi, scegli Modifica. Per informazioni dettagliate, vedi [the section called "Policy delle chiavi"](#).

Materiale chiave e rotazioni

Dove: scheda Materiale chiave e rotazioni

Questa scheda viene visualizzata solo per le chiavi di crittografia simmetriche con AWS_KMS origine (che supportano sia la rotazione automatica che su richiesta) e per le chiavi di crittografia simmetriche a regione singola con EXTERNAL origine (che supportano la rotazione su richiesta).

La scheda è composta da tre pannelli:

Rotazione automatica: abilita e disabilita la [rotazione automatica](#) del materiale chiave in una chiave [KMS gestita dal cliente](#). Per modificare lo stato di rotazione delle chiavi di una [chiave gestita dal cliente](#), utilizza la casella di controllo. Non è possibile abilitare o disabilitare la rotazione del materiale della chiave in una [Chiave gestita da AWS](#). Le Chiavi gestite da AWS vengono ruotate automaticamente ogni anno.

Rotazione su richiesta: avvia una rotazione [su richiesta](#) del materiale chiave in una chiave gestita [dal cliente](#). Per le chiavi importate, è necessario che sia già presente un materiale chiave importato PENDING_ROTATION affinché l'opzione Ruota ora sia disponibile.

Materiali chiave: elenca tutti i materiali chiave associati alla chiave KMS. Ogni materiale chiave ha un identificatore univoco e la relativa riga mostra informazioni aggiuntive sul materiale chiave, come la data di rotazione in cui il materiale chiave è diventato disponibile per l'uso in KMS. Per le chiavi importate, ogni riga ha anche un menu Azioni che può essere utilizzato per eliminare un materiale chiave specifico o reimportarlo nella chiave KMS.

Specifica della chiave

Dove: scheda Configurazione crittografica

Il tipo di materiale chiave nella chiave KMS. AWS KMS supporta chiavi KMS con crittografia simmetrica (SYMMETRIC_DEFAULT), chiavi KMS HMAC di diverse lunghezze, chiavi KMS per

chiavi RSA di diverse lunghezze e chiavi a curva ellittica con curve diverse. Per informazioni dettagliate, vedi [Key spec](#).

Tipo di chiavi

Dove: scheda Configurazione crittografica

Indica se la chiave KMS è Simmetrica o Asimmetrica.

Utilizzo delle chiavi

Dove: scheda Configurazione crittografica

Indica se una chiave KMS può essere utilizzata per le operazioni Encrypt and decrypt (Crittografa e decrittografa), Sign and verify (Firma e verifica) o Generate and verify MAC (Genera e verifica MAC). Per informazioni dettagliate, vedi [Key usage](#).

Origin

Dove: scheda Configurazione crittografica

L'origine del materiale della chiave per la chiave KMS. I valori validi sono:

- AWS KMS per il materiale della chiave generato da AWS KMS
- AWS CloudHSM per le chiavi KMS in un [archivio delle chiavi di AWS CloudHSM](#)
- Esterno per il [materiale della chiave importato](#) (BYOK)
- Archivio delle chiavi esterne per le chiavi KMS in un [archivio delle chiavi esterne](#)

Algoritmi MAC

Dove: scheda Configurazione crittografica

Elenca gli algoritmi MAC che possono essere utilizzati con la chiave KMS HMAC in AWS KMS. Questo campo viene visualizzato solo quando il valore Key spec (Specifica della chiave) è una specifica di chiave HMAC (HMAC_*). Per informazioni sugli algoritmi MAC supportati da AWS KMS, consulta la sezione [Specifiche della chiave per le chiavi KMS HMAC](#).

Chiave primaria

Dove: scheda Regionalità

Indica che questa chiave KMS è una [chiave primaria multi-regione](#). Gli utenti autorizzati possono utilizzare questa sezione per [cambiare la chiave primaria](#) con una diversa chiave multi-regione correlata. Questo campo viene visualizzato solo quando la chiave KMS è una chiave primaria multi-regione.

Chiavi pubbliche

Dove: scheda Chiave pubblica

Visualizza la chiave pubblica di una chiave KMS asimmetrica. Gli utenti autorizzati possono utilizzare questa scheda per [copiare e scaricare la chiave pubblica](#).

Regionalità

Dove: sezione Configurazione generale e schede Regionalità

Indica se una chiave KMS è una chiave di singola regione, una [chiave primaria multi-regione](#), o una [chiave di replica in multi-regione](#). Questo campo viene visualizzato solo quando la chiave KMS è una chiave multi-regione.

Chiavi multi-regione correlate

Dove: scheda Regionalità

Visualizza tutte le relative [chiavi primarie e di replica multi-regione](#), ad eccezione della chiave KMS corrente. Questo campo viene visualizzato solo quando la chiave KMS è una chiave multi-regione.

Nella sezione chiavi multi-regione correlate di una chiave primaria, gli utenti autorizzati possono [creare nuove chiavi di replica](#).

Chiave di replica

Dove: scheda Regionalità

Indica che questa chiave KMS è una [chiave di replica multi-regione](#). Questo campo viene visualizzato solo quando la chiave KMS è una chiave di replica multi-regione.

Algoritmi di firma

Dove: scheda Configurazione crittografica

Elenca gli algoritmi di firma che possono essere utilizzati con la chiave KMS in AWS KMS. Questo campo viene visualizzato solo quando Key type (Tipo di chiave) è Asymmetric (Asimmetrico) e Key usage (Utilizzo della chiave) è Sign and verify (Firma e verifica). Per informazioni sugli algoritmi di firma AWS KMS supportati, vedere [Specifiche della chiave RSA per la firma e la verifica](#) e [Specifiche della chiave basata su curva ellittica](#).

Status

Dove: sezione Configurazione generale

Lo stato della chiave KMS. È possibile utilizzare la chiave KMS nelle [operazioni di crittografia](#) solo quando lo stato è Enabled (Abilitato). Per una descrizione dettagliata di ogni stato di chiave KMS e il relativo effetto sulle operazioni che è possibile eseguire sulla chiave KMS, consulta [Stati chiave delle AWS KMS chiavi](#).

Tag

Dove: scheda Tag

Coppie chiave-valore opzionali che descrivono la chiave KMS. Per aggiungere o modificare i tag per una chiave KMS, nella scheda Tag scegli Modifica.

Quando aggiungi tag alle tue AWS risorse, AWS genera un rapporto sull'allocazione dei costi con utilizzo e costi aggregati per tag. I tag possono essere utilizzati anche per controllare l'accesso a una chiave KMS. Per informazioni sul tagging delle chiavi KMS, consulta [Tag in AWS KMS](#) e [ABAC per AWS KMS](#).

Utilizzo dell'API AWS KMS

L'[DescribeKey](#) operazione restituisce dettagli sulla chiave KMS specificata. Per identificare la chiave KMS, utilizza [ID chiave](#), [ARN di chiave](#), [nome alias](#) o [alias ARN](#).

A differenza dell'[ListKeys](#) operazione, che visualizza solo le chiavi KMS nell'account e nella regione del chiamante, gli utenti autorizzati possono utilizzare l'[DescribeKey](#) operazione per ottenere dettagli sulle chiavi KMS in altri account.

Note

La risposta [DescribeKey](#) include sia membri `KeySpec` e `CustomerMasterKeySpec` con gli stessi valori. Il membro `CustomerMasterKeySpec` è obsoleto.

Ad esempio, questa chiamata a [DescribeKey](#) restituisce informazioni su una chiave KMS di crittografia simmetrica. I campi nella risposta variano in base alle [specifiche della AWS KMS key](#), allo [stato della chiave](#) e all'[origine del materiale della chiave](#). Per esempi in più linguaggi di programmazione, consulta [Utilizzo DescribeKey con un AWS SDK o una CLI](#).

```
$ aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab  
  
{
```

```

    "KeyMetadata": {
      "Origin": "AWS_KMS",
      "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "Description": "",
      "KeyManager": "CUSTOMER",
      "Enabled": true,
      "KeySpec": "SYMMETRIC_DEFAULT",
      "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
      "KeyUsage": "ENCRYPT_DECRYPT",
      "KeyState": "Enabled",
      "CreationDate": 1499988169.234,
      "MultiRegion": false,
      "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "AWSAccountId": "111122223333",
      "EncryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
      ],
      "CurrentKeyMaterialId":
"123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    }
  }
}

```

Questo esempio chiama l'operazione `DescribeKey` su una chiave KMS asimmetrica utilizzata per la firma e la verifica. La risposta include gli algoritmi di firma supportati da AWS KMS per questa chiave KMS.

```

$ aws kms describe-key --key-id 0987dcba-09fe-87dc-65ba-ab0987654321

{
  "KeyMetadata": {
    "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "Origin": "AWS_KMS",
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
    "KeyState": "Enabled",
    "KeyUsage": "SIGN_VERIFY",
    "CreationDate": 1569973196.214,
    "Description": "",
    "KeySpec": "ECC_NIST_P521",
    "CustomerMasterKeySpec": "ECC_NIST_P521",
    "AWSAccountId": "111122223333",
    "Enabled": true,

```

```
    "MultiRegion": false,
    "KeyManager": "CUSTOMER",
    "SigningAlgorithms": [
        "ECDSA_SHA_512"
    ]
}
```

Identifica diversi tipi di chiave

I seguenti argomenti spiegano come identificare i diversi tipi di chiave nella AWS KMS console e nelle [DescribeKey](#)risposte.

Per informazioni sulla navigazione nella scheda Configurazione crittografica nella pagina dei dettagli di una chiave KMS, consulta. [the section called “Accedi ed elenca i dettagli chiave KMS”](#)

Argomenti

- [Identifica le chiavi KMS asimmetriche](#)
- [Identifica le chiavi HMAC KMS](#)
- [Identifica le chiavi KMS multiregionali](#)
- [Identifica le chiavi KMS con materiale chiave importato](#)
- [Identifica le chiavi KMS negli AWS CloudHSM archivi delle chiavi](#)
- [Identifica le chiavi KMS negli archivi di chiavi esterni](#)

Identifica le chiavi KMS asimmetriche

Nella AWS KMS console

La colonna Tipo di chiave della tabella Customer managed keys mostra se ogni chiave KMS è simmetrica o asimmetrica. Puoi filtrare la tabella in base al valore del tipo di chiave per visualizzare solo le chiavi KMS asimmetriche. Per ulteriori informazioni, consulta [the section called “Ordina e filtra le tue chiavi KMS”](#).

La scheda Configurazione crittografica nella pagina dei dettagli di una chiave KMS mostra il Tipo di chiave, che indica se la chiave è simmetrica o asimmetrica. Visualizza anche l'utilizzo delle chiavi, che indica se la chiave KMS asimmetrica viene utilizzata per la crittografia e la decrittografia, la firma e la verifica o la derivazione di segreti condivisi.

Nelle risposte DescribeKey

Quando si richiama l'DescribeKey operazione su una chiave KMS asimmetrica, la risposta include i KeyUsage valori KeySpec and, che possono essere utilizzati per determinare se una chiave KMS è simmetrica o asimmetrica.

Se il KeySpec valore è, la chiave è SYMMETRIC_DEFAULT una chiave KMS di crittografia simmetrica. Per i dettagli sulle specifiche delle chiavi asimmetriche, vedere. [Riferimento alle specifiche chiave](#)

Se il KeyUsage valore è SIGN_VERIFY o KEY_AGREEMENT, la chiave è una chiave KMS asimmetrica.

L'DescribeKey operazione restituisce anche i seguenti dettagli per le chiavi KMS asimmetriche.

- Per le chiavi KMS asimmetriche con un KeyUsage valore di, l'operazione restituisce il ENCRYPT_DECRYPT, che elenca gli algoritmi di crittografia EncryptionAlgorithms validi per la chiave.
- Per le chiavi KMS asimmetriche con un KeyUsage valore di, l'operazione restituisce il SIGN_VERIFY, che elenca gli algoritmi di firma validi per SigningAlgorithms la chiave.
- Per le chiavi KMS asimmetriche con un KeyUsage valore di, l'operazione restituisce il KEY_AGREEMENT, che elenca gli algoritmi di accordo chiave KeyAgreementAlgorithms validi per la chiave.

Per ulteriori informazioni sulle chiavi KMS asimmetriche, vedere. [the section called “Chiavi asimmetriche”](#)

Identifica le chiavi HMAC KMS

Nella console AWS KMS

Le chiavi HMAC KMS sono incluse nella tabella delle chiavi gestite dai clienti, ma non è possibile ordinare o filtrare questa tabella in base alle specifiche della chiave o ai valori di utilizzo delle chiavi che identificano le chiavi HMAC. Per semplificare la ricerca delle chiavi HMAC, puoi assegnare loro un alias o un tag distintivo. In questo modo, potrai ordinare o filtrare le chiavi in base all'alias o al tag.

La scheda Configurazione crittografica nella pagina dei dettagli di una chiave KMS visualizza il tipo di chiave, che indica se la chiave è simmetrica o asimmetrica. Le chiavi KMS HMAC sono

simmetriche. La scheda Configurazione crittografica mostra anche l'utilizzo delle chiavi. Per le chiavi HMAC KMS, il valore di utilizzo della chiave è sempre Genera e verifica MAC.

Nelle risposte DescribeKey

Quando si richiama l'DescribeKey operazione su una chiave KMS HMAC, la risposta include i valori KeySpec e KeyUsage. Per le chiavi HMAC KMS, il valore di utilizzo della chiave è sempre GENERATE_VERIFY_MAC e il valore della specifica chiave inizia sempre con. HMAC_

Per ulteriori informazioni sulle chiavi HMAC KMS, vedere. [the section called “Chiavi HMAC”](#)

Identifica le chiavi KMS multiregionali

Nella console AWS KMS

La tabella delle chiavi gestite dal cliente mostra solo le chiavi KMS nella regione selezionata. È possibile visualizzare le chiavi primarie e di replica multiregione nella Regione selezionata. Per cambiare la AWS regione, usa il selettore della regione nell'angolo in alto a destra della console.

Per semplificare l'identificazione delle chiavi multiregionali nella tabella Chiavi gestite dal cliente, aggiungi la colonna Regionalità alla tabella. Per assistenza, consulta [the section called “Personalizza le tue tabelle chiave KMS”](#).

La pagina di dettaglio per le chiavi KMS multiregionali include una scheda Regionalità. La tab Regionalità per una chiave primaria include i pulsanti Modifica Regione primaria e Crea nuove chiavi di replica. (La tab Regionalità per una chiave di replica non dispone di alcun pulsante.) La sezione Chiavi multiregione correlate elenca tutte le chiavi multiregione correlate a quella corrente. Se la chiave corrente è una chiave di replica, l'elenco include la chiave primaria.

Se si sceglie una chiave multiregionale correlata dalla tabella Chiavi multiregionali correlate, la AWS KMS console passa alla regione della chiave selezionata e apre la pagina di dettaglio della chiave. Ad esempio, se si sceglie la chiave di replica nella sa-east-1 regione dalla sezione Esempi di chiavi multiregionali correlate di seguito, la AWS KMS console passa alla sa-east-1 regione per visualizzare la pagina di dettaglio di quella chiave di replica. È possibile eseguire questa operazione per visualizzare l'alias o la policy chiave per la chiave di replica. Per modificare di nuovo la Regione, utilizza il selettore Regione nell'angolo in alto a destra della pagina.

Nelle risposte DescribeKey

Per impostazione predefinita, le operazioni AWS KMS API sono regionali e restituiscono solo le risorse nella regione corrente o specificata. Tuttavia, quando si richiama

L'DescribeKeyoperazione su una chiave KMS multiregionale, la risposta include tutte le chiavi multiregionali correlate in altre AWS regioni nell'elemento. MultiRegionConfiguration

Per ulteriori informazioni sulle chiavi KMS multiregionali, consulta [the section called “Chiavi multi-regione”](#)

Identifica le chiavi KMS con materiale chiave importato

Nella console AWS KMS

Per facilitare l'identificazione delle chiavi KMS con materiale chiave importato nella tabella Customer managed keys, aggiungi la colonna Origin alla tabella. La colonna Origine semplifica l'identificazione delle chiavi KMS il cui valore della proprietà dell'origine è Esterna (Importa il materiale della chiave). Per assistenza, consulta [the section called “Personalizza le tue tabelle chiave KMS”](#).

La scheda Configurazione crittografica nella pagina dei dettagli di una chiave KMS mostra Origin, che identifica la fonte del materiale chiave per la chiave KMS. Per le chiavi KMS con materiale chiave importato, il valore di origine è sempre Esterno (materiale chiave di importazione).

La pagina dei dettagli include anche una scheda Materiale chiave che fornisce informazioni dettagliate sul materiale chiave importato. Le chiavi di crittografia simmetriche con EXTERNAL Origin supportano le rotazioni su richiesta e possono avere più materiali chiave associati. Per tali chiavi, la scheda è denominata Materiale chiave e rotazioni.

Nelle risposte DescribeKey

Quando si richiama l'DescribeKeyoperazione su una chiave KMS con materiale chiave importato, la risposta include i ValidTo valori OriginExpirationModel, e. Per le chiavi KMS con materiale chiave importato, il valore di origine è sempre. EXTERNAL Il ExpirationModel valore indica se il materiale chiave è impostato per scadere o meno e il ValidTo valore indica quando scadrà il materiale chiave. Quando a una chiave sono associati più materiali chiave, il ValidTo valore indica la prima data di scadenza per tutti i materiali chiave (ad eccezione di quello in sospeso di rotazione) ed ExpirationModel è impostato su DOES_NOT_EXPIRE solo se nessuno di questi materiali chiave è impostato per scadere. Per ulteriori informazioni, consulta [Impostazione di una data di scadenza \(facoltativo\)](#).

Per ulteriori informazioni sulle chiavi KMS con materiale chiave importato, vedere [the section called “Materiale della chiave importato”](#)

Identifica le chiavi KMS negli AWS CloudHSM archivi delle chiavi

Nella console AWS KMS

Per semplificare l'identificazione delle chiavi KMS negli archivi di AWS CloudHSM chiavi nella tabella Customer managed keys, aggiungi la colonna Origin alla tabella. La colonna Origin (Origine) semplifica l'identificazione delle chiavi KMS il cui valore della proprietà Origin è AWS CloudHSM. Per assistenza, consulta [the section called “Personalizza le tue tabelle chiave KMS”](#).

La scheda Configurazione crittografica nella pagina dei dettagli di una chiave KMS mostra Origin, che identifica la fonte del materiale chiave per la chiave KMS. Per le chiavi KMS negli archivi di chiavi, il AWS CloudHSM valore di origine è sempre. AWS CloudHSM

Per una chiave KMS in un archivio AWS CloudHSM chiavi, la scheda Configurazione crittografica include una sezione aggiuntiva, Archivio chiavi personalizzato, che fornisce informazioni sull'archivio chiavi e sul AWS CloudHSM cluster associati alla AWS CloudHSM chiave KMS.

Nelle risposte DescribeKey

Quando si richiama l'DescribeKey operazione su una chiave KMS in un archivio di AWS CloudHSM chiavi, la risposta include il Origin, che identifica la fonte del materiale chiave. Per le chiavi KMS in un AWS CloudHSM archivio di chiavi, il valore di origine è sempre. AWS_CLOUDHSM L'operazione restituisce anche i seguenti campi speciali per le chiavi KMS negli AWS CloudHSM archivi di chiavi:

- CloudHsmClusterId
- CustomKeyStoreId

Per ulteriori informazioni sugli archivi di AWS CloudHSM chiavi, vedere [the section called “AWS CloudHSM negozi chiave”](#).

Identifica le chiavi KMS negli archivi di chiavi esterni

Nella console AWS KMS

Per semplificare l'identificazione delle chiavi KMS negli archivi di chiavi esterni nella tabella Customer managed keys, aggiungi la colonna Origin alla tabella. La colonna Origin semplifica l'identificazione delle chiavi KMS con un valore della proprietà External key store origin. Per assistenza, consulta [the section called “Personalizza le tue tabelle chiave KMS”](#).

La scheda Configurazione crittografica nella pagina dei dettagli di una chiave KMS mostra l'origine, che identifica la fonte del materiale chiave per la chiave KMS. Per le chiavi KMS negli archivi di chiavi esterni, il valore di origine è sempre Archivio chiavi esterno.

Per una chiave KMS in un archivio di chiavi esterno, la scheda Configurazione crittografica include due sezioni aggiuntive, Archivio chiavi personalizzato e Chiave esterna. La tabella Custom key store fornisce informazioni sull'archivio chiavi esterno associato alla chiave KMS. La tabella delle chiavi esterne viene visualizzata nella AWS KMS console solo per le chiavi KMS negli archivi di chiavi esterni. Fornisce informazioni sulla chiave esterna associata alla chiave KMS. La [chiave esterna](#) è una chiave crittografica esterna AWS che funge da materiale chiave per la chiave KMS nell'archivio delle chiavi esterno. Quando si crittografa o si decrittografa con la chiave KMS, l'operazione viene eseguita dal [gestore delle chiavi esterne](#) utilizzando la chiave esterna specificata.

I seguenti valori vengono visualizzati nella sezione External key (Chiave esterna).

ID chiave esterna

L'identificatore della chiave esterna nel relativo gestore delle chiavi esterne. Si tratta del valore utilizzato dal proxy dell'archivio delle chiavi esterne per identificare la chiave esterna. Puoi specificare l'ID della chiave esterna in fase di creazione della chiave KMS e non puoi modificarla in seguito. Se il valore dell'ID della chiave esterna utilizzato per creare la chiave KMS cambia o diventa non valido, è necessario [pianificare l'eliminazione della chiave KMS e creare una nuova chiave KMS](#) con il valore ID della chiave esterna corretto.

Nelle risposte DescribeKey

Quando si richiama l'DescribeKeyoperazione su una chiave KMS in un archivio di chiavi esterno, la risposta include il `Origin`, che identifica l'origine del materiale chiave. Per le chiavi KMS in un AWS CloudHSM archivio di chiavi, il valore di origine è sempre `EXTERNAL_KEY_STORE`. L'operazione restituisce anche l'`CustomKeyStoreId`elemento, che identifica l'archivio di chiavi esterno associato alle chiavi KMS.

Per ulteriori informazioni sugli archivi di chiavi esterni, vedere. [the section called “Archivi delle chiavi esterne”](#)

Personalizza la visualizzazione della console

Puoi personalizzare la visualizzazione della AWS KMS console per facilitare la ricerca delle chiavi KMS. Personalizza le tabelle visualizzate nelle pagine Chiavi gestite da AWS e nelle pagine delle

chiavi gestite dai clienti per visualizzare le informazioni di cui hai più bisogno oppure ordina e filtra le chiavi KMS restituite nelle tabelle.

Argomenti

- [Ordina e filtra le tue chiavi KMS](#)
- [Personalizza le tue tabelle chiave KMS](#)

Ordina e filtra le tue chiavi KMS

Per semplificare la ricerca delle chiavi KMS nella console, puoi ordinare e filtrare le tabelle delle chiavi.

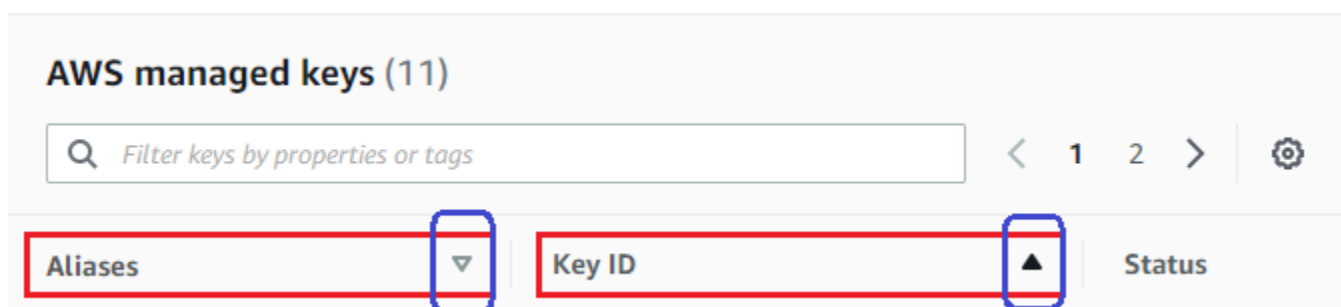
Ordina

Puoi ordinare le chiavi KMS in ordine crescente o decrescente in base ai valori di colonna. Questa caratteristica ordina tutte le chiavi KMS nella tabella, anche se non vengono visualizzate nella pagina della tabella corrente.

Le colonne ordinabili sono indicate da una freccia accanto al nome della colonna. Sulla pagina Chiavi gestite da AWS è possibile ordinare in base a Alias o ID chiave. Nella pagina Customer managed keys (Chiavi gestite dal cliente), è possibile ordinare per Alias, ID chiave o Key type (Tipo di chiave).

Per ordinare in ordine crescente, scegliere l'intestazione della colonna fino a quando la freccia non punta verso l'alto. Per ordinare in ordine decrescente, scegliere l'intestazione della colonna fino a quando la freccia non punta verso il basso. Puoi eseguire l'ordinamento in base a una colonna alla volta.

Ad esempio, puoi ordinare le chiavi KMS in ordine crescente per ID chiave, anziché alias, che è l'impostazione predefinita.



Quando si ordinano le chiavi KMS nella pagina Chiavi gestite dal cliente in ordine crescente per Tipo di chiave, tutte le chiavi asimmetriche vengono visualizzate prima di tutte quelle simmetriche.

Filtro

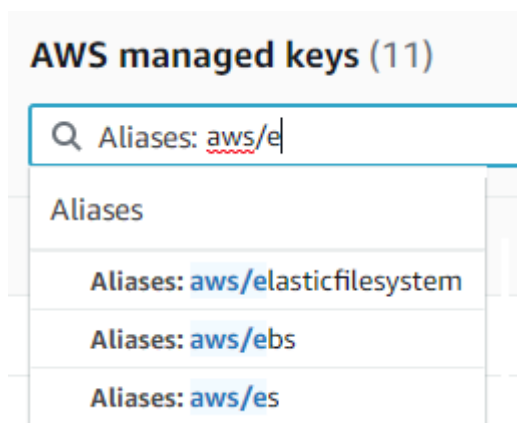
È possibile filtrare le chiavi KMS in base ai valori delle proprietà o ai tag. Il filtro si applica a tutte le chiavi KMS nella tabella, anche se non vengono visualizzate nella pagina della tabella corrente. Il filtro non fa distinzione tra maiuscole e minuscole.

Le proprietà filtrabili sono elencate nella casella filtro. Sulla pagina Chiavi gestite da AWS, è possibile filtrare in base all'alias e all'ID chiave. Nella pagina Chiavi gestite dal cliente, puoi filtrare per Alias, ID Chiave, Tipo di chiave e tag.

- Sulla pagina Chiavi gestite da AWS, è possibile filtrare in base all'alias e all'ID chiave.
- Sulla pagina Chiavi gestite dal cliente, è possibile filtrare in base ai tag o in base all'alias, all'ID chiave, al tipo di chiave o alle proprietà di regionalità.

Per filtrare in base al valore di una caratteristica, scegli il filtro, il nome della proprietà e scegli dall'elenco dei valori effettivi delle proprietà. Per filtrare in base a un tag, scegli il codice tag, quindi scegli dall'elenco dei valori effettivi dei tag. Dopo aver scelto una chiave di proprietà o di tag, puoi anche digitare tutto il valore della proprietà o solo parte di esso. Vedrai un'anteprima dei risultati prima di effettuare la tua scelta.

Ad esempio, per visualizzare le chiavi KMS con un nome alias contenente `aws/e`, scegli la casella del filtro, scegli Alias, digita `aws/e`, e premi `Enter` o `Return` per aggiungere il filtro.



Filtri della tabella delle chiavi KMS consigliati

Filtro per chiavi KMS asimmetriche

Per visualizzare solo le chiavi KMS asimmetriche nella pagina Chiavi gestite dal cliente) fare clic sulla casella del filtro, scegliere Tipo di chiave, quindi scegliere Tipo di chiave: Asimmetrica. L'opzione Asimmetrica viene visualizzata solo quando nella tabella sono presenti chiavi KMS asimmetriche.

Filtro per chiavi multiregionali

Per visualizzare solo le chiavi multi-regione, nella scheda Chiavi gestite dal cliente scegli la casella di filtro, scegli Regionalità e quindi Regionalità: Multiregione. L'opzione Multiregione viene visualizzata solo quando nella tabella sono presenti chiavi multi-regione.

Filtro per tag

Per visualizzare solo le chiavi KMS con un determinato tag, scegli la casella filtro, scegli la chiave di tag, quindi scegli tra i valori effettivi dei tag. È anche possibile digitare o tutto il valore del tag o solo parte di esso.

Nella tabella risultante vengono visualizzate tutte le chiavi KMS con il tag scelto. Tuttavia, il tag non viene visualizzato. Per visualizzare il tag, scegli l'alias o l'ID chiave della chiave KMS e nella pagina dei dettagli scegli la scheda Tag. Le schede appaiono nella sezione Configurazione generale.

Per questo filtro sono necessari sia la chiave di tag che il valore del tag. Non troverà le chiavi KMS digitando solo la chiave di tag o solo il suo valore. Per filtrare i tag in base alla chiave o al valore del tag, usa l'[ListResourceTags](#) operazione per ottenere le chiavi KMS con tag, quindi usa le funzionalità di filtro del tuo linguaggio di programmazione.

Filtra per testo

Per cercare del testo, nella casella filtro digita tutto o solo parte di un alias, di un ID chiave, di un tipo di chiave o di una chiave di tag. (Dopo aver selezionato la chiave di tag, puoi cercare un valore di tag). Vedrai un'anteprima dei risultati prima di effettuare la tua scelta.

Ad esempio, per visualizzare le chiavi KMS con test nelle chiavi di tag o nelle proprietà filtrabili, digitare test nella casella filtro. L'anteprima mostra le chiavi KMS che il filtro selezionerà. In questo caso, test appare solo nella proprietà Alias.

Personalizza le tue tabelle chiave KMS

Puoi personalizzare le tabelle visualizzate nelle pagine relative alle chiavi gestite dal cliente Chiavi gestite da AWS e quelle relative alle chiavi gestite dai Console di gestione AWS clienti in base alle tue esigenze. Puoi scegliere le colonne della tabella, il numero di colonne AWS KMS keys su ogni pagina (dimensione della pagina) e la disposizione del testo. La configurazione scelta viene salvata quando viene confermata e riapplicata ogni volta che si aprono le pagine.

Per personalizzare le tabelle delle chiavi KMS

1. Nella pagina Chiavi gestite da AWS o in Chiavi gestite dal cliente, scegli l'icona delle impostazioni



nell'angolo in alto a destra della pagina.

2. Nella pagina Preferences (Preferenze), scegliere le impostazioni preferite e quindi Confirm (Conferma).

Ti consigliamo di utilizzare l'impostazione Page size (Dimensione pagina) per aumentare il numero di chiavi KMS visualizzate su ogni pagina, soprattutto se usi un dispositivo facile da scorrere.

Le colonne di dati visualizzate possono variare a seconda della tabella, del ruolo lavorativo, e dei tipi di chiavi KMS nell'account e nella regione. Nella tabella seguente sono riportate alcune configurazioni suggerite. Per le descrizioni delle colonne, consulta [Utilizzo della console AWS KMS](#).

Configurazioni consigliate per le tabelle delle chiavi KMS

Puoi personalizzare le colonne visualizzate nella tabella delle chiavi KMS per visualizzare le informazioni necessarie sulle chiavi KMS.

Chiavi gestite da AWS

Per impostazione predefinita, la tabella Chiave gestita da AWS mostra le colonne Alias, ID chiave e Stato. Queste colonne sono ideali per la maggior parte dei casi d'uso.

Chiavi KMS di crittografia simmetrica

Se utilizzi soltanto chiavi KMS di crittografia simmetrica con materiale della chiave generato da AWS KMS, è probabile che le colonne Aliases (Alias), Key ID (ID chiave), Status (Stato) e Creation date (Data di creazione) siano quelle più utili.

Chiavi KMS asimmetriche

Se utilizzi chiavi KMS asimmetriche, oltre alle colonne Alias, ID chiave e Stato, si consiglia di aggiungere le colonne Tipo di chiave, Key spec (Specifiche della chiave) e Key usage (Utilizzo della chiave). Queste colonne mostrano se una chiave KMS è simmetrica o asimmetrica, il tipo di materiale della chiave e se la chiave KMS può essere utilizzata per la crittografia o la firma.

Chiavi KMS HMAC

Se utilizzi chiavi KMS HMAC, oltre alle colonne Aliases (Alias), Key ID (ID chiave) e Status (Stato), ti consigliamo di aggiungere le colonne Key spec (Specifiche della chiave) e Key usage (Utilizzo della chiave). Queste colonne mostrano se una chiave KMS è una chiave HMAC. Poiché non puoi ordinare le chiavi KMS in base alle specifiche o all'utilizzo delle chiavi, utilizza alias e tag per identificare le chiavi HMAC e quindi utilizza le [funzionalità di filtro](#) della AWS KMS console per filtrare in base a alias o tag.

Materiale della chiave importato

Se si dispone di chiavi KMS con [materiale della chiave importato](#), prendi in considerazione l'aggiunta delle colonne Origine e Data di scadenza. Queste colonne ti mostreranno se il materiale chiave in una chiave KMS viene importato o generato da AWS KMS e quando il materiale chiave scade, se del caso. Il campo Data di creazione mostra la data di creazione della chiave KMS (senza il materiale della chiave). Non riflette alcuna caratteristica del materiale della chiave.

Chiavi nello store delle chiavi personalizzate

Se disponi di chiavi KMS negli [archivi delle chiavi personalizzate](#), prendi in considerazione l'aggiunta delle colonne Origin (Origine) e Custom key store ID (ID dell'archivio chiavi personalizzate). Queste colonne mostrano che la chiave KMS si trova in un archivio delle chiavi personalizzate e identificano tale archivio.

Chiavi multi-regione

Se hai [chiavi multi-regione](#), prendi in considerazione l'aggiunta della colonna Regionalità. In questo modo viene mostrato se una chiave KMS è una chiave di singola regione, una [chiave primaria multi-regione](#) o una [chiave di replica multi-regione](#).

Trova le chiavi KMS e il materiale chiave in un archivio di AWS CloudHSM chiavi

Se gestisci un archivio di AWS CloudHSM chiavi, potresti dover identificare le chiavi KMS in ogni AWS CloudHSM archivio di chiavi. Ad esempio, potresti aver bisogno di eseguire alcune delle operazioni seguenti.

- Tieni traccia delle chiavi KMS nell'archivio AWS CloudHSM chiavi nei AWS CloudTrail registri.
- Prevedi l'effetto sulle chiavi KMS della disconnessione di un archivio di chiavi. AWS CloudHSM
- Pianifica l'eliminazione delle chiavi KMS prima di eliminare un AWS CloudHSM archivio di chiavi.

Inoltre, potresti voler identificare le chiavi del tuo AWS CloudHSM cluster che fungono da materiale chiave per le tue chiavi KMS. Sebbene AWS KMS gestisca le chiavi KMS e il materiale chiave, mantieni comunque il controllo e la responsabilità della gestione del AWS CloudHSM cluster, nonché dei backup HSMs e delle chiavi del. HSMs Potrebbe essere necessario identificare le chiavi per controllare il materiale chiave, proteggerlo dall'eliminazione accidentale o eliminarlo dai HSMs backup del cluster dopo l'eliminazione della chiave KMS.

Tutto il materiale chiave per le chiavi KMS nel tuo AWS CloudHSM key store è di proprietà [dell'utente kmsuser crittografico](#) (CU). AWS KMS imposta l'attributo key label, che è visualizzabile solo in AWS CloudHSM, sull'Amazon Resource Name (ARN) della chiave KMS.

Per trovare le chiavi KMS e il materiale della chiave, utilizza una delle tecniche seguenti.

- [Trova le chiavi KMS in un AWS CloudHSM archivio di chiavi](#)— Come identificare le chiavi KMS in uno o tutti i tuoi archivi di chiavi. AWS CloudHSM
- [Trova tutte le chiavi di un archivio di AWS CloudHSM chiavi](#): come trovare nel cluster tutte le chiavi che fungono da materiale della chiave per le chiavi KMS nell'archivio delle chiavi di AWS CloudHSM .
- [Trova la AWS CloudHSM chiave per una chiave KMS](#)— Come trovare la chiave nel cluster che funge da materiale chiave per una particolare chiave KMS nell'archivio delle AWS CloudHSM chiavi.
- [Trova la chiave KMS per una chiave AWS CloudHSM](#) — Come trovare la chiave KMS per un determinata chiave nel cluster.

Trova le chiavi KMS in un AWS CloudHSM archivio di chiavi

Se gestisci un archivio di AWS CloudHSM chiavi, potresti dover identificare le chiavi KMS in ogni AWS CloudHSM archivio di chiavi. Puoi utilizzare queste informazioni per tenere traccia delle operazioni delle chiavi KMS nei AWS CloudTrail registri, prevedere l'effetto della disconnessione di un archivio chiavi personalizzato sulle chiavi KMS o pianificare l'eliminazione delle chiavi KMS prima di eliminare un archivio chiavi. AWS CloudHSM

Per trovare le chiavi KMS in un archivio di chiavi (console) AWS CloudHSM

Per trovare le chiavi KMS in un particolare archivio AWS CloudHSM chiavi, nella pagina Customer managed keys, visualizza i valori nei campi Customer Key Store Name o Custom Key Store ID. Per identificare le chiavi KMS in qualsiasi archivio di AWS CloudHSM chiavi, cerca le chiavi KMS con un valore Origin pari a. AWS CloudHSM Per aggiungere colonne facoltative alla visualizzazione, scegli l'icona che raffigura un ingranaggio nell'angolo in alto a destra della pagina.

Per trovare le chiavi KMS in un archivio di chiavi (AWS CloudHSM API)

Per trovare le chiavi KMS in un archivio di AWS CloudHSM chiavi, usa le [DescribeKey](#) operazioni [ListKeys](#) and, quindi filtra per CustomKeyId valore. Prima di eseguire gli esempi seguenti, sostituisci i valori ID fittizi del key store personalizzato con un valore valido.

Bash

Per trovare le chiavi KMS in un particolare archivio AWS CloudHSM chiavi, recupera tutte le chiavi KMS nell'account e nella regione. Quindi filtra in base all'ID dell'archivio delle chiavi personalizzate.

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;  
do aws kms describe-key --key-id $key |  
grep '"CustomKeyId": "cks-1234567890abcdef0"' --context 100; done
```

Per ottenere le chiavi KMS in qualsiasi archivio di AWS CloudHSM chiavi dell'account e della regione, cerca CustomKeyType con il valore di. AWS_CloudHSM

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;  
do aws kms describe-key --key-id $key |  
grep '"CustomKeyType": "AWS_CloudHSM"' --context 100; done
```


PowerShell

Per trovare le chiavi KMS in un particolare archivio AWS CloudHSM chiavi, utilizza i `KmsKey` cmdlet [Get-KmsKeyList](#) e [Get-](#) per ottenere tutte le chiavi KMS nell'account e nella regione. Quindi filtra in base all'ID dell'archivio delle chiavi personalizzate.

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreId -eq  
'cks-1234567890abcdef0'
```

Per ottenere le chiavi KMS in qualsiasi AWS CloudHSM archivio di chiavi dell'account e della regione, filtra in base al valore di `CustomKeyStoreType` `AWS_CLOUDHSM`

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreType -eq 'AWS_CLOUDHSM'
```

Trova tutte le chiavi di un archivio di AWS CloudHSM chiavi

Puoi identificare le chiavi del tuo AWS CloudHSM cluster che fungono da materiale chiave per il tuo archivio di AWS CloudHSM chiavi. A tale scopo, utilizzate il comando [key list](#) nella CLI di CloudHSM.

Puoi anche usare il comando `key list` per trovare la chiave AWS KMS for an. AWS CloudHSM. Quando AWS KMS crea il materiale chiave per una chiave KMS nel tuo AWS CloudHSM cluster, scrive l'Amazon Resource Name (ARN) della chiave KMS nell'etichetta della chiave. Il comando `key list` restituisce `ilkey-reference.label`

Note

Le seguenti procedure utilizzano lo strumento da riga di comando AWS CloudHSM Client SDK 5, [CloudHSM](#) CLI. La CLI di CloudhSM sostituisce con. `key-handle` `key-reference`. Il 1° gennaio 2025, AWS CloudHSM terminerà il supporto per gli strumenti da riga di comando di Client SDK 3, la CloudHSM Management Utility (CMU) e la Key Management Utility (KMU). Per ulteriori informazioni sulle differenze tra gli strumenti da riga di comando di Client SDK 3 e lo strumento da riga di comando di Client SDK 5, consulta [Migrare da Client SDK 3 CMU e KMU a Client SDK 5 CloudHSM CLI nella Guida per l'utente AWS CloudHSM](#)

Per eseguire questa procedura, è necessario disconnettere temporaneamente l'archivio delle AWS CloudHSM chiavi in modo da poter accedere come `CU.kmsuser`

1. Disconnetti il AWS CloudHSM key store, se non è già disconnesso, quindi accedi come `kmsuser` spiegato in [Come disconnettersi ed eseguire l'accesso](#)

Note

Quando un archivio delle chiavi personalizzate è disconnesso, tutti i tentativi di creare chiavi KMS nell'archivio delle chiavi personalizzate o di utilizzare le chiavi KMS in operazioni di crittografia avrà esito negativo. Questa azione può impedire agli utenti di archiviare e accedere ai dati sensibili.

2. Utilizza il comando [key list](#) nella CLI di CloudHSM per trovare tutte le chiavi per l'utente corrente presente nel cluster. AWS CloudHSM

Per impostazione predefinita, vengono visualizzate solo 10 chiavi dell'utente attualmente connesso e solo la `key-reference` e la `label` vengono visualizzate come output. Per altre opzioni, consulta l'[elenco delle chiavi nella Guida](#) per l'AWS CloudHSM utente.

```
aws-cloudhsm > key list
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x00000000000000123",
        "attributes": {
          "label": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
      },
      {
        "key-reference": "0x00000000000000456",
        "attributes": {
          "label": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
        }
      },
      ...8 keys later...
    ],
    "total_key_count": 56,
    "returned_key_count": 10,
    "next_token": "10"
```

```
}  
}
```

3. Disconnettersi e ricollegare l'archivio delle AWS CloudHSM chiavi come descritto in [Come scollegarsi e riconnettersi](#).

Trova la chiave KMS per una chiave AWS CloudHSM

Se conosci il riferimento o l'ID di una chiave di cui è `kmsuser` proprietario nel cluster, puoi utilizzare quel valore per identificare la chiave KMS associata nel tuo archivio di AWS CloudHSM chiavi.

Quando AWS KMS crea il materiale chiave per una chiave KMS nel tuo AWS CloudHSM cluster, scrive l'Amazon Resource Name (ARN) della chiave KMS nell'etichetta della chiave. A meno che tu non abbia modificato il valore dell'etichetta, puoi utilizzare il comando [key list](#) nella CLI di CloudHSM per identificare la chiave KMS associata alla chiave. AWS CloudHSM

Note

Le seguenti procedure utilizzano lo strumento da riga di comando AWS CloudHSM Client SDK 5, [CloudHSM CLI](#). La CLI di CloudhSM sostituisce con. `key-handle` `key-reference`. Il 1° gennaio 2025, AWS CloudHSM terminerà il supporto per gli strumenti da riga di comando di Client SDK 3, la CloudHSM Management Utility (CMU) e la Key Management Utility (KMU). Per ulteriori informazioni sulle differenze tra gli strumenti da riga di comando di Client SDK 3 e lo strumento da riga di comando di Client SDK 5, consulta [Migrare da Client SDK 3 CMU e KMU a Client SDK 5 CloudHSM CLI nella Guida per l'utente](#). AWS CloudHSM

Per eseguire queste procedure è necessario disconnettere temporaneamente l'archivio delle AWS CloudHSM chiavi in modo da poter accedere come CU. `kmsuser`

Note

Quando un archivio delle chiavi personalizzate è disconnesso, tutti i tentativi di creare chiavi KMS nell'archivio delle chiavi personalizzate o di utilizzare le chiavi KMS in operazioni di crittografia avrà esito negativo. Questa azione può impedire agli utenti di archiviare e accedere ai dati sensibili.

Argomenti

- [Identifica la chiave KMS associata a un riferimento chiave](#)
- [Identifica la chiave KMS associata all'ID di una chiave di backup](#)

Identifica la chiave KMS associata a un riferimento chiave

Le seguenti procedure mostrano come utilizzare il comando [key list](#) nella CLI di CloudHSM key-reference con il filtro degli attributi per trovare la chiave nel cluster che funge da materiale chiave per una particolare chiave KMS nel tuo archivio di chiavi. AWS CloudHSM

1. Disconnetti l'archivio delle AWS CloudHSM chiavi, se non è già disconnesso, quindi accedi come spiegato in. kmsuser [Come disconnettersi ed eseguire l'accesso](#)
2. Utilizza il comando [key list](#) nella CLI di CloudHSM per filtrare in base all'attributo. key-reference Specificate l'verboseargomento per includere tutti gli attributi e le informazioni chiave per la chiave corrispondente. Se non si specifica l'verboseargomento, l'operazione di elenco delle chiavi restituisce solo il riferimento di chiave e l'attributo label della chiave corrispondente.

Prima di eseguire questo comando, sostituisci l'esempio key-reference con uno valido dal tuo account.

```
aws-cloudhsm > key list --filter attr.key-reference="0x0000000000120034" --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x0000000000120034",
        "key-info": {
          "key-owners": [
            {
              "username": "kmsuser",
              "key-coverage": "full"
            }
          ],
          "shared-users": [],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "aes",
```

```

    "label": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "id": "0xbacking-key-id",
    "check-value": "0x29bbd1",
    "class": "my_test_key",
    "encrypt": true,
    "decrypt": true,
    "token": true,
    "always-sensitive": true,
    "derive": false,
    "destroyable": true,
    "extractable": false,
    "local": true,
    "modifiable": true,
    "never-extractable": false,
    "private": true,
    "sensitive": true,
    "sign": false,
    "trusted": false,
    "unwrap": true,
    "verify": false,
    "wrap": true,
    "wrap-with-trusted": false,
    "key-length-bytes": 32
  }
},
"total_key_count": 1,
"returned_key_count": 1
}
}

```

3. Disconnettersi e ricollegare l'archivio delle AWS CloudHSM chiavi come descritto in [Come scollegarsi e riconnettersi](#).

Identifica la chiave KMS associata all'ID di una chiave di backup

Tutte le voci di CloudTrail registro per le operazioni crittografiche con una chiave KMS in un archivio di AWS CloudHSM chiavi includono un `additionalEventData` campo con e. `customKeyId` `backingKeyId` Il valore restituito nel `backingKeyId` campo è correlato all'attributo chiave `id` CloudHSM. È possibile filtrare l'operazione dell'[elenco di chiavi](#) in base all'id attributo per identificare la chiave KMS associata a uno specifico. `backingKeyId`

1. Disconnetti l'archivio delle AWS CloudHSM chiavi, se non è già disconnesso, quindi accedi come spiegato kmsuser in. [Come disconnettersi ed eseguire l'accesso](#)
2. Utilizza il comando [key list](#) nella CLI di CloudHSM con il filtro degli attributi per trovare la chiave nel cluster che funge da materiale chiave per una particolare chiave KMS nel tuo archivio di chiavi. AWS CloudHSM

L'esempio seguente mostra come filtrare in base all'attributo. id AWS CloudHSM riconosce il id valore come valore esadecimale. Per filtrare l'operazione dell'elenco di chiavi in base all'id attributo, è necessario innanzitutto convertire il backingKeyId valore identificato nella voce di CloudTrail registro in un formato che lo riconosca. AWS CloudHSM

- a. Utilizzate il seguente comando Linux per convertire il file backingKeyId in una rappresentazione esadecimale.

```
echo backingKeyId | tr -d '\n' | xxd -p
```

L'esempio seguente mostra come convertire l'array di backingKeyId byte in una rappresentazione esadecimale.

```
echo 5890723622dc15f699aa9ab2387a9f744b2b884c18b2186ee8ada4f556a2eb9d | tr -d '\n' | xxd -p
353839303732333632326463313566363939616139616232333837613966373434623262383834633138623
```

- b. Antepone la rappresentazione esadecimale di with. backingKeyId 0x

```
0x353839303732333632326463313566363939616139616232333837613966373434623262383834633138623
```

- c. Utilizzate il backingKeyId valore convertito per filtrare in base all'attributo. id Specificate l'verboseargomento per includere tutti gli attributi e le informazioni chiave per la chiave corrispondente. Se non si specifica l'verboseargomento, l'operazione di elenco delle chiavi restituisce solo il riferimento di chiave e l'attributo label della chiave corrispondente.

```
aws-cloudhsm > key list --filter
  attr.id="0x353839303732333632326463313566363939616139616232333837613966373434623262383"
  --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
```

```

    "key-reference": "0x00000000000120034",
    "key-info": {
      "key-owners": [
        {
          "username": "kmsuser",
          "key-coverage": "full"
        }
      ],
      "shared-users": [],
      "cluster-coverage": "full"
    },
    "attributes": {
      "key-type": "aes",
      "label": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "id":
      "0x35383930373233363232646331356636393961613961623233383761396637343462326238383463313
      "check-value": "0x29bbd1",
      "class": "my_test_key",
      "encrypt": true,
      "decrypt": true,
      "token": true,
      "always-sensitive": true,
      "derive": false,
      "destroyable": true,
      "extractable": false,
      "local": true,
      "modifiable": true,
      "never-extractable": false,
      "private": true,
      "sensitive": true,
      "sign": false,
      "trusted": false,
      "unwrap": true,
      "verify": false,
      "wrap": true,
      "wrap-with-trusted": false,
      "key-length-bytes": 32
    }
  },
  ],
  "total_key_count": 1,
  "returned_key_count": 1
}

```

```
}
```

3. Disconnettersi e ricollegare l'archivio delle AWS CloudHSM chiavi come descritto in. [Come scollegarsi e riconnettersi](#)

Trova la AWS CloudHSM chiave per una chiave KMS

Puoi utilizzare l'ID della chiave KMS di una chiave KMS in un archivio di AWS CloudHSM chiavi per identificare la chiave del AWS CloudHSM cluster che funge da materiale chiave.

Quando AWS KMS crea il materiale chiave per una chiave KMS nel tuo AWS CloudHSM cluster, scrive l'Amazon Resource Name (ARN) della chiave KMS nell'etichetta della chiave. A meno che tu non abbia modificato il valore dell'etichetta, puoi utilizzare il comando [key list](#) nella CLI di CloudHSM per trovare la risorsa chiave e l'id del materiale chiave per la chiave KMS.

Tutte le voci di CloudTrail registro per le operazioni crittografiche con una chiave KMS in un archivio di chiavi includono un AWS CloudHSM campo con `and. additionalEventData customKeyStoreId backingKeyId` Il valore restituito nel `backingKeyId` campo è l'attributo `id` AWS CloudHSM chiave. Puoi filtrare l'operazione AWS CloudHSM CLI dell'elenco di chiavi in base alla chiave KMS ARN per identificare l'attributo chiave CloudHSM associato a una chiave KMS specifica. `id`

Per eseguire questa procedura, è necessario disconnettere temporaneamente l'archivio delle AWS CloudHSM chiavi in modo da poter accedere come `CU.kmsuser`

Note

Le seguenti procedure utilizzano lo strumento da riga di comando AWS CloudHSM Client SDK 5, [CloudHSM](#) CLI. La CLI di CloudhSM sostituisce con. `key-handle` `key-reference` Il 1° gennaio 2025, AWS CloudHSM terminerà il supporto per gli strumenti da riga di comando di Client SDK 3, la CloudHSM Management Utility (CMU) e la Key Management Utility (KMU). Per ulteriori informazioni sulle differenze tra gli strumenti da riga di comando di Client SDK 3 e lo strumento da riga di comando di Client SDK 5, consulta [Migrare da Client SDK 3 CMU e KMU a Client SDK 5 CloudHSM CLI nella Guida per l'utente](#).AWS CloudHSM

1. Disconnetti l'archivio delle AWS CloudHSM chiavi, se non è già disconnesso, quindi accedi come spiegato in. `kmsuser` [Come disconnettersi ed eseguire l'accesso](#)

Note

Quando un archivio delle chiavi personalizzate è disconnesso, tutti i tentativi di creare chiavi KMS nell'archivio delle chiavi personalizzate o di utilizzare le chiavi KMS in operazioni di crittografia avrà esito negativo. Questa azione può impedire agli utenti di archiviare e accedere ai dati sensibili.

2. Usa il comando [key list](#) nella CLI di CloudHSM e `label` filtra per trovare la chiave KMS per una particolare chiave nel tuo cluster. AWS CloudHSM Specificate l'verboseargomento per includere tutti gli attributi e le informazioni chiave per la chiave corrispondente. Se non si specifica l'verboseargomento, l'operazione di elenco delle chiavi restituisce solo gli attributi di riferimento e etichetta della chiave corrispondente.

L'esempio seguente mostra come filtrare in base all'`label`attributo che memorizza la chiave KMS ARN. Prima di eseguire questo comando, sostituisci la chiave KMS di esempio ARN con una chiave valida del tuo account.

```
aws-cloudhsm > key list --filter attr.label="arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab" --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x000000000000120034",
        "key-info": {
          "key-owners": [
            {
              "username": "kmsuser",
              "key-coverage": "full"
            }
          ],
          "shared-users": [],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "aes",
          "label": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "id": "0xbacking-key-id",
```

```
    "check-value": "0x29bbd1",
    "class": "my_test_key",
    "encrypt": true,
    "decrypt": true,
    "token": true,
    "always-sensitive": true,
    "derive": false,
    "destroyable": true,
    "extractable": false,
    "local": true,
    "modifiable": true,
    "never-extractable": false,
    "private": true,
    "sensitive": true,
    "sign": false,
    "trusted": false,
    "unwrap": true,
    "verify": false,
    "wrap": true,
    "wrap-with-trusted": false,
    "key-length-bytes": 32
  }
},
"total_key_count": 1,
"returned_key_count": 1
}
```

3. Disconnettersi e ricollegare l'archivio delle AWS CloudHSM chiavi come descritto in. [Come scollegarsi e riconnettersi](#)

Attivazione e disattivazione delle chiavi

È possibile disabilitare e riabilitare le chiavi gestite del cliente. Quando crei una chiave KMS è abilitata per impostazione predefinita. Se disabiliti una chiave KMS, non può essere utilizzata in nessuna [operazione di crittografia](#) fino a quando non lo riabiliti.

Poiché è temporaneo e facilmente annullato, la disabilitazione di una chiave KMS è un'alternativa sicura all'eliminazione di una chiave KMS, un'operazione distruttiva e irreversibile. Se state pensando di eliminare una chiave KMS, disattivatela prima e impostate un [CloudWatch allarme](#) o un meccanismo simile per essere certi che non avrete mai bisogno di usare la chiave per decrittografare i dati crittografati.

Quando disabiliti una chiave KMS, diventa immediatamente inutilizzabile (in base alla coerenza finale). Tuttavia, le risorse crittografate con [chiavi di dati](#) protette dalla chiave KMS non sono interessate fino a quando la chiave KMS non viene nuovamente utilizzata, ad esempio per decrittografare la chiave dati. Questo problema riguarda Servizi AWS molti dei quali utilizzano chiavi dati per proteggere le risorse. Per informazioni dettagliate, consultare [In che modo le chiavi KMS inutilizzabili influiscono sulle chiavi dati](#).

Non è possibile abilitare o disabilitare [Chiavi gestite da AWS](#) o [Chiavi di proprietà di AWS](#). Chiavi gestite da AWS sono permanentemente abilitati all'uso da parte [dei servizi che utilizzano AWS KMS](#). Chiavi di proprietà di AWS sono gestiti esclusivamente dal servizio che li possiede.

Note

AWS KMS non ruota il materiale chiave delle chiavi gestite dal cliente quando sono disattivate. Per ulteriori informazioni, consulta [Come funziona la rotazione dei tasti](#).

Utilizzo della console AWS KMS

È possibile utilizzare la AWS KMS console per abilitare e disabilitare [le chiavi gestite dal cliente](#).

1. Accedi Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.

3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
4. Scegli la casella di controllo delle chiavi KMS che vuoi abilitare o disabilitare.
5. Per abilitare una chiave KMS, scegli Azioni chiave, Abilita. Per disabilitare una chiave KMS, scegli Azioni chiave, Disabilita.

Utilizzando l'API AWS KMS

L'[EnableKey](#) operazione abilita un disabilitato AWS KMS key. Questi esempi utilizzano la [AWS Command Line Interface \(AWS CLI\)](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato. Il parametro `key-id` è obbligatorio.

Questa operazione non restituisce alcun output. Per visualizzare lo stato della chiave, utilizzare l'[DescribeKey](#) operazione.

```
$ aws kms enable-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

L'[DisableKey](#) operazione disabilita una chiave KMS abilitata. Il parametro `key-id` è obbligatorio.

```
$ aws kms disable-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Questa operazione non restituisce alcun output. Per vedere lo stato della chiave, usa l'[DescribeKey](#) operazione e guarda il `Enabled` campo.

```
$ aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
  "KeyMetadata": {
    "Origin": "AWS_KMS",
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "Description": "",
    "KeyManager": "CUSTOMER",
    "MultiRegion": false,
    "Enabled": false,
    "KeyState": "Disabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "CreationDate": 1502910355.475,
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "KeySpec": "SYMMETRIC_DEFAULT",
```

```
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",  
    "EncryptionAlgorithms": [  
        "SYMMETRIC_DEFAULT"  
    ]  
}
```

Ruotare AWS KMS keys

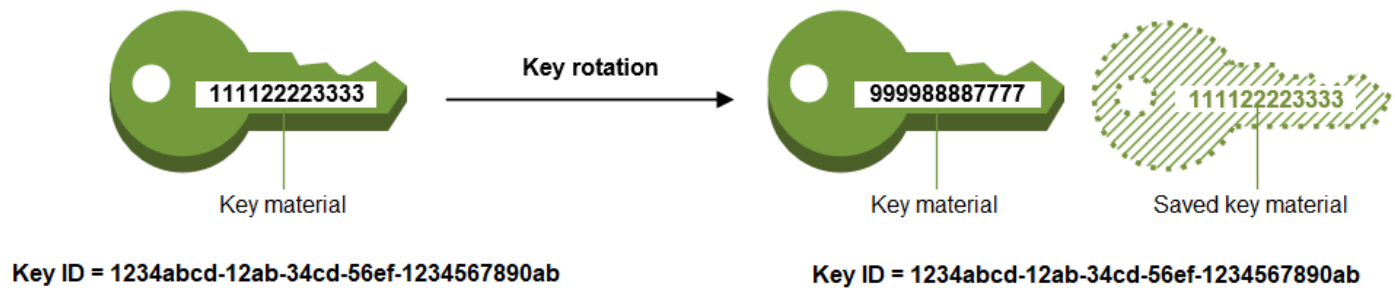
Per creare nuovo materiale crittografico per le [chiavi gestite dal cliente](#), puoi creare nuove chiavi KMS e modificare le applicazioni o gli alias per utilizzarle. In alternativa, puoi ruotare il materiale chiave associato a una chiave KMS esistente abilitando la rotazione automatica delle chiavi o eseguendo la rotazione su richiesta.

Per impostazione predefinita, quando abiliti la rotazione automatica delle chiavi per una chiave KMS, AWS KMS genera nuovo materiale crittografico per la chiave KMS ogni anno. Puoi anche specificare un'impostazione personalizzata [rotation-period](#) per definire il numero di giorni dopo l'attivazione della rotazione automatica delle chiavi che AWS KMS ruoterà il materiale della chiave e il numero di giorni tra ogni rotazione automatica successiva. Se è necessario avviare immediatamente la rotazione del materiale chiave, è possibile eseguire la rotazione su richiesta, indipendentemente dal fatto che la rotazione automatica dei tasti sia abilitata o meno. Le rotazioni su richiesta non modificano i programmi di rotazione automatica esistenti.

Puoi [tenere traccia della rotazione](#) del materiale chiave per le tue chiavi KMS in Amazon CloudWatch e nella AWS Key Management Service console. AWS CloudTrail Puoi anche utilizzare l'[GetKeyRotationStatus](#) operazione per verificare se la rotazione automatica è abilitata per una chiave KMS e identificare eventuali rotazioni su richiesta in corso. È possibile utilizzare l'[ListKeyRotations](#) operazione per visualizzare i dettagli delle rotazioni completate.

La rotazione delle chiavi modifica solo il materiale chiave corrente, che è il segreto crittografico utilizzato nelle operazioni di crittografia. Quando si utilizza la chiave KMS ruotata per decrittografare il testo cifrato, AWS KMS utilizza il materiale chiave utilizzato per crittografarlo. Non è possibile selezionare un particolare materiale chiave per le operazioni di decrittografia, sceglie automaticamente il materiale chiave corretto. AWS KMS Poiché esegue la decrittografia AWS KMS in modo trasparente con il materiale chiave appropriato, è possibile utilizzare in sicurezza una chiave KMS ruotata nelle applicazioni e senza modifiche al codice. Servizi AWS

La chiave KMS è la stessa risorsa logica, indipendentemente dal fatto o dal numero di volte che supporta le modifiche del materiale chiave. Le proprietà della chiave KMS non vengono modificate, come illustrato nella seguente immagine.



Potresti decidere di creare una nuova chiave KMS e di utilizzarla al posto della chiave KMS originale. Lo stesso risultato si ottiene ruotando il materiale della chiave in una chiave KMS esistente, quindi questa operazione viene spesso considerata come una [rotazione manuale della chiave](#). [La rotazione manuale è un'ottima scelta quando si desidera ruotare chiavi KMS che non sono idonee per la rotazione automatica o su richiesta, tra cui chiavi KMS asimmetriche, chiavi KMS HMAC, chiavi KMS negli archivi di chiavi personalizzati e chiavi KMS multiregionali con materiale chiave importato.](#)

Note

La rotazione delle chiavi non ha alcun effetto sui dati protetti dalla chiave KMS. Non ruota le chiavi dati generate dalla chiave KMS né cripta nuovamente i dati protetti dalla chiave KMS. La rotazione delle chiavi non mitigherà l'effetto di una chiave dati compromessa.

Rotazione delle chiavi e prezzi

AWS KMS addebita una tariffa mensile per la prima e la seconda rotazione del materiale chiave utilizzato per la chiave KMS. Questo aumento di prezzo è limitato alla seconda rotazione e le eventuali rotazioni successive non verranno fatturate. Per informazioni dettagliate, consulta [Prezzi di AWS Key Management Service](#).

Note

Puoi utilizzare il [AWS Cost Explorer Service](#) per visualizzare un dettaglio dei costi di archiviazione delle chiavi. Ad esempio, puoi filtrare la vista per visualizzare i costi totali per le chiavi fatturate come chiavi KMS correnti e ruotate specificando `$REGION-KMS-Keys` per Tipo di utilizzo e raggruppando i dati in base a Operazione API. Potresti ancora visualizzare le istanze dell'operazione dell'API con legacy Unknown per le date nella cronologia.

Rotazione delle chiavi e quote

Quando si calcolano le quote di risorse chiave, ogni chiave KMS conta come una chiave indipendentemente dal numero di versioni del materiale chiave ruotato.

Per informazioni più dettagliate sul materiale chiave e sulla rotazione, consultare [Dettagli di crittografia di AWS Key Management Service](#).

Argomenti

- [Perché ruotare le chiavi KMS?](#)
- [Come funziona la rotazione dei tasti](#)
- [Abilita la rotazione automatica dei tasti](#)
- [Disabilita la rotazione automatica delle chiavi](#)
- [Esegui la rotazione dei tasti su richiesta](#)
- [Elenca le rotazioni e i materiali chiave](#)
- [Ruota i tasti manualmente](#)
- [Cambia la chiave primaria in un set di chiavi multiregionali](#)

Perché ruotare le chiavi KMS?

[Le migliori pratiche crittografiche scoraggiano il riutilizzo estensivo delle chiavi che crittografano direttamente i dati, come le chiavi di dati che generano.](#) AWS KMS Quando eseguono la crittografia di milioni di messaggi, le chiavi dati a 256 bit possono esaurirsi e iniziare a produrre testo criptato con trame sottili che possono essere sfruttate da abili malintenzionati per individuare i bit delle chiavi. È meglio utilizzare le chiavi dati una volta, o solo poche volte, per mitigare questo esaurimento delle chiavi.

Tuttavia, le chiavi KMS vengono spesso utilizzate per lo più come chiavi di wrapping, note anche come chiavi di crittografia delle chiavi. Invece di crittografare i dati, le chiavi di wrapping eseguono la crittografia delle chiavi di dati che eseguono a loro volta la crittografia dei tuoi dati. Per tale motivo, vengono utilizzate molto meno spesso delle chiavi di dati e non vengono quasi mai riutilizzate abbastanza da rischiare l'esaurimento delle chiavi.

Anche se questo rischio di esaurimento è bassissimo, potrebbe essere necessario ruotare le chiavi KMS a causa di regole aziendali/contrattuali o norme governative. Quando sei obbligato a ruotare le chiavi KMS, ti consigliamo di utilizzare la rotazione automatica dei tasti laddove è supportata, di

utilizzare la rotazione su richiesta se la rotazione automatica non è supportata e la rotazione manuale dei tasti quando non è supportata né la rotazione automatica né quella su richiesta.

Potresti prendere in considerazione l'esecuzione di rotazioni su richiesta per dimostrare le principali funzionalità di rotazione dei materiali o per convalidare gli script di automazione. [Si consiglia di utilizzare le rotazioni su richiesta per le rotazioni non pianificate e di utilizzare la rotazione automatica dei tasti con un periodo di rotazione personalizzato, ove possibile.](#)

Come funziona la rotazione dei tasti

AWS KMS la rotazione dei tasti è progettata per essere trasparente e facile da usare. AWS KMS supporta la rotazione opzionale automatica e su richiesta solo per le [chiavi gestite dal cliente](#).

rotazione automatica dei tasti

AWS KMS ruota automaticamente la chiave KMS alla data di rotazione successiva definita dal periodo di rotazione. Non devi ricordare o pianificare l'aggiornamento.

La rotazione automatica delle chiavi è supportata solo sulle chiavi KMS a crittografia simmetrica con materiale chiave generato (origine). AWS KMS AWS_KMS

La rotazione automatica è opzionale per le chiavi KMS gestite dal cliente. AWS KMS ruota sempre ogni anno il materiale chiave per le chiavi KMS AWS gestite. La rotazione delle chiavi KMS di AWS proprietà è gestita dal proprietario della Servizio AWS chiave.

Rotazione su richiesta

Avvia immediatamente la rotazione del materiale chiave associato alla tua chiave KMS, indipendentemente dal fatto che la rotazione automatica delle chiavi sia abilitata o meno.

La rotazione delle chiavi su richiesta è supportata su chiavi KMS di crittografia simmetrica con materiale chiave che AWS KMS genera (AWS_KMSorigine) e chiavi KMS di crittografia simmetrica con materiale chiave importato (origine). EXTERNAL

Rotazione manuale

Non è supportata né la rotazione automatica né quella su richiesta per i seguenti tipi di chiavi KMS, ma è possibile [ruotare queste chiavi KMS](#) manualmente.

- [Chiavi KMS asimmetriche](#)
- [Chiavi KMS HMAC](#)
- Chiavi KMS nell'[archivio delle chiavi personalizzate](#)

Gestione del materiale chiave

AWS KMS conserva tutto il materiale chiave per una chiave KMS con AWS_KMS origine, anche se la rotazione dei tasti è disabilitata. AWS KMS elimina il materiale chiave solo quando elimini la chiave KMS.

Gestisci i materiali chiave per le chiavi di crittografia simmetriche con origine. EXTERNAL È possibile eliminare qualsiasi materiale chiave utilizzando l'[DeleteImportedKeyMaterial](#) operazione o impostare una scadenza per l'importazione del materiale. La chiave KMS diventa inutilizzabile non appena uno dei suoi materiali scade o viene eliminato.

Utilizzo del materiale chiave

Quando si utilizza una chiave KMS ruotata per crittografare i dati, AWS KMS utilizza il materiale chiave corrente. Quando si utilizza la chiave KMS ruotata per decrittare il testo cifrato, AWS KMS usa la stessa versione del materiale chiave utilizzata per la crittografia. Non è possibile selezionare una versione particolare del materiale chiave per le operazioni di decrittografia, sceglie AWS KMS automaticamente la versione corretta.

Periodo di rotazione

Il periodo di rotazione definisce il numero di giorni dopo l'attivazione della rotazione automatica dei tasti che AWS KMS farà ruotare il materiale chiave e il numero di giorni tra ogni rotazione automatica dei tasti successiva. Se non specificate un valore per `RotationPeriodInDays` quando abilitate la rotazione automatica dei tasti, il valore predefinito è 365 giorni.

Puoi usare la chiave [kms: RotationPeriodInDays](#) condition per limitare ulteriormente i valori che i principals possono specificare nel parametro. `RotationPeriodInDays`

Data di rotazione

La data di rotazione riflette la data in cui il materiale chiave corrente per una chiave KMS è stato aggiornato come risultato di una rotazione automatica (pianificata) o di una rotazione delle chiavi su richiesta.

Data di rotazione

AWS KMS ruota automaticamente la chiave KMS alla data di rotazione definita dal periodo di rotazione. Il periodo di rotazione predefinito è 365 giorni.

Chiavi gestite dal cliente

Siccome la rotazione automatica delle chiavi è facoltativa nelle [chiavi gestite dal cliente](#) e può essere abilitata e disabilitata in qualunque momento, la data di rotazione dipende dalla

data dell'ultima abilitazione. La data può cambiare se si modifica il periodo di rotazione di una chiave su cui in precedenza era stata abilitata la rotazione automatica della chiave. La data di rotazione può cambiare molte volte nel corso della durata della chiave.

Ad esempio, se crei una chiave gestita dal cliente il 1° gennaio 2022 e abiliti la rotazione automatica delle chiavi con il periodo di rotazione predefinito di 365 giorni il 15 marzo 2022, AWS KMS ruota il materiale chiave il 15 marzo 2023, il 15 marzo 2024 e successivamente ogni 365 giorni.

Gli esempi seguenti presuppongono che la rotazione automatica dei tasti sia stata abilitata con il periodo di rotazione predefinito di 365 giorni. Questi esempi illustrano casi speciali che potrebbero influire sul periodo di rotazione di una chiave.

- Disabilitazione della rotazione delle chiavi: se [disabiliti la rotazione automatica delle chiavi](#) in un determinato momento, la chiave KMS continua a utilizzare la versione del materiale della chiave che utilizzava quando la rotazione era disabilitata. Se abilitate nuovamente la rotazione automatica dei tasti, AWS KMS ruota il materiale chiave in base alla nuova data di attivazione della rotazione.
- Chiavi KMS disabilite: sebbene una chiave KMS sia disabilitata, non la ruota. AWS KMS Tuttavia, lo stato della rotazione della chiave non cambia e non è possibile modificarlo quando la chiave KMS è disattivata. Quando la chiave KMS viene riattivata, se il materiale chiave ha superato l'ultima data di rotazione programmata, lo ruota immediatamente. AWS KMS Se il materiale chiave non ha perso l'ultima data di rotazione programmata, AWS KMS riprende la pianificazione di rotazione della chiave originale.
- Chiavi KMS in attesa di cancellazione: mentre una chiave KMS è in attesa di cancellazione, non la ruota. AWS KMS Lo stato di rotazione della chiave è impostato su `false` e non è possibile modificarlo quando l'eliminazione è in sospeso. Se l'eliminazione è stata annullata, viene ripristinato il precedente stato di rotazione della chiave. Se il materiale chiave ha superato l'ultima data di rotazione programmata, lo ruota immediatamente. AWS KMS Se il materiale chiave non ha perso l'ultima data di rotazione programmata, AWS KMS riprende il programma di rotazione della chiave originale.

Chiavi gestite da AWS

AWS KMS ruota automaticamente Chiavi gestite da AWS ogni anno (circa 365 giorni). Non è possibile abilitare o disabilitare la rotazione delle chiavi per le [Chiavi gestite da AWS](#).

Il materiale chiave di un Chiave gestita da AWS viene ruotato per la prima volta un anno dopo la data di creazione e successivamente ogni anno (circa 365 giorni dall'ultima rotazione).

 Note

A maggio 2022, AWS KMS ha modificato il programma di rotazione Chiavi gestite da AWS da ogni tre anni (circa 1.095 giorni) a ogni anno (circa 365 giorni).

Chiavi di proprietà di AWS

Non è possibile abilitare o disabilitare la rotazione delle chiavi per le Chiavi di proprietà di AWS. La strategia di [rotazione delle chiavi](#) per un Chiave di proprietà di AWS è determinata dal AWS servizio che crea e gestisce la chiave. Per informazioni dettagliate, consulta l'argomento Crittografia dei dati inattivi nella guida per l'utente o nella guida per gli sviluppatori del servizio.

Tasti multiregionali rotanti

[È possibile abilitare e disabilitare la rotazione automatica ed eseguire la rotazione su richiesta del materiale chiave in chiavi multiregionali con crittografia simmetrica con origine.](#) AWS_KMS La rotazione delle chiavi è una [proprietà condivisa](#) delle chiavi multiregionali.

Puoi abilitare e disabilitare la rotazione automatica delle chiavi solo sulla chiave primaria. Si avvia la rotazione su richiesta solo sulla chiave primaria.

- Quando AWS KMS sincronizza le chiavi multiregionali, copia l'impostazione della proprietà di rotazione dei tasti dalla chiave primaria a tutte le relative chiavi di replica.
- Quando AWS KMS ruota il materiale chiave, crea nuovo materiale chiave per la chiave primaria e quindi copia il nuovo materiale chiave oltre i confini della regione in tutte le chiavi di replica correlate. Il materiale chiave non esce AWS KMS mai non crittografato. Questo passaggio viene controllato con attenzione per garantire che il materiale chiave sia completamente sincronizzato prima che qualsiasi chiave venga utilizzata in un'operazione di crittografia.
- AWS KMS non cripta alcun dato con il nuovo materiale chiave finché tale materiale chiave non è disponibile nella chiave primaria e in tutte le relative chiavi di replica.
- Quando si replica una chiave primaria che è stata ruotata, la nuova chiave di replica contiene il materiale chiave corrente e tutte le versioni precedenti del materiale chiave per le relative chiavi multiregione.

Questo modello assicura che le chiavi multiregione correlate siano completamente interoperabili. Qualsiasi chiave multiregione può decrittare qualsiasi testo cifrato crittografato da una chiave

multiregione correlata, anche se il testo cifrato è stato crittografato prima della creazione della chiave.

AWS servizi

È possibile abilitare la rotazione automatica delle [chiavi gestite dal cliente](#) utilizzate per la crittografia lato server nei AWS servizi. La rotazione annuale è trasparente e compatibile con AWS i servizi.

Monitoraggio della rotazione delle chiavi

Quando AWS KMS ruota il materiale chiave per una chiave [Chiave gestita da AWS](#) o una [chiave gestita dal cliente](#), scrive un KMS CMK Rotation evento su Amazon EventBridge e un [RotateKey evento](#) nel tuo AWS CloudTrail registro. Puoi utilizzare questi record per verificare che la chiave KMS sia stata ruotata.

Puoi utilizzare la AWS Key Management Service console per visualizzare il numero di rotazioni rimanenti su richiesta e un elenco di tutte le rotazioni dei materiali chiave completate per una chiave KMS.

È possibile utilizzare l'[ListKeyRotations](#) operazione per visualizzare i dettagli delle rotazioni completate.

Consistenza finale

La rotazione delle chiavi è soggetta agli stessi eventuali effetti di coerenza AWS KMS delle altre operazioni di gestione. Potrebbe esserci un leggero ritardo prima che il nuovo materiale chiave sia disponibile in AWS KMS. Tuttavia, la rotazione del materiale chiave non causa alcuna interruzione o ritardo nelle operazioni di crittografia. Il materiale chiave corrente viene utilizzato nelle operazioni di crittografia fino a quando il nuovo materiale chiave non è disponibile in AWS KMS. Quando il materiale chiave per una chiave multiregionale viene ruotato automaticamente, AWS KMS utilizza il materiale chiave corrente finché il nuovo materiale chiave non è disponibile in tutte le regioni con una chiave multiregionale correlata.

Abilita la rotazione automatica dei tasti

Per impostazione predefinita, quando abiliti la rotazione automatica delle chiavi per una chiave KMS, AWS KMS genera nuovo materiale crittografico per la chiave KMS ogni anno. Puoi anche specificare un'impostazione personalizzata [rotation-period](#) per definire il numero di giorni dopo l'attivazione della rotazione automatica delle chiavi che AWS KMS ruoterà il materiale della chiave e il numero di giorni tra ogni rotazione automatica successiva.

La rotazione automatica delle chiavi ha i seguenti vantaggi:

- Le proprietà della , tra cui l'[ID chiave](#), l'[ARN chiave](#), la Regione, le policy e le autorizzazioni, non cambiano quando la chiave viene ruotata.
- Non è necessario modificare le applicazioni o gli alias che fanno riferimento all'ID o all'ARN della chiave KMS.
- La rotazione del materiale della chiave non influisce sull'uso della chiave KMS in nessun Servizio AWS.
- Dopo aver abilitato la rotazione dei tasti, AWS KMS ruota automaticamente la chiave KMS alla data di rotazione successiva definita dal periodo di rotazione. Non devi ricordare o pianificare l'aggiornamento.

È possibile abilitare la rotazione automatica dei tasti nella AWS KMS console o utilizzando l'[EnableKeyRotation](#) operazione. Per abilitare la rotazione automatica dei tasti, sono necessarie `kms:EnableKeyRotation` le autorizzazioni. Per ulteriori informazioni sulle AWS KMS autorizzazioni, vedere. [Riferimento per le autorizzazioni](#)

Utilizzo della console AWS KMS

1. Accedi Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente. Non è possibile abilitare o disabilitare la rotazione delle Chiavi gestite da AWS. Queste vengono ruotate automaticamente ogni anno.
4. Scegli l'alias o l'ID chiave di una chiave KMS.
5. Scegliere la scheda Key rotation (Rotazione chiave).

[La scheda Rotazione delle chiavi viene visualizzata solo nella pagina di dettaglio delle chiavi KMS con crittografia simmetrica con il materiale chiave AWS KMS generato \(l'origine è AWS_KMS\), incluse le chiavi KMS di crittografia simmetrica multiregione.](#)

Non è possibile ruotare automaticamente le chiavi KMS asimmetriche, le chiavi KMS HMAC, le chiavi KMS con [materiale della chiave importato](#) o le chiavi KMS negli [archivi delle chiavi personalizzate](#). Tuttavia è possibile [ruotare queste chiavi manualmente](#).

6. Nella sezione Rotazione automatica delle chiavi, scegli Modifica.
7. Per Rotazione dei tasti, selezionate Abilita.

Note

Se una chiave KMS è disabilitata o è in attesa di eliminazione, AWS KMS non ruota il materiale della chiave e non è possibile aggiornare lo stato o il periodo di rotazione automatica delle chiavi. Abilita la chiave KMS o annulla l'eliminazione per aggiornare la configurazione di rotazione automatica delle chiavi. Per informazioni dettagliate, consulta [Come funziona la rotazione dei tasti](#) e [Stati chiave delle AWS KMS chiavi](#).

8. (Facoltativo) Digitate un periodo di rotazione compreso tra 90 e 2560 giorni. Il valore predefinito è 365 giorni. Se non si specifica un periodo di rotazione personalizzato, AWS KMS ruoterà il materiale chiave ogni anno.

È possibile utilizzare la chiave [kms: RotationPeriodInDays](#) condition per limitare i valori che i principali possono specificare per il periodo di rotazione.

9. Scegli Save (Salva).

Utilizzo dell'API AWS KMS

Puoi utilizzare l'[API AWS Key Management Service \(AWS KMS\)](#) per abilitare la rotazione automatica delle chiavi e visualizzare lo stato di rotazione corrente di qualsiasi chiave gestita dal cliente. Questi esempi utilizzano la [AWS Command Line Interface \(AWS CLI\)](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

L'[EnableKeyRotation](#) operazione consente la rotazione automatica delle chiavi per la chiave KMS specificata. Per identificare la chiave KMS in questa operazione, usa il relativo [ID chiave](#) o la [chiave ARN](#). Per impostazione predefinita, la rotazione automatica è disabilitata per le chiavi gestite dal cliente.

È possibile utilizzare la chiave di [kms:RotationPeriodInDays](#) condizione per limitare i valori che i committenti possono specificare per il RotationPeriodInDays parametro di una richiesta.

EnableKeyRotation

L'esempio seguente abilita la rotazione delle chiavi con un periodo di rotazione di 180 giorni sulla chiave KMS di crittografia simmetrica specificata e utilizza l'[GetKeyRotationStatus](#) operazione per visualizzare il risultato.

```
$ aws kms enable-key-rotation \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --rotation-period-in-days 180

$ aws kms get-key-rotation-status --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyRotationEnabled": true,
  "RotationPeriodInDays": 180,
  "NextRotationDate": "2024-02-14T18:14:33.587000+00:00"
}
```

Disabilita la rotazione automatica delle chiavi

Dopo aver abilitato la rotazione automatica delle chiavi su una chiave gestita dal cliente, puoi scegliere di disabilitarla in qualsiasi momento.

Se disabiliti la rotazione automatica dei tasti, la chiave KMS continua a utilizzare la versione del materiale chiave che utilizzava quando la rotazione era disabilitata. Se abiliti nuovamente la rotazione automatica dei tasti, AWS KMS ruota il materiale chiave in base alla nuova data di attivazione della rotazione.

La disabilitazione della rotazione automatica non influisce sulla capacità di [eseguire rotazioni su richiesta, né annulla le rotazioni](#) su richiesta in corso.

È possibile disabilitare la rotazione automatica dei tasti nella AWS KMS console o utilizzando l'operazione. [DisableKeyRotation](#) Per disabilitare la rotazione automatica dei tasti, sono necessarie `kms:DisableKeyRotation` le autorizzazioni. Per ulteriori informazioni sulle AWS KMS autorizzazioni, vedere. [Riferimento per le autorizzazioni](#)

Utilizzo della console AWS KMS

1. Accedi Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente. Non è possibile abilitare o disabilitare la rotazione delle Chiavi gestite da AWS. Queste vengono ruotate automaticamente ogni anno.

4. Scegli l'alias o l'ID chiave di una chiave KMS.
5. Scegliere la scheda Key rotation (Rotazione chiave).

La scheda Rotazione delle chiavi viene visualizzata solo nella pagina di dettaglio delle chiavi KMS con crittografia simmetrica con il materiale chiave AWS KMS generato (l'origine è AWS_KMS), incluse le chiavi KMS di crittografia simmetrica multiregione.

Non è possibile ruotare automaticamente le chiavi KMS asimmetriche, le chiavi KMS HMAC, le chiavi KMS con [materiale della chiave importato](#) o le chiavi KMS negli [archivi delle chiavi personalizzate](#). Tuttavia è possibile [ruotare queste chiavi manualmente](#).

6. Nella sezione Rotazione automatica delle chiavi, scegli Modifica.
7. Per Rotazione dei tasti, selezionate Disabilita.

Note

Se una chiave KMS è disabilitata o è in attesa di eliminazione, AWS KMS non ruota il materiale della chiave e non è possibile aggiornare lo stato o il periodo di rotazione automatica delle chiavi. Abilita la chiave KMS o annulla l'eliminazione per aggiornare la configurazione di rotazione automatica delle chiavi. Per informazioni dettagliate, consulta [Come funziona la rotazione dei tasti](#) e [Stati chiave delle AWS KMS chiavi](#).

8. Scegli Save (Salva).

Utilizzo dell'API AWS KMS

Puoi utilizzare l'[API AWS Key Management Service \(AWS KMS\)](#) per disabilitare la rotazione automatica delle chiavi e visualizzare lo stato di rotazione corrente di qualsiasi chiave gestita dal cliente. Questo esempio utilizza [AWS Command Line Interface \(AWS CLI\)](#), ma è possibile utilizzare qualsiasi linguaggio di programmazione supportato.

L'[DisableKeyRotation](#) operazione disattiva la rotazione automatica dei tasti. Per identificare la chiave KMS in questa operazione, usa il relativo [ID chiave](#) o la [chiave ARN](#). Per impostazione predefinita, la rotazione automatica è disabilitata per le chiavi gestite dal cliente.

L'esempio seguente disattiva la rotazione automatica delle chiavi sulla chiave KMS di crittografia simmetrica specificata e utilizza l'operazione per visualizzare il [GetKeyRotationStatus](#) risultato.

```
$ aws kms disable-key-rotation --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

```
$ aws kms get-key-rotation-status --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyRotationEnabled": false
}
```

Esegui la rotazione dei tasti su richiesta

È possibile eseguire la rotazione su richiesta del materiale chiave nelle chiavi KMS gestite dal cliente, indipendentemente dal fatto che la rotazione automatica delle chiavi sia abilitata o meno. La disabilitazione della rotazione automatica ([DisableKeyRotation](#)) non influisce sulla capacità di eseguire rotazioni su richiesta, né annulla le rotazioni su richiesta in corso. Le rotazioni su richiesta non modificano i programmi di rotazione automatici esistenti. Ad esempio, considera una chiave KMS con rotazione automatica abilitata con un periodo di rotazione di 730 giorni. Se la chiave è programmata per ruotare automaticamente il 14 aprile 2024 e tu esegui una rotazione su richiesta il 10 aprile 2024, la chiave ruoterà automaticamente, come previsto, il 14 aprile 2024 e successivamente ogni 730 giorni.

È possibile eseguire la rotazione dei tasti su richiesta un massimo di 10 volte per chiave KMS. Puoi utilizzare la AWS KMS console per visualizzare il numero di rotazioni su richiesta rimanenti disponibili per una chiave KMS.

La rotazione delle chiavi su richiesta è supportata solo sulle chiavi KMS con crittografia [simmetrica](#). [Non è possibile eseguire la rotazione su richiesta di chiavi KMS asimmetriche, chiavi KMS HMAC o chiavi KMS in un archivio di chiavi personalizzato](#). Per eseguire la rotazione su richiesta di un set di chiavi [multiregionali correlate, richiama la rotazione su richiesta sulla chiave primaria](#).

Gli utenti autorizzati con `kms:RotateKeyOnDemand` e `kms:GetKeyRotationStatus` autorizzazioni possono utilizzare la AWS KMS console e l'AWS KMS API per avviare la rotazione delle chiavi su richiesta e visualizzare lo stato di rotazione delle chiavi. [ListKeyRotations](#) Da utilizzare per visualizzare le rotazioni completate per una chiave KMS.

Argomenti

- [Avvio della rotazione dei tasti su richiesta \(console\)](#)
- [Avvio della rotazione delle chiavi su richiesta \(API\)](#)[AWS KMS](#)

Avvio della rotazione dei tasti su richiesta (console)

1. [Accedi a Console di gestione AWS e apri la console AWS Key Management Service \(AWS KMS\) su `https://console.aws.amazon.com/kms`.](https://console.aws.amazon.com/kms)
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente. (Non è possibile eseguire la rotazione su richiesta di chiavi gestite da AWS. Vengono ruotate automaticamente ogni anno.)
4. Scegli l'alias o l'ID chiave di una chiave KMS.
5. Scegliete la scheda Materiale chiave e rotazioni.

La scheda Materiale chiave e rotazioni viene visualizzata solo nella pagina di dettaglio delle chiavi KMS con crittografia simmetrica che supportano la rotazione automatica o su richiesta. Ciò include le chiavi KMS con materiale chiave AWS KMS generato (AWS_KMSorigine) e le chiavi KMS con materiale chiave importato (origine ESTERNA)

Non è possibile eseguire la rotazione su richiesta di chiavi KMS asimmetriche, chiavi KMS HMAC o chiavi KMS negli archivi di chiavi personalizzati. Tuttavia è possibile ruotare queste chiavi manualmente.

6. Scegli Ruota ora. Per le chiavi di crittografia simmetriche con materiale chiave importato, l'opzione Ruota ora è disponibile solo se in precedenza hai importato nuovo materiale chiave e lo stato di rotazione è in sospeso.

Note

Per le chiavi multiregionali, è possibile ruotare solo la chiave Region principale.

7. Leggi e considera l'avviso e le informazioni sul numero di rotazioni rimanenti su richiesta della chiave. Verranno inoltre visualizzate informazioni come l'ID, la descrizione e l'ora di scadenza del materiale chiave che diventeranno aggiornate dopo la rotazione. Se decidi di non voler procedere con la rotazione su richiesta, scegli Annulla.
8. Scegli il tasto Rotazione per confermare la rotazione su richiesta.

Note

La rotazione su richiesta è soggetta agli stessi eventuali effetti di coerenza delle altre operazioni di gestione. AWS KMS Potrebbe esserci un leggero ritardo prima che il

nuovo materiale chiave sia disponibile in AWS KMS. Il banner nella parte superiore della console ti avvisa quando la rotazione su richiesta è completa.

Avvio della rotazione delle chiavi su richiesta (API)AWS KMS

È possibile utilizzare l'[API AWS Key Management Service \(AWS KMS\)](#) per avviare la rotazione delle chiavi su richiesta e visualizzare lo stato di rotazione corrente di qualsiasi chiave gestita dal cliente. Questo esempio utilizza il [AWS Command Line Interface \(AWS CLI\)](#), ma è possibile utilizzare qualsiasi linguaggio di programmazione supportato.

L'[RotateKeyOnDemand](#) operazione avvia immediatamente la rotazione delle chiavi su richiesta per la chiave KMS specificata. Per identificare la chiave KMS in queste operazioni, utilizza l'[ID chiave](#) o l'[ARN di chiave](#).

L'esempio seguente avvia la rotazione delle chiavi su richiesta sulla chiave KMS di crittografia simmetrica specificata e utilizza l'[GetKeyRotationStatus](#) operazione per verificare che la rotazione su richiesta sia in corso. OnDemandRotationStartDateNella kms:GetKeyRotationStatus risposta identifica la data e l'ora in cui è stata avviata una rotazione su richiesta in corso. In questo esempio, la chiave KMS ha anche la rotazione automatica abilitata con un periodo di 365 giorni.

```
$ aws kms rotate-key-on-demand --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
}

$ aws kms get-key-rotation-status --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyRotationEnabled": true,
  "NextRotationDate": "2024-03-14T18:14:33.587000+00:00",
  "OnDemandRotationStartDate": "2024-02-24T18:44:48.587000+00:00"
  "RotationPeriodInDays": 365
}
```

Se la chiave KMS non supporta la rotazione automatica o non ha la rotazione automatica abilitata, la kms:GetKeyRotationStatus risposta avrebbe meno campi, come mostrato nell'esempio seguente:

```
$ aws kms rotate-key-on-demand --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

```
{
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
}

$ aws kms get-key-rotation-status --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyRotationEnabled": false,
  "OnDemandRotationStartDate": "2024-02-24T18:44:48.587000+00:00"
}
```

Elenca le rotazioni e i materiali chiave

Le chiavi KMS che supportano la rotazione automatica o su richiesta possono avere più materiali chiave associati. Queste chiavi hanno un materiale chiave iniziale e un materiale chiave aggiuntivo per ogni rotazione automatica o su richiesta.

Gli utenti autorizzati con `kms:ListKeyRotations` autorizzazione possono utilizzare la AWS KMS console e l'[ListKeyRotations](#) API per elencare tutti i materiali chiave associati a una chiave KMS, compresi quelli derivanti dalle rotazioni automatiche e su richiesta completate.

Argomenti

- [Elenca le rotazioni e i materiali chiave \(console\)](#)
- [Rotazioni degli elenchi e materiali chiave \(API\)](#) **AWS KMS**

Elenca le rotazioni e i materiali chiave (console)

1. Accedi Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
4. Scegli l'alias o l'ID chiave di una chiave KMS.
5. Scegliete la scheda Materiale chiave e rotazioni.
 - La scheda Materiale chiave e rotazioni viene visualizzata solo nella pagina di dettaglio delle chiavi KMS con crittografia simmetrica che supportano la rotazione automatica o su richiesta.

Ciò include le chiavi KMS con materiale chiave AWS KMS generato (AWS_KMSorigine) e le chiavi KMS con materiale chiave importato (origine). EXTERNAL

- La tabella Materiali chiave nella scheda Materiale chiave e rotazioni elenca tutti i materiali chiave associati alla chiave KMS. Per ogni materiale chiave, la voce corrispondente mostra l'identificatore univoco assegnato da AWS KMS, la data di rotazione e lo stato del materiale chiave. La data di rotazione indica quando il materiale chiave è diventato attuale dopo una rotazione della chiave automatica o su richiesta. Non esiste una data di rotazione associata al primo materiale o al materiale Pending rotation chiave. Lo stato del materiale chiave determina il modo in cui AWS KMS viene utilizzato il materiale chiave. Il materiale chiave corrente viene utilizzato sia per la crittografia che per la decrittografia. Il materiale chiave non corrente viene utilizzato solo per la decrittografia. Lo stato chiave del materiale Pending rotation indica che il materiale chiave è predisposto per la rotazione. Questo materiale chiave non viene utilizzato per alcuna operazione crittografica finché una rotazione della chiave su richiesta non lo rende il materiale chiave corrente. Le informazioni aggiuntive visualizzate per il materiale chiave dipendono dal tipo di chiave KMS.
- Per le chiavi KMS a crittografia simmetrica con AWS_KMS origine, ogni riga mostra anche il tipo di rotazione, ovvero. On-demand Automatic
- Le chiavi KMS di crittografia simmetrica con materiale chiave importato (EXTERNALorigine) supportano solo la On-demand rotazione, quindi non esiste una colonna di tipo di rotazione. Invece, ogni riga mostra uno stato di importazione, una descrizione specificata dall'utente, informazioni sulla scadenza e un menu Azioni. Lo stato di importazione è Importato, a indicare che il materiale chiave è disponibile all'interno, AWS KMS oppure In attesa di importazione, indica che il materiale chiave non è disponibile all'interno. AWS KMS Il menu Azioni può essere utilizzato per eliminare il materiale chiave importato o reimportare il materiale chiave. L'azione Elimina materiale chiave è disabilitata se lo stato di importazione del materiale chiave è In attesa di importazione. L'azione Reimport key material è sempre disponibile. Non è necessario attendere che un materiale chiave scada o venga eliminato prima di reimportarlo.

Rotazioni degli elenchi e materiali chiave (API)AWS KMS

È possibile utilizzare l'[API AWS Key Management Service \(AWS KMS\)](#) per avviare la rotazione delle chiavi su richiesta e visualizzare lo stato di rotazione corrente di qualsiasi chiave gestita dal cliente. Questo esempio utilizza [AWS Command Line Interface \(AWS CLI\)](#), ma è possibile utilizzare qualsiasi linguaggio di programmazione supportato.

L'[ListKeyRotations](#) operazione elenca tutte le rotazioni e i materiali chiave per la chiave KMS specificata. Per identificare la chiave KMS in queste operazioni, utilizza l'[ID chiave](#) o l'[ARN di chiave](#).

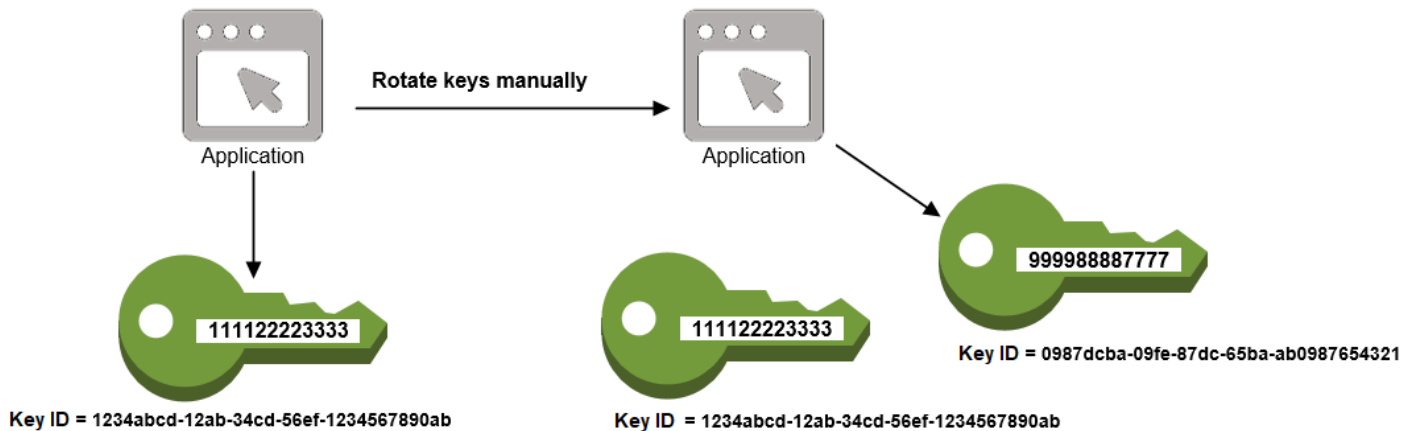
Questa operazione supporta un parametro opzionale `IncludeKeyMaterial`. Il valore predefinito di questo parametro è `ROTATIONS_ONLY`. Se si omette questo parametro, AWS KMS restituisce informazioni sui materiali chiave creati dalla rotazione automatica o su richiesta dei tasti. Quando specificate un valore di `ALL_KEY_MATERIAL`, AWS KMS aggiunge alla risposta il primo materiale chiave e qualsiasi materiale chiave importato in attesa di rotazione. Questo parametro può essere utilizzato solo con chiavi KMS che supportano la rotazione automatica o su richiesta.

```
$ aws kms list-key-rotations --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --include-key-material ALL_KEY_MATERIAL
{
  "Rotations": [
    {
      "KeyId": 1234abcd-12ab-34cd-56ef-1234567890ab,
      "KeyMaterialId":
123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0,
      "KeyMaterialDescription": "KeyMaterialA",
      "ImportState": "PENDING_IMPORT",
      "KeyMaterialState": "NON_CURRENT"
    },
    {
      "KeyId": 1234abcd-12ab-34cd-56ef-1234567890ab,
      "KeyMaterialId":
96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068,
      "ImportState": "IMPORTED",
      "KeyMaterialState": "CURRENT",
      "ExpirationModel": "KEY_MATERIAL_DOES_NOT_EXPIRE",
      "RotationDate": "2025-05-01T15:50:51.045000-07:00",
      "RotationType": "ON_DEMAND"
    }
  ],
  "Truncated": false
}
```

Ruota i tasti manualmente

Potresti voler creare una nuova chiave KMS e utilizzarla al posto di una chiave KMS corrente invece di utilizzare la rotazione automatica o su richiesta. Quando la nuova chiave KMS ha materiale

crittografico diverso rispetto a quello dell'attuale chiave KMS, l'utilizzo della nuova chiave KMS ha lo stesso effetto che si ottiene modificando la chiave di supporto in una chiave KMS esistente. Il processo di sostituzione di una chiave KMS con un'altra è noto come rotazione manuale delle chiavi.



La rotazione manuale è un'ottima scelta quando desideri ruotare chiavi KMS che non sono idonee alla rotazione automatica o su richiesta, come le chiavi KMS asimmetriche, le chiavi KMS HMAC e le chiavi KMS negli archivi di chiavi personalizzati.

Note

Quando inizi a utilizzare la nuova chiave KMS, assicurati di mantenere attiva la chiave KMS originale in modo da poter decrittografare i dati crittografati dalla chiave KMS originale. AWS KMS

Quando ruoti manualmente le chiavi KMS, devi aggiornare anche i riferimenti all'ID o all'ARN della chiave KMS nelle applicazioni. Gli [alias](#), che consentono di associare un nome descrittivo a una chiave KMS, agevolano questo processo. Usa un alias per fare riferimento a una chiave KMS nelle tue applicazioni. Quindi, se desideri modificare la chiave KMS utilizzata dall'applicazione, invece di modificare il codice dell'applicazione, modifica la chiave KMS di destinazione dell'alias. Per informazioni dettagliate, vedi [Scopri come utilizzare gli alias nelle tue applicazioni](#).

Note

Gli alias che rimandano alla versione più recente di una chiave KMS ruotata manualmente sono una buona soluzione per operazioni crittografiche come `Encrypt`, `DescribeKey`, `Sign`, `GetPublicKey` and `DeriveSharedSecretVerify`.

[GenerateDataKeyGenerateDataKeyPairGenerateMacVerifyMac](#) Gli alias non sono consentiti nelle operazioni che gestiscono le chiavi KMS, come o. [DisableKeyScheduleKeyDeletion](#) Quando richiami l'operazione [Decrypt](#) su chiavi KMS di crittografia simmetrica ruotate manualmente, ometti il parametro dal comando. KeyId AWS KMS utilizza automaticamente la chiave KMS che ha crittografato il testo cifrato.

Il KeyId parametro è obbligatorio quando si chiama Decrypt o si [verifica](#) con una chiave KMS asimmetrica o si chiama con una chiave KMS HMAC. [VerifyMac](#) Queste richieste avranno esito negativo se il valore del parametro KeyId è un alias che non punta più alla chiave KMS che ha eseguito l'operazione di crittografia, ad esempio quando una chiave viene ruotata manualmente. Per evitare questo errore, è necessario tenere traccia e specificare la chiave KMS corretta per ciascuna operazione.

Per modificare la chiave KMS di destinazione di un alias, utilizza l'operazione nell'API. [UpdateAlias](#) AWS KMS Ad esempio, questo comando aggiorna l'alias alias/TestKey per puntare a una nuova chiave KMS. Poiché l'operazione non restituisce alcun output, l'esempio utilizza l'[ListAliases](#) operazione per mostrare che l'alias è ora associato a una chiave KMS diversa e il LastUpdatedDate campo è aggiornato. I ListAliases comandi utilizzano il [queryparametro](#) in AWS CLI per ottenere solo l'alias/TestKeyalias.

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/TestKey`]'
{
  "Aliases": [
    {
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/TestKey",
      "AliasName": "alias/TestKey",
      "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "CreationDate": 1521097200.123,
      "LastUpdatedDate": 1521097200.123
    },
  ]
}
```

```
$ aws kms update-alias --alias-name alias/TestKey --target-key-id
0987dcba-09fe-87dc-65ba-ab0987654321
```

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/TestKey`]'
{
  "Aliases": [
```

```
{
  "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/TestKey",
  "AliasName": "alias/TestKey",
  "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
  "CreationDate": 1521097200.123,
  "LastUpdatedDate": 1604958290.722
},
]
```

Cambia la chiave primaria in un set di chiavi multiregionali

Ogni set di chiavi multiregione correlate deve avere una chiave primaria. Ma puoi cambiare la chiave primaria. Questa azione, nota come aggiornamento della Regione principale, converte la chiave primaria corrente in una chiave di replica e converte una delle chiavi di replica correlate nella chiave primaria. È possibile eseguire questa operazione se è necessario eliminare la chiave primaria corrente mantenendo le chiavi di replica o individuare la chiave primaria nella stessa Regione degli amministratori delle chiavi.

È possibile selezionare qualsiasi chiave di replica correlata come nuova chiave primaria. Sia la chiave primaria che la chiave di replica devono avere come [stato della chiave](#) `Enabled` all'avvio dell'operazione.

Lo stato della **Updating** chiave

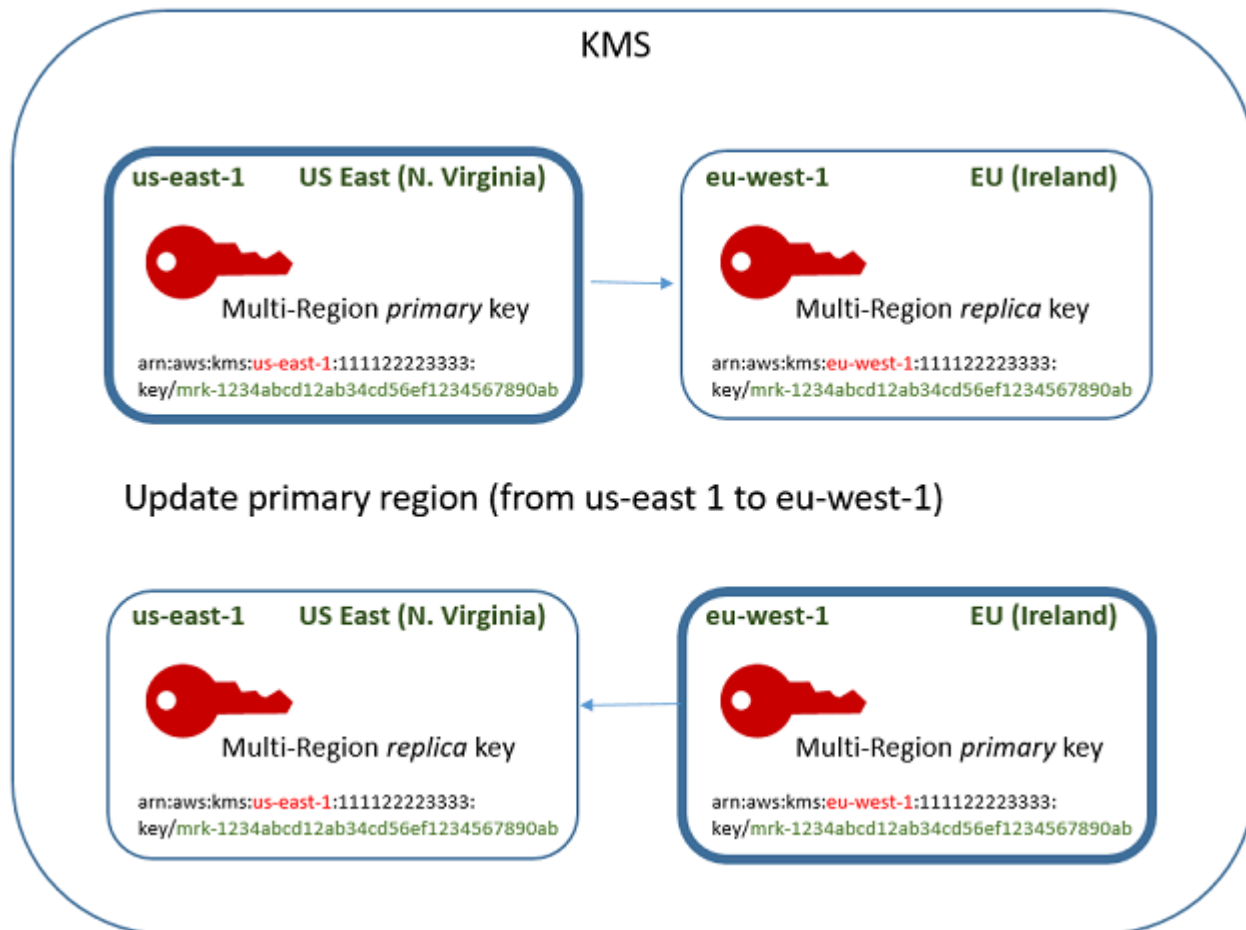
Anche dopo il completamento dell'`UpdatePrimaryRegion` operazione, il processo di aggiornamento della regione principale potrebbe essere ancora in corso per qualche secondo. Durante questo periodo, le chiavi primarie vecchie e nuove hanno uno stato di chiave transitorio [Aggiornamento in corso](#). Quando lo stato della chiave è `Updating`, è possibile utilizzare le chiavi nelle operazioni di crittografia, ma non è possibile replicare la nuova chiave primaria o eseguire determinate operazioni di gestione, ad esempio l'attivazione o la disattivazione di queste chiavi. Operazioni come [DescribeKey](#) potrebbero visualizzare sia la vecchia che la nuova chiave primaria come repliche. Lo stato della chiave `Enabled` viene ripristinato al termine dell'aggiornamento.

Per informazioni sull'effetto dello stato della chiave `Updating`, consulta [Stati chiave delle AWS KMS chiavi](#).

Come funziona

Supponi di avere una chiave primaria negli Stati Uniti orientali (Virginia settentrionale) (`us-east-1`) e una chiave replica in Europa (Irlanda) (`eu-west-1`). È possibile utilizzare la funzionalità di

aggiornamento per modificare la chiave primaria negli Stati Uniti orientali (Virginia settentrionale) (us-east-1) in una chiave di replica e modificare la chiave di replica in Europa (Irlanda) (eu-west-1) nella chiave primaria.



Al termine del processo di aggiornamento, la chiave multiregione nella Regione Europa (Irlanda) (eu-west-1) è una chiave primaria multiregione e la chiave nella Regione Stati Uniti orientali (Virginia) (us-east-1) è la chiave di replica. Se sono presenti altre chiavi di replica correlate, queste diventano repliche della nuova chiave primaria. La prossima volta che AWS KMS sincronizzerà le proprietà condivise delle chiavi multiregionali, otterrà le [proprietà condivise](#) dalla nuova chiave primaria e le copierà nelle relative chiavi di replica, inclusa la precedente chiave primaria.

L'operazione di aggiornamento non ha alcun effetto sull'[ARN della chiave](#) di qualsiasi chiave multiregione. Inoltre, non ha alcun effetto sulle proprietà condivise, come il materiale chiave, o sulle proprietà indipendenti, come la policy chiave. Tuttavia, potresti voler [aggiornare la policy chiave](#) della nuova chiave primaria. Ad esempio, potresti voler aggiungere [kms: ReplicateKey](#) permission for trusted principals alla nuova chiave primaria e rimuoverla dalla nuova chiave di replica.

Aggiorna la regione principale

È possibile convertire una chiave di replica in una chiave primaria, che trasforma la precedente chiave primaria in una replica. Per aggiornare la regione principale, è necessaria l'UpdatePrimaryRegion autorizzazione [kms](#): in entrambe le regioni.

Puoi aggiornare la regione principale nella AWS KMS console o utilizzando l'[UpdatePrimaryRegion](#) operazione.

Utilizzo della AWS KMS console

È possibile aggiornare la chiave primaria nella AWS KMS console. Inizia nella pagina dei dettagli delle chiavi per la chiave primaria corrente.

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
4. Selezionare l'alias o l'ID chiave della [chiave primaria multiregione](#). In questo modo si apre la pagina dei dettagli delle chiavi per la chiave primaria.

Per identificare una chiave primaria multiregione, utilizza l'icona dello strumento nell'angolo in alto a destra per aggiungere la colonna Regionalità nella tabella.

5. Seleziona la tab Regionalità.
6. Nella sezione Chiave primaria, scegli Modifica Regione primaria.
7. Scegliere la Regione della nuova chiave primaria. È possibile scegliere una sola Regione dal menu.

Il menu Modifica Regioni principali include solo le Regioni che dispongono di una chiave multiregione correlata. Potresti non avere l'[autorizzazione per aggiornare la Regione primaria](#) in tutte le Regioni del menu.

8. Scegli Modifica Regione primaria.

Utilizzando l'API AWS KMS

Per modificare la chiave primaria in un set di chiavi multiregionali correlate, utilizzare l'[UpdatePrimaryRegion](#) operazione.

Usa il parametro `KeyId` per identificare la chiave primaria corrente. Utilizzate il `PrimaryRegion` parametro per indicare Regione AWS la nuova chiave primaria. Se la chiave primaria non dispone già di una replica nella nuova Regione primaria, l'operazione ha esito negativo.

Nell'esempio seguente la chiave primaria viene modificata da chiave multiregione nella Regione `us-west-2` a sua replica nella Regione `eu-west-1`. Il parametro `KeyId` identifica la chiave primaria corrente nella Regione `us-west-2`. Il `PrimaryRegion` parametro specifica Regione AWS la nuova chiave primaria, `eu-west-1`.

```
$ aws kms update-primary-region \
    --key-id arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \
    --primary-region eu-west-1
```

In caso di esito positivo, questa operazione non restituisce alcun output; solo il codice di stato HTTP. Per vedere l'effetto, chiamate l'[DescribeKey](#) operazione su uno dei tasti multiregione. Potresti dover attendere fino a quando lo stato della chiave ritorna `Enabled`. Quando lo stato della chiave è [Aggiornamento in corso](#), i valori per la chiave potrebbero essere ancora in flusso.

Ad esempio, la seguente chiamata `DescribeKey` ottiene i dettagli sulla chiave multiregione nella Regione `eu-west-1`. L'output indica che la chiave multiregione nella Regione `eu-west-1` è ora la chiave primaria. La chiave multiregione correlata (stesso ID chiave) nella Regione `us-west-2` è ora una chiave di replica.

```
$ aws kms describe-key \
    --key-id arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \

{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
    "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
    "CreationDate": 1609193147.831,
    "Enabled": true,
    "Description": "multi-region-key",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "Origin": "AWS_KMS",
```

```

    "KeyManager": "CUSTOMER",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "EncryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
    ],
    "MultiRegion": true,
    "MultiRegionConfiguration": {
        "MultiRegionKeyType": "PRIMARY",
        "PrimaryKey": {
            "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
            "Region": "eu-west-1"
        },
        "ReplicaKeys": [
            {
                "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "us-west-2"
            }
        ]
    }
}

```

Eliminare un AWS KMS key

L'eliminazione di un AWS KMS key è distruttiva e potenzialmente pericolosa. Infatti, elimina il materiale della chiave e tutti i metadati associati alla chiave KMS ed è irreversibile. Dopo l'eliminazione di una chiave KMS, non è più possibile decrittare i dati crittografati usando tale chiave KMS, che quindi non possono più essere recuperati. (Le uniche eccezioni sono rappresentate da [chiavi di replica multi-regione](#) e chiavi KMS asimmetriche e HMAC con materiale della chiave importato.) Questo rischio è significativo per le [chiavi KMS asimmetriche utilizzate per la crittografia](#) in cui, senza preavviso o errore, gli utenti possono continuare a generare testi cifrati con la chiave pubblica che non possono essere decrittografati dopo l'eliminazione della chiave privata. AWS KMS

Dovresti eliminare una chiave KMS solo quando hai la certezza di non doverla più utilizzare. In caso di dubbio, valuta la possibilità di [disabilitare la chiave KMS](#) invece di eliminarla. Puoi riabilitare una chiave KMS disabilitata e [annullare l'eliminazione pianificata](#) di una chiave KMS, ma non puoi recuperare una chiave KMS eliminata.

È possibile solamente pianificare l'eliminazione di una chiave gestita dal cliente. Chiavi gestite da AWS Non Chiavi di proprietà di AWS è possibile eliminare o.

Prima di eliminare una chiave KMS, potresti voler sapere quanti testi cifrati sono stati crittografati con quella chiave KMS. AWS KMS non memorizza queste informazioni e non memorizza nessuno dei testi cifrati. Per ottenere queste informazioni, devi determinare l'utilizzo passato di una chiave KMS. Per assistenza, vai a [Determina l'utilizzo passato di una chiave KMS](#).

AWS KMS non elimina mai le tue chiavi KMS a meno che tu non ne pianifichi esplicitamente l'eliminazione e che scada il periodo di attesa obbligatorio.

Potresti scegliere di eliminare una chiave KMS per uno o più dei seguenti motivi:

- Per completare il ciclo di vita delle chiavi per le chiavi KMS che non sono più necessarie
- Per evitare i [costi](#) di gestione delle chiavi KMS inutilizzate
- Per ridurre il numero di chiavi KMS conteggiate nella [quota di risorse delle chiavi KMS](#)

Note

Se [chiudi le tue Account AWS](#) chiavi KMS diventano inaccessibili e non ti vengono più addebitate le spese.

AWS KMS [registra una voce nel AWS CloudTrail registro quando pianifichi l'eliminazione della chiave KMS e quando la chiave KMS viene effettivamente eliminata.](#)

Informazioni sul periodo di attesa

Poiché eliminare una chiave KMS è distruttivo e potenzialmente pericoloso, è AWS KMS necessario impostare un periodo di attesa di 7-30 giorni. Il periodo di attesa predefinito è di 30 giorni.

Tuttavia, il periodo di attesa effettivo potrebbe essere fino a 24 ore più lungo di quello pianificato. Per ottenere la data e l'ora effettive in cui la chiave KMS verrà eliminata, utilizzare l'operazione [DescribeKey](#) O nella console AWS KMS, nella [pagina dei dettagli](#) per la chiave KMS, nella sezione Configurazione generale, consulta Data di eliminazione pianificata. Assicurati di segnare il fuso orario.

Durante il periodo di attesa, lo stato della chiave KMS e quello della chiave è In attesa di eliminazione.

- Una chiave KMS che è in attesa di eliminazione non può essere utilizzata in nessuna [operazione di crittografia](#).
- AWS KMS non [ruota il materiale chiave](#) delle chiavi KMS in attesa di eliminazione.

Al termine del periodo di attesa, AWS KMS elimina la chiave KMS, i relativi alias e tutti i metadati correlati. AWS KMS

La pianificazione dell'eliminazione di una chiave KMS potrebbe non influire immediatamente sulle chiavi di dati crittografate dalla chiave KMS. Per informazioni dettagliate, consultare [In che modo le chiavi KMS inutilizzabili influiscono sulle chiavi dati](#).

Utilizza il periodo di attesa per assicurarti che la chiave KMS non sia necessaria ora o in futuro. Puoi [configurare un CloudWatch allarme Amazon](#) per avvisarti se una persona o un'applicazione tenta di utilizzare la chiave KMS durante il periodo di attesa. Per ripristinare la chiave KMS, puoi annullare l'eliminazione della chiave prima del termine del periodo di attesa. Al termine del periodo di attesa, non è possibile annullare l'eliminazione della chiave ed AWS KMS elimina la chiave KMS.

Considerazioni speciali

Prima di pianificare l'eliminazione delle chiavi, consulta le seguenti considerazioni speciali sull'eliminazione delle chiavi KMS per scopi speciali.

Eliminazione delle chiavi KMS asimmetriche

Gli utenti [autorizzati](#) possono eliminare chiavi KMS simmetriche o asimmetriche. La procedura per pianificare l'eliminazione di queste chiavi KMS è la stessa per entrambi i tipi di chiave. Tuttavia, poiché la [chiave pubblica di una chiave KMS asimmetrica può essere scaricata e utilizzata all'esterno, l'operazione comporta rischi aggiuntivi significativi AWS KMS, in particolare per le chiavi](#) KMS asimmetriche utilizzate per la crittografia (l'utilizzo della chiave è). ENCRYPT_DECRYPT

- Quando pianifichi l'eliminazione di una chiave KMS, lo stato della chiave KMS cambia in In attesa di eliminazione e la chiave KMS non può essere utilizzata nelle [operazioni di crittografia](#). Tuttavia, la pianificazione dell'eliminazione non ha alcun effetto sulle chiavi pubbliche esterne a. AWS KMS Gli utenti che dispongono della chiave pubblica possono continuare a utilizzarle per crittografare i messaggi. Non ricevono alcuna notifica del cambiamento dello stato della chiave. A meno che l'eliminazione non venga annullata, il testo cifrato creato con la chiave pubblica non può essere decrittato.
- Allarmi, log e altre strategie che rilevano il tentativo di utilizzo di chiavi KMS in attesa di eliminazione non possono rilevare l'utilizzo della chiave pubblica al di fuori di AWS KMS.
- Quando la chiave KMS viene eliminata, tutte le AWS KMS azioni che coinvolgono quella chiave KMS hanno esito negativo. Tuttavia, gli utenti che dispongono della chiave pubblica possono continuare a utilizzarla per crittografare i messaggi. Questi testi cifrati non possono essere decrittati.

Se devi eliminare una chiave KMS asimmetrica con un utilizzo di chiave ENCRYPT_DECRYPT, utilizza le voci di CloudTrail registro per determinare se la chiave pubblica è stata scaricata e condivisa. In caso affermativo, verifica che la chiave pubblica non venga utilizzata al di fuori di AWS KMS. Valuta l'opportunità di [disattivare la chiave KMS](#) anziché di eliminarla.

Il rischio rappresentato dall'eliminazione di una chiave asimmetrica è ridotto per le chiavi KMS asimmetriche con materiale della chiave importato. Per informazioni dettagliate, consultare [Deleting KMS keys with imported key material](#).

Eliminazione di chiavi multiregionali

Per eliminare una chiave primaria, è necessario pianificare l'eliminazione di tutte le chiavi di replica e quindi attendere l'eliminazione delle chiavi di replica. Il periodo di attesa richiesto per l'eliminazione di una chiave primaria inizia quando viene eliminata l'ultima delle relative chiavi di replica. Se è necessario eliminare una chiave primaria da una determinata Regione senza eliminarne le chiavi di replica, modificare la chiave primaria in una chiave di replica [aggiornando la Regione principale](#).

Puoi eliminare una chiave di replica in qualsiasi momento. Non dipende dallo stato della chiave di qualsiasi altra chiave KMS. Se si elimina accidentalmente una chiave di replica, è possibile ricrearla replicando la stessa chiave primaria nella stessa regione. La nuova chiave di replica creata avrà le stesse [proprietà condivise](#) della chiave di replica originale.

Eliminazione delle chiavi KMS con materiale chiave importato

L'eliminazione del materiale della chiave di una chiave KMS con il materiale della chiave importato è temporanea e reversibile. Per ripristinare la chiave, reimporta il materiale della chiave.

Al contrario, l'eliminazione di una chiave KMS è irreversibile. Se [pianifichi l'eliminazione delle chiavi](#) e il periodo di attesa richiesto scade, elimina AWS KMS in modo permanente e irreversibile la chiave KMS, il relativo materiale chiave e tutti i metadati associati alla chiave KMS.

Tuttavia, il rischio e le conseguenze dell'eliminazione di una chiave KMS con materiale della chiave importato dipendono dal tipo ("specifica chiave") di chiave KMS.

- Chiavi di crittografia simmetrica: se elimini una chiave KMS di crittografia simmetrica, tutto il testo criptato rimanente crittografati da tale chiave sarà irrecuperabile. Non puoi creare una nuova chiave KMS di crittografia simmetrica in grado di decrittografare i testi criptati di una chiave KMS di crittografia simmetrica eliminata, neppure se disponi dello stesso materiale della chiave. I metadati univoci di ogni chiave KMS sono associati crittograficamente a ogni testo criptato simmetrico. Questa funzionalità di sicurezza garantisce che solo la chiave KMS che ha crittografato il testo criptato simmetrico possa decrittografarlo, ma impedisce di ricreare una chiave KMS equivalente.
- Chiavi asimmetriche e HMAC: se disponi del materiale chiave originale, puoi creare una nuova chiave KMS con le stesse proprietà crittografiche di una chiave KMS asimmetrica o HMAC che è stata eliminata. AWS KMS genera firme e testi cifrati RSA standard, firme ECC e tag HMAC, che non includono funzionalità di sicurezza esclusive. Inoltre, puoi utilizzare una chiave HMAC o la chiave privata di una coppia di chiavi asimmetrica esternamente a AWS.

Una nuova chiave KMS creata con lo stesso materiale della chiave asimmetrica o HMAC avrà un identificatore della chiave diverso. Dovrai creare una nuova policy della chiave, creare nuovamente eventuali alias e aggiornare le concessioni e le policy IAM esistenti in modo che facciano riferimento alla nuova chiave.

Eliminazione delle chiavi KMS da qualsiasi archivio di chiavi AWS CloudHSM

Quando pianifichi l'eliminazione di una chiave KMS da un archivio chiavi, [lo stato della AWS CloudHSM chiave cambia in In sospeso](#) di eliminazione. La chiave KMS rimane nello stato In attesa di eliminazione durante l'intero periodo di attesa, anche se la chiave KMS diventa

indisponibile a seguito della [disconnessione dell'archivio delle chiavi personalizzate](#). Ciò consente di annullare l'eliminazione della chiave KMS in qualsiasi momento durante il periodo di attesa.

Allo scadere del periodo di attesa, AWS KMS elimina la chiave KMS da AWS KMS. Quindi AWS KMS fa del suo meglio per eliminare il materiale chiave dal cluster associato. AWS CloudHSM. Se AWS KMS non riesce a eliminare tale materiale, ad esempio quando lo store delle chiavi personalizzate è disconnesso da AWS KMS, è possibile che tu debba [eliminare manualmente il materiale della chiave orfano](#) dal cluster.

AWS KMS non elimina il materiale chiave dai backup del cluster. Anche se elimini la chiave KMS dal cluster AWS KMS e ne elimini il materiale chiave dal AWS CloudHSM cluster, i cluster creati dai backup potrebbero contenere il materiale chiave eliminato. Per eliminare definitivamente il materiale chiave, utilizza l'[DescribeKey](#) operazione per identificare la data di creazione della chiave KMS. Quindi [elimina tutti i backup del cluster](#) che potrebbero contenere quel materiale.

Quando pianifichi l'eliminazione di una chiave KMS da un archivio chiavi, la AWS CloudHSM chiave KMS diventa immediatamente inutilizzabile (fatta salva l'eventuale coerenza). Tuttavia, le risorse crittografate con [chiavi di dati](#) protette dalla chiave KMS non sono interessate fino a quando la chiave KMS non viene nuovamente utilizzata, ad esempio per decrittografare la chiave dati. Questo problema riguarda Servizi AWS molti dei quali utilizzano chiavi dati per proteggere le risorse. Per informazioni dettagliate, consultare [In che modo le chiavi KMS inutilizzabili influiscono sulle chiavi dati](#).

Eliminazione delle chiavi KMS da un archivio di chiavi esterno

L'eliminazione di una chiave KMS da un archivio delle chiavi esterne non ha alcun effetto sulla [chiave esterna](#) utilizzata come materiale della chiave.

Se pianifichi l'eliminazione di una chiave KMS da un archivio delle chiavi esterne, il relativo [stato chiave](#) diventa Pending deletion (Eliminazione in attesa). La chiave KMS rimane nello stato Pending deletion (Eliminazione in attesa) durante l'intero periodo di attesa, anche se la chiave KMS non è più disponibile dopo la [disconnessione dell'archivio delle chiavi esterne](#). Ciò consente di annullare l'eliminazione della chiave KMS in qualsiasi momento durante il periodo di attesa. Allo scadere del periodo di attesa, AWS KMS elimina la chiave KMS da AWS KMS.

Quando pianifichi l'eliminazione di una chiave KMS da un archivio delle chiavi esterne, la chiave KMS diventa immediatamente inutilizzabile (in base alla coerenza finale). Tuttavia, le risorse crittografate con [chiavi di dati](#) protette dalla chiave KMS non sono interessate fino a quando la chiave KMS non viene nuovamente utilizzata, ad esempio per decrittografare la chiave dati. Questo problema riguarda i Servizi AWS, molti dei quali proteggono le risorse tramite le chiavi

dati. Per informazioni dettagliate, consultare [In che modo le chiavi KMS inutilizzabili influiscono sulle chiavi dati](#).

Controlla l'accesso all'eliminazione delle chiavi

Se utilizzi le policy IAM per consentire le AWS KMS autorizzazioni, le identità IAM che dispongono AWS dell'accesso da amministratore ("Action": "*") o dell'accesso AWS KMS completo ("Action": "kms:*") possono già pianificare e annullare l'eliminazione delle chiavi KMS. Per consentire agli amministratori delle chiavi di pianificare e annullare l'eliminazione delle chiavi nella policy delle chiavi, utilizza la AWS KMS console o l'API. AWS KMS

In genere, solo gli amministratori delle chiavi sono in grado di pianificare o annullare l'eliminazione delle chiavi. Tuttavia, puoi concedere queste autorizzazioni ad altre identità IAM aggiungendo `kms:ScheduleKeyDeletion` e `kms:CancelKeyDeletion` alla policy della chiave o a una policy IAM. Puoi anche utilizzare la chiave [kms:ScheduleKeyDeletionPendingWindowInDays](#) condition per limitare ulteriormente i valori che i principali possono specificare nel `PendingWindowInDays` parametro di una richiesta. [ScheduleKeyDeletion](#)

Consenti agli amministratori delle chiavi di pianificare e annullare l'eliminazione delle chiavi

Utilizzo della console di AWS KMS

Per concedere agli amministratori delle chiavi l'autorizzazione per pianificare e annullare l'eliminazione delle chiavi.

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
4. Scegli l'alias o l'ID chiave della chiave KMS di cui intendi modificare le autorizzazioni.
5. Scegli la scheda Key policy (Policy delle chiavi).
6. Il passaggio successivo è diverso per la visualizzazione predefinita e la visualizzazione della policy della policy delle chiavi. La visualizzazione predefinita è disponibile solo se utilizzi la policy

delle chiavi di console predefinita. In caso contrario, è disponibile solo la visualizzazione della policy.

Quando è disponibile la visualizzazione predefinita, nella scheda Key policy (Policy delle chiavi) viene visualizzato il pulsante Switch to policy view (Passa alla visualizzazione della policy) o Switch to default view (Passa alla visualizzazione predefinita).

- Nella visualizzazione predefinita:
 - In Key deletion (Eliminazione chiave), seleziona Allow key administrators to delete this key (Consenti agli amministratori delle chiavi di eliminare questa chiave).
- Nella visualizzazione della policy:
 - a. Scegli Modifica.
 - b. Nell'istruzione della policy per gli amministratori delle chiavi, aggiungi le autorizzazioni `kms:ScheduleKeyDeletion` e `kms:CancelKeyDeletion` all'elemento Action.

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSKeyAdmin"},
  "Action": [
    "kms:Create*",
    "kms:Describe*",
    "kms:Enable*",
    "kms:List*",
    "kms:Put*",
    "kms:Update*",
    "kms:Revoke*",
    "kms:Disable*",
    "kms:Get*",
    "kms:Delete*",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "*"
}
```

- c. Scegli Save changes (Salva modifiche).

Utilizzando l'API AWS KMS

È possibile utilizzare il AWS Command Line Interface per aggiungere le autorizzazioni per la pianificazione e l'annullamento dell'eliminazione delle chiavi.

Per aggiungere l'autorizzazione per pianificare e annullare l'eliminazione di chiavi

1. Utilizzare il comando [aws kms get-key-policy](#) per recuperare la policy delle chiavi esistente, quindi salvare il documento di policy in un file.
2. Apri il documento della policy nell'editor di testo preferito. Nell'istruzione della policy per gli amministratori delle chiavi, aggiungi le autorizzazioni `kms:ScheduleKeyDeletion` e `kms:CancelKeyDeletion`. L'esempio seguente mostra un'istruzione di policy con queste due autorizzazioni:

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSKeyAdmin"},
  "Action": [
    "kms:Create*",
    "kms:Describe*",
    "kms:Enable*",
    "kms:List*",
    "kms:Put*",
    "kms:Update*",
    "kms:Revoke*",
    "kms:Disable*",
    "kms:Get*",
    "kms>Delete*",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "*"
}
```

3. Utilizza il comando [aws kms put-key-policy](#) per applicare la policy chiave alla chiave KMS.

Pianifica l'eliminazione delle chiavi

Le seguenti procedure descrivono come pianificare l'eliminazione delle chiavi e annullare l'eliminazione delle chiavi AWS KMS keys (chiavi KMS) AWS KMS utilizzando l' AWS KMS API Console di gestione AWS and.

Warning

L'eliminazione di una chiave KMS è un'operazione distruttiva e potenzialmente pericolosa. Dovresti procedere solo quando hai la certezza di non dover più utilizzare la chiave KMS e che non ne avrai bisogno in futuro. In caso di dubbio, dovresti [disabilitare la chiave KMS](#) invece di eliminarla.

Prima di eliminare una chiave KMS, devi disporre delle autorizzazioni necessarie. Per informazioni su come concedere queste autorizzazioni agli amministratori delle chiavi, consulta [Controlla l'accesso all'eliminazione delle chiavi](#). Puoi anche utilizzare la chiave di condizione [kms:ScheduleKeyDeletionPendingWindowInDays](#) per limitare ulteriormente il periodo di attesa, applicando, ad esempio, un periodo di attesa minimo.

AWS KMS registra una voce nel AWS CloudTrail registro quando [pianifichi l'eliminazione](#) della chiave KMS e quando la chiave [KMS viene](#) effettivamente eliminata.

Utilizzo della console di AWS KMS

In Console di gestione AWS, puoi pianificare e annullare l'eliminazione di più chiavi KMS contemporaneamente.

Per pianificare l'eliminazione di chiavi

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) su <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.

Non è possibile pianificare l'eliminazione di [Chiavi gestite da AWS](#) o [Chiavi di proprietà di AWS](#).

4. Scegli la casella di controllo accanto alla chiave KMS che desideri eliminare.

5. Scegliere Key actions (Operazioni sulle chiavi), Schedule key deletion (Pianifica eliminazione chiave).
6. Leggere l'avviso e le informazioni sull'annullamento dell'eliminazione durante il periodo di attesa. Se decidi di annullare l'eliminazione, nella parte inferiore della pagina scegli Annulla.
7. Per Waiting period (in days) (Periodo di attesa (in giorni)), immettere un numero di giorni compreso tra 7 e 30.
8. Controlla le chiavi KMS che stai eliminando.
9. Seleziona la casella di controllo accanto a Conferma che desideri pianificare l'eliminazione di questa chiave in **<number of days>** giorni. .
10. Scegliere Schedule deletion (Pianifica eliminazione).

Lo stato della chiave KMS diventa In attesa di eliminazione.

Utilizzo dell'API AWS KMS

Utilizza il comando [aws kms schedule-key-deletion](#) per pianificare l'eliminazione di una [chiave gestita dal cliente](#), come mostrato nel seguente esempio.

Non è possibile pianificare l'eliminazione di un Chiave gestita da AWS o Chiave di proprietà di AWS.

```
$ aws kms schedule-key-deletion --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --  
pending-window-in-days 10
```

Se utilizzato correttamente, AWS CLI restituisce un output simile a quello mostrato nell'esempio seguente:

```
{  
  "KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "DeletionDate": 1598304792.0,  
  "KeyState": "PendingDeletion",  
  "PendingWindowInDays": 10  
}
```

Annulla l'eliminazione della chiave

Dopo aver [pianificato l'eliminazione di una chiave KMS](#), puoi annullare l'eliminazione della chiave mentre è ancora nello stato di [eliminazione in sospeso](#). È possibile annullare l'eliminazione della

chiave nella AWS KMS console o utilizzando l'[CancelKeyDeletion](#) operazione. Dopo aver annullato l'eliminazione in sospeso di una chiave KMS, lo stato della chiave KMS è `Disabled`. Per ulteriori informazioni sull'attivazione della chiave KMS, consulta [Attivazione e disattivazione delle chiavi](#).

Utilizzo della console AWS KMS

Per annullare l'eliminazione di chiavi

1. Apri la AWS KMS console in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
4. Scegli la casella di controllo accanto alla chiave KMS da ripristinare.
5. Scegliere Key actions (Operazioni sulle chiavi), Cancel key deletion (Annulla eliminazione chiave).

Lo stato della chiave KMS cambia da `In attesa di eliminazione` a `Disabilitata`. Per utilizzare la chiave KMS è necessario [abilitarla](#).

Utilizzando l'API AWS KMS

Utilizzate il [aws kms cancel-key-deletion](#) comando per annullare l'eliminazione delle chiavi da AWS CLI , come illustrato nell'esempio seguente.

```
$ aws kms cancel-key-deletion --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Se utilizzato con successo, AWS CLI restituisce un output simile a quello mostrato nell'esempio seguente:

```
{
  "KeyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Lo stato della chiave KMS cambia da `In attesa di eliminazione` a `Disabilitata`. Per utilizzare la chiave KMS è necessario [abilitarla](#).

Crea un allarme che rileva l'uso di una chiave KMS in attesa di eliminazione

Puoi combinare le funzionalità di AWS CloudTrail Amazon CloudWatch Logs e Amazon Simple Notification Service (Amazon SNS) per creare un CloudWatch allarme Amazon che ti avvisa quando qualcuno nel tuo account tenta di utilizzare una chiave KMS in attesa di eliminazione. Se ricevi questa notifica, potresti voler annullare l'eliminazione della chiave KMS e riconsiderare la decisione di eliminarla.

Le seguenti procedure creano un allarme che ti avvisa ogni volta che il messaggio di errore "**Key ARN** is pending deletion" viene scritto nei tuoi file di registro. CloudTrail Questo messaggio di errore indica che una persona o un'applicazione ha cercato di utilizzare la chiave KMS in una [operazione di crittografia](#). Poiché la notifica è collegata al messaggio di errore, non viene attivata quando utilizzi operazioni API consentite sulle chiavi KMS in attesa di eliminazione, ad esempio ListKeys, CancelKeyDeletion e PutKeyPolicy. Per visualizzare un elenco delle operazioni AWS KMS API che restituiscono questo messaggio di errore, consulta [Stati chiave delle AWS KMS chiavi](#).

L'e-mail di notifica che ricevi non include la chiave KMS o l'operazione di crittografia. Puoi trovare tali informazioni nel [file di registro di CloudTrail](#). L'e-mail segnala invece che lo stato dell'allarme è stato modificato da OK ad Alarm (Allarme). Per ulteriori informazioni sugli CloudWatch allarmi e sui cambiamenti di stato, consulta [Using Amazon CloudWatch alarms](#) nella Amazon CloudWatch User Guide.

Warning

Questo CloudWatch allarme Amazon non è in grado di rilevare l'uso della chiave pubblica di una chiave KMS asimmetrica al di fuori di. AWS KMS Per informazioni dettagliate sui rischi particolari derivanti dall'eliminazione delle chiavi KMS asimmetriche utilizzate per la crittografia a chiave pubblica, inclusa la creazione di testi cifrati che non possono essere decrittati, consulta [Deleting asymmetric KMS keys](#).

In questa procedura, crei un filtro metrico del gruppo di CloudWatch log che trova le istanze dell'eccezione di eliminazione in sospeso. Quindi, si crea un CloudWatch allarme basato sulla metrica del gruppo di log. Per informazioni sui filtri delle metriche dei gruppi di log, consulta [Creazione di metriche da eventi di log utilizzando filtri](#) nella Amazon CloudWatch Logs User Guide.

1. Crea un filtro CloudWatch metrico che analizzi i log. CloudTrail

Segui le istruzioni in [Creazione di un filtro di parametri per un gruppo di log](#) utilizzando i seguenti valori obbligatori. Per gli altri campi, accetta i valori predefiniti e immetti i nomi come richiesto.

Campo	Valore
Modello di filtro	<code>{ \$.eventSource = kms* && \$.errorMessage = "* is pending deletion." }</code>
Valore del parametro	1

2. Crea un CloudWatch allarme basato sul filtro metrico creato nel passaggio 1.

Segui le istruzioni riportate in [Creare un CloudWatch allarme basato su un filtro metrico di gruppo di log](#) utilizzando i seguenti valori obbligatori. Per gli altri campi, accetta i valori predefiniti e immetti i nomi come richiesto.

Campo	Valore
Filtro di parametri	Il nome del filtro di parametri che hai creato nella fase 1.
Tipo di soglia	Statico
Condizioni	Ogni volta <i>metric-name</i> è maggiore/uguale a 1
Punti dati da segnalare	1 di 1
Trattamento dei dati mancanti	Considera dati mancanti come buoni (non superano la soglia)

Dopo aver completato questa procedura, riceverai una notifica ogni volta che il nuovo CloudWatch allarme entra nello ALARM stato. Se ricevi una notifica per questo allarme, per crittografare o decrittografare i dati è possibile che sia ancora necessaria una chiave KMS pianificata per

l'eliminazione. In questo caso, [annulla l'eliminazione della chiave KMS](#) e riconsidera la decisione di eliminarla.

Determina l'utilizzo passato di una chiave KMS

Prima di eliminare una chiave KMS, potresti voler sapere quanti testi cifrati sono stati crittografati con quella chiave. AWS KMS non memorizza queste informazioni e non memorizza nessuno dei testi cifrati. Sapere in che modo una chiave KMS è stata utilizzata in passato potrebbe aiutarti a decidere se ti servirà in futuro. In questo argomento vengono suggerite diverse strategie che consentono di determinare l'utilizzo passato di una chiave KMS.

Warning

Queste strategie per determinare l'utilizzo passato ed effettivo sono efficaci solo per AWS gli utenti e AWS KMS le operazioni. Non sono in grado di rilevare l'uso della chiave pubblica di una asimmetrica al di fuori di AWS KMS. Per informazioni dettagliate sui rischi particolari derivanti dall'eliminazione delle chiavi KMS asimmetriche utilizzate per la crittografia a chiave pubblica, inclusa la creazione di testi cifrati che non possono essere decrittati, consulta [Deleting asymmetric KMS keys](#).

Argomenti

- [Esamina le autorizzazioni delle chiavi KMS per determinare l'ambito del potenziale utilizzo](#)
- [Esamina AWS CloudTrail i log per determinare l'utilizzo effettivo](#)

Esamina le autorizzazioni delle chiavi KMS per determinare l'ambito del potenziale utilizzo

Stabilire chi o cosa ha attualmente accesso a una chiave KMS potrebbe aiutarti a determinare in quale misura è stata utilizzata la chiave KMS e se è ancora necessaria. Per ulteriori informazioni su come determinare chi o cosa ha attualmente accesso a una chiave KMS, consulta [Determinare l'accesso a AWS KMS keys](#).

Esamina AWS CloudTrail i log per determinare l'utilizzo effettivo

Potresti voler utilizzare la cronologia di utilizzo di una chiave KMS per stabilire se in una determinata chiave KMS sono crittografati i testi cifrati.

Tutte le attività dell' AWS KMS API vengono registrate nei file di AWS CloudTrail registro. Se hai [creato un CloudTrail percorso](#) nella regione in cui si trova la tua chiave KMS, puoi esaminare i file di CloudTrail registro per visualizzare una cronologia di tutte le attività AWS KMS API per una particolare chiave KMS. Se un trail non è disponibile, è comunque possibile visualizzare gli eventi recenti nella [cronologia degli eventi CloudTrail](#) . Per informazioni dettagliate sulle modalità di AWS KMS utilizzo CloudTrail, consulta. [Registrazione delle chiamate AWS KMS API con AWS CloudTrail](#)

Gli esempi seguenti mostrano le voci di CloudTrail registro generate quando viene utilizzata una chiave KMS per proteggere un oggetto archiviato in Amazon Simple Storage Service (Amazon S3). In questo esempio, l'oggetto viene caricato in Amazon S3 utilizzando le informazioni riportate in [Protezione dei dati utilizzando la crittografia lato server con chiavi KMS \(SSE-KMS\)](#). Quando carichi un oggetto in Amazon S3 con SSE-KMS, specifichi la chiave KMS da utilizzare per proteggere l'oggetto. Amazon S3 utilizza l' AWS KMS [GenerateDataKey](#) operazione per richiedere una chiave dati univoca per l'oggetto e questo evento di richiesta viene registrato CloudTrail con una voce simile alla seguente:

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AR0ACKCEVSQ6C2EXAMPLE:example-user",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admins/example-user",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2015-09-10T23:12:48Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AR0ACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/Admins",
        "accountId": "111122223333",
        "userName": "Admins"
      }
    },
    "invokedBy": "internal.amazonaws.com"
  },
  "eventTime": "2015-09-10T23:58:18Z",
  "eventSource": "kms.amazonaws.com",
```

```

"eventName": "GenerateDataKey",
"awsRegion": "us-west-2",
"sourceIPAddress": "internal.amazonaws.com",
"userAgent": "internal.amazonaws.com",
"requestParameters": {
  "encryptionContext": {"aws:s3:arn": "arn:aws:s3:::example_bucket/example_object"},
  "keySpec": "AES_256",
  "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
},
"responseElements": null,
"requestID": "cea04450-5817-11e5-85aa-97ce46071236",
"eventID": "80721262-21a5-49b9-8b63-28740e7ce9c9",
"readOnly": true,
"resources": [{
  "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "accountId": "111122223333"
}],
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}

```

Quando successivamente scarichi questo oggetto da Amazon S3, Amazon S3 invia Decrypt una richiesta AWS KMS per decrittografare la chiave dati dell'oggetto utilizzando la chiave KMS specificata. Quando esegui questa operazione, i tuoi file di CloudTrail log includono una voce simile alla seguente:

```

{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROACKCEVSQ6C2EXAMPLE:example-user",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admins/example-user",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2015-09-10T23:12:48Z"
      },
      "sessionIssuer": {
        "type": "Role",

```

```

    "principalId": "AROACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::111122223333:role/Admins",
    "accountId": "111122223333",
    "userName": "Admins"
  }
},
  "invokedBy": "internal.amazonaws.com"
},
"eventTime": "2015-09-10T23:58:39Z",
"eventSource": "kms.amazonaws.com",
"eventName": "Decrypt",
"awsRegion": "us-west-2",
"sourceIPAddress": "internal.amazonaws.com",
"userAgent": "internal.amazonaws.com",
"requestParameters": {
  "encryptionContext": {"aws:s3:arn": "arn:aws:s3:::example_bucket/example_object"}},
"responseElements": null,
"requestID": "db750745-5817-11e5-93a6-5b87e27d91a0",
"eventID": "ae551b19-8a09-4cfc-a249-205ddba330e3",
"readOnly": true,
"resources": [{
  "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "accountId": "111122223333"
}],
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}

```

Tutte le attività dell' AWS KMS API vengono registrate da CloudTrail. Esaminando queste voci di registro, potresti essere in grado di determinare l'utilizzo passato di una chiave KMS specifica e ciò potrebbe aiutarti a decidere se eliminarla o meno.

Per vedere altri esempi di come l'attività delle AWS KMS API viene visualizzata nei file di CloudTrail registro, vai a [Registrazione delle chiamate AWS KMS API con AWS CloudTrail](#). Per ulteriori informazioni su questo argomento, CloudTrail consulta la [Guida per AWS CloudTrail l'utente](#).

Eliminare il materiale chiave importato

È possibile eliminare il materiale chiave importato da una chiave KMS in qualsiasi momento. Inoltre, quando il materiale chiave importato con una data di scadenza scade, AWS KMS elimina il materiale chiave. In entrambi i casi, quando il materiale chiave viene eliminato, [lo stato della chiave](#) KMS

cambia in In attesa di importazione e la chiave KMS non può essere utilizzata in alcuna operazione crittografica.

Alle chiavi di crittografia simmetriche possono essere associati più materiali chiave e l'eliminazione o la scadenza di qualsiasi materiale chiave in uno stato diverso da quello di *PENDING_ROTATION* modifica lo stato della chiave in In attesa di importazione. Per queste chiavi, KMS assegna un identificatore univoco a ciascun materiale chiave. Puoi utilizzare l'[ListKeyRotations](#) API per visualizzare questi identificatori di materiali chiave. È possibile eliminare un materiale chiave specifico specificandone l'identificatore utilizzando il `key-material-id` parametro nell'API. [DeleteImportedKeyMaterial](#)

Considerazioni sulle chiavi multiregionali

- Quando elimini il materiale chiave di una chiave regionale principale che si trova nel nostro *PENDING_ROTATION* *PENDING_MULTI_REGION_IMPORT_AND_ROTATION* stato, elimini anche il materiale chiave per le chiavi regionali di replica.
- Se elimini il materiale chiave di una chiave di regione di replica, la chiave di regione principale e le altre chiavi di regione di replica rimangono invariate.

Warning

Il `key-material-id` parametro è facoltativo e, se non lo specificate, AWS KMS eliminerà il materiale chiave corrente.

Oltre alla disabilitazione della chiave KMS e alla revoca delle autorizzazioni, l'eliminazione del materiale della chiave può essere utilizzata come strategia per interrompere rapidamente, ma temporaneamente, l'uso della chiave KMS. Al contrario, la pianificazione dell'eliminazione di una chiave KMS con il materiale della chiave importato interrompe rapidamente anche l'uso della chiave KMS. Tuttavia, se l'eliminazione non viene annullata durante il periodo di attesa, la chiave KMS, i materiali chiave associati e tutti i metadati chiave vengono eliminati definitivamente. Per informazioni dettagliate, vedi [Deleting KMS keys with imported key material](#).

Per eliminare il materiale chiave, puoi utilizzare la AWS KMS console o l'operazione API.

[DeleteImportedKeyMaterial](#) AWS KMS registra una voce nel AWS CloudTrail registro quando si [elimina materiale chiave importato e quando si AWS KMS elimina materiale chiave scaduto](#).

In che modo l'eliminazione di materiale chiave influisce sui servizi AWS

Quando elimini un materiale chiave, la chiave KMS diventa immediatamente inutilizzabile (fatta salva l'eventuale coerenza). Tuttavia, le risorse crittografate con [chiavi di dati](#) protette dalla chiave KMS non sono interessate fino a quando la chiave KMS non viene nuovamente utilizzata, ad esempio per decrittografare la chiave dati. Questo problema riguarda Servizi AWS molti dei quali utilizzano chiavi dati per proteggere le risorse. Per informazioni dettagliate, vedi [In che modo le chiavi KMS inutilizzabili influiscono sulle chiavi dati](#).

Utilizzo della AWS KMS console

È possibile utilizzare la AWS KMS console per eliminare il materiale chiave.

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
4. Esegui una di queste operazioni:
 - Seleziona la casella di controllo di una chiave KMS con materiale chiave importato. Scegliere Key actions (Operazioni della chiave), Delete key material (Elimina materiale della chiave). Per le chiavi di crittografia simmetriche a cui sono associati più materiali chiave, verrà eliminato il materiale chiave corrente.
 - Per le chiavi KMS con crittografia simmetrica con materiale chiave importato, scegli l'alias o l'ID della chiave KMS. Scegli la scheda Materiale chiave e rotazioni. La tabella dei materiali chiave elencherà tutti i materiali chiave associati alla chiave. Scegliete Elimina materiale chiave dal menu Azioni nella riga corrispondente al materiale chiave che desiderate eliminare.
5. Confermare che si intende eliminare il materiale della chiave, quindi selezionare Delete key material (Elimina materiale chiave). Lo stato della chiave KMS, che corrisponde al relativo [stato di chiave](#), diventa In attesa di importazione. Se il materiale chiave eliminato era inalterato, PENDING_ROTATION lo stato della chiave KMS non viene modificato.

Utilizzo dell'API AWS KMS

Per utilizzare l'[AWS KMS API](#) per eliminare il materiale chiave, invia una [DeleteImportedKeyMaterial](#) richiesta. L'esempio seguente mostra come eseguire questa operazione con l'[AWS CLI](#).

Sostituisci *1234abcd-12ab-34cd-56ef-1234567890ab* con l'ID chiave della chiave KMS il cui materiale desideri eliminare. Puoi utilizzare l'ID chiave o l'ARN della chiave KMS, mentre non puoi utilizzare un alias per questa operazione. Il comando seguente elimina il materiale chiave corrente, che può essere l'unico materiale chiave associato alla chiave.

```
$ aws kms delete-imported-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Per eliminare un materiale chiave specifico, specificate il materiale chiave identificato utilizzando il `key-material-id` parametro. Sostituiscilo *123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0* con l'identificatore del materiale chiave che desideri eliminare.

```
$ aws kms delete-imported-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
--key-material-id 123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0
```

Generazione di chiavi dati

Le chiavi di dati sono chiavi simmetriche che possono essere usate per crittografare i dati, incluse grandi quantità di dati e altre chiavi crittografiche dati. A differenza delle chiavi KMS simmetriche, che non possono essere scaricate, le chiavi dati ti vengono restituite per essere utilizzate all'esterno di AWS KMS.

Quando AWS KMS genera chiavi dati, restituisce una chiave dati in testo semplice per l'uso immediato (opzionale) e una copia crittografata della chiave dati che puoi archiviare in sicurezza con i dati. Quando sei pronto per decrittografare i dati, chiedi innanzitutto di AWS KMS decrittografare la chiave dati crittografata.

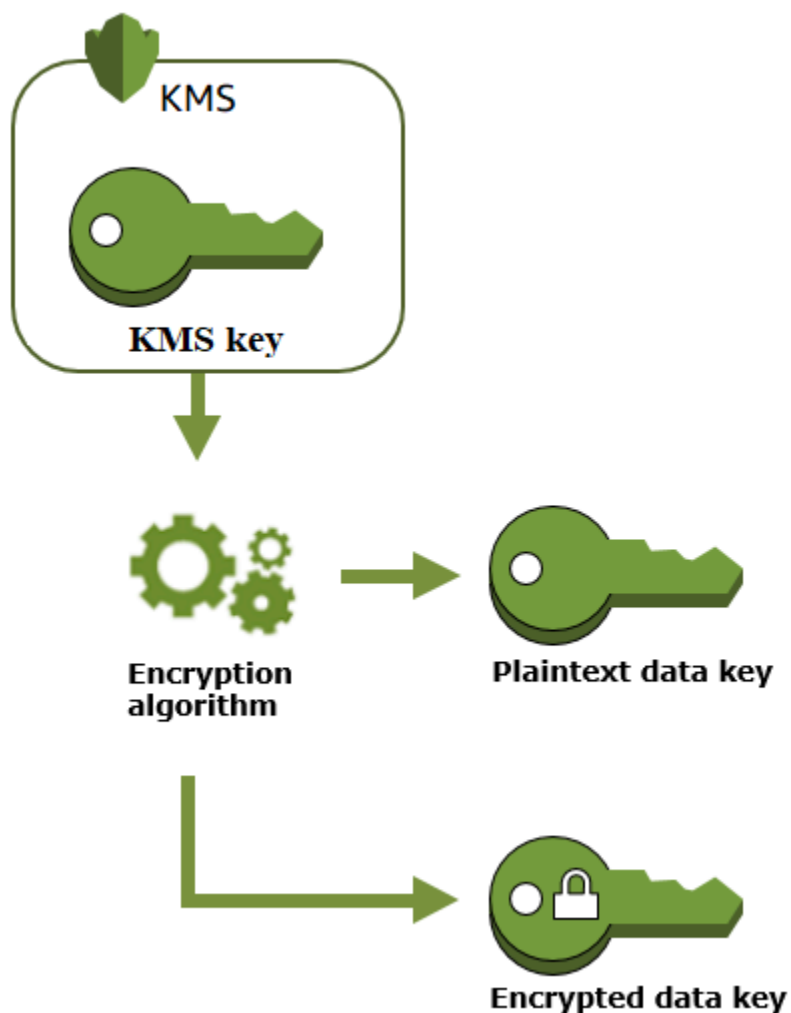
AWS KMS genera, crittografa e decrittografa le chiavi di dati. Tuttavia, AWS KMS non archivia, gestisce o tiene traccia delle chiavi dati né esegue operazioni crittografiche con le chiavi dati. È necessario utilizzare e gestire le chiavi dati all'esterno di AWS KMS. Per informazioni sull'utilizzo sicuro delle chiavi dati, consulta [AWS Encryption SDK](#).

Argomenti

- [Crea una chiave di chiavi](#)
- [Come funzionano le operazioni crittografiche con chiavi dati](#)
- [In che modo le chiavi KMS inutilizzabili influiscono sulle chiavi dati](#)

Crea una chiave di chiavi

Per creare una chiave dati, richiama l'[GenerateDataKey](#) operazione. AWS KMS genera la chiave dati. Quindi crittografa una copia della chiave di dati con una [chiave KMS di crittografia simmetrica](#) da te specificata. L'operazione restituisce una copia in testo normale e un'altra copia della chiave dati crittografata con la chiave KMS. L'immagine seguente mostra questa operazione.



AWS KMS supporta anche l'[GenerateDataKeyWithoutPlaintext](#) operazione, che restituisce solo una chiave dati crittografata. Quando devi usare la chiave dati, chiedi di AWS KMS [decrittografarla](#).

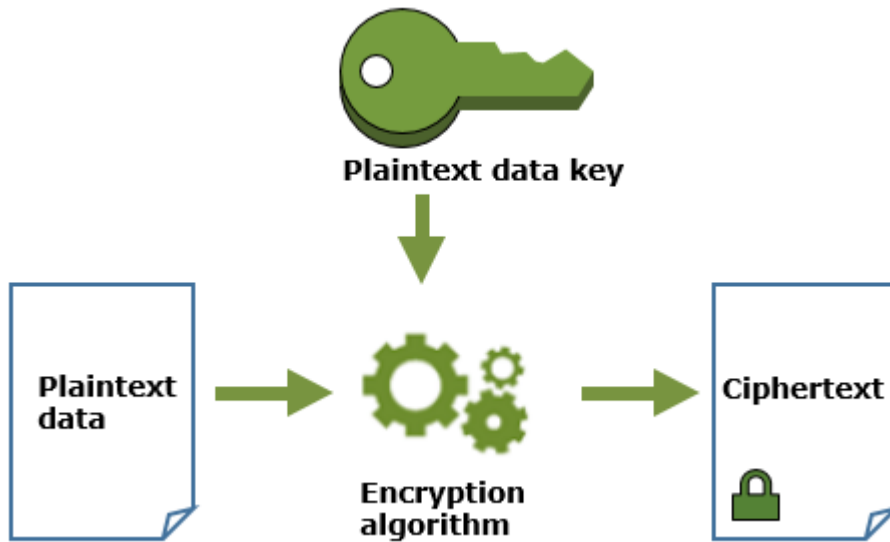
Come funzionano le operazioni crittografiche con chiavi dati

I seguenti argomenti spiegano come funzionano le chiavi dati generate da un'[GenerateDataKeyWithoutPlaintext](#) operazione [GenerateDataKey](#) o.

Crittografia dei dati con una chiave di dati

AWS KMS non può utilizzare una chiave dati per crittografare i dati. Ma puoi usare la chiave dati all'esterno AWS KMS, ad esempio usando OpenSSL o una libreria crittografica come la. [AWS Encryption SDK](#)

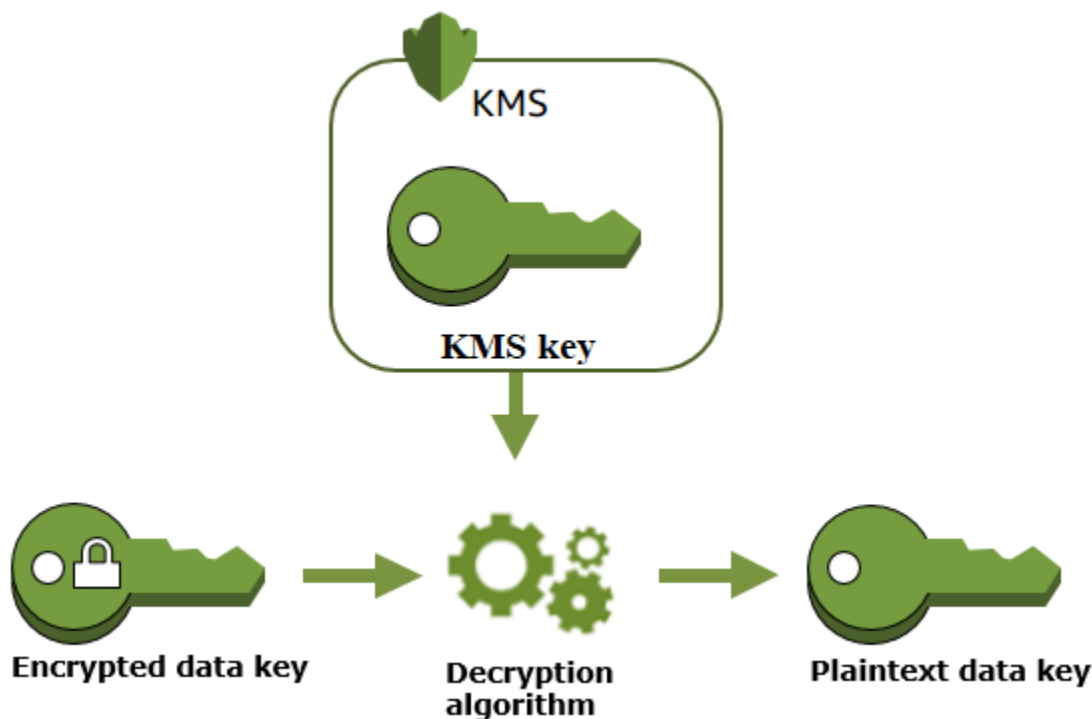
Dopo aver utilizzato la chiave di dati in testo normale per crittografare i dati, eliminarla dalla memoria il prima possibile. È possibile archiviare la chiave di dati crittografati con i dati crittografati in totale sicurezza, in modo che sia disponibile per decrittografare i dati.



Decrittografare i dati con una chiave di dati

[Per decrittografare i dati, passa la chiave dei dati crittografati all'operazione Decrypt.](#) AWS KMS utilizza la chiave KMS per decrittografare la chiave dati e quindi restituisce la chiave dati in testo semplice. Utilizza la chiave di dati in testo normale per decrittografare i dati, quindi rimuovi la chiave di dati in testo normale dalla memoria al più presto.

Il seguente diagramma illustra come utilizzare l'operazione Decrypt per decrittografare una chiave di dati crittografati.



In che modo le chiavi KMS inutilizzabili influiscono sulle chiavi dati

Quando una chiave KMS diventa inutilizzabile, l'effetto è quasi immediato (in base alla coerenza finale). Lo [stato](#) della chiave KMS si modifica per riflettere la nuova condizione e tutte le richieste di utilizzo della chiave KMS nelle [operazioni di crittografia](#) hanno esito negativo.

Tuttavia, l'effetto sulle chiavi di dati crittografate dalla chiave KMS e sui dati crittografati dalla chiave dati viene ritardato fino a quando la chiave KMS non viene nuovamente utilizzata, ad esempio per decrittografare la chiave dati.

Le chiavi KMS possono diventare inutilizzabili per diversi motivi, incluse le azioni seguenti che è possibile eseguire.

- [Disattivazione della chiave KMS](#)
- [Pianificazione dell'eliminazione della chiave KMS](#)
- [Eliminazione del materiale della chiave](#) da una chiave KMS con materiale della chiave importato o scadenza del materiale della chiave importato. Se a una chiave KMS con EXTERNAL origine sono associati più materiali chiave, l'eliminazione o la scadenza di qualsiasi materiale chiave renderà la chiave inutilizzabile.
- [Disconnessione dell'archivio AWS CloudHSM chiavi](#) che ospita la chiave KMS o [eliminazione della chiave dal AWS CloudHSM cluster che funge da materiale chiave per la](#) chiave KMS.

- [Disconnessione dell'archivio delle chiavi esterne](#) che ospita la chiave KMS o qualsiasi altra azione che interferisca con le richieste di crittografia e decrittografia al proxy, inclusa l'eliminazione della chiave esterna dal relativo gestore delle chiavi esterne.

Questo effetto è particolarmente importante per i molti Servizi AWS che utilizzano le chiavi di dati per proteggere le risorse gestite dal servizio. L'esempio seguente utilizza Amazon Elastic Block Store (Amazon EBS) e Amazon Elastic Compute Cloud EC2 (Amazon). Le chiavi dati vengono Servizi AWS utilizzate in modi diversi. Per maggiori dettagli, consulta la sezione Protezione dei dati del capitolo Sicurezza per il Servizio AWS.

Considera ad esempio questo scenario:

1. Puoi [creare un volume EBS crittografato](#) e specificare una chiave KMS per proteggerlo. Amazon EBS chiede a AWS KMS di utilizzare la chiave KMS per [generare una chiave dati crittografata](#) per il volume. Amazon EBS archivia la chiave dati crittografata con i metadati del volume.
2. Quando colleghi il volume EBS a un' EC2 istanza, Amazon EC2 usa la tua chiave KMS per decrittografare la chiave dati crittografata del volume EBS. Amazon EC2 utilizza la chiave dati nell'hardware Nitro, che è responsabile della crittografia di tutto il disco sul volume I/O EBS. La chiave dati persiste nell'hardware Nitro mentre il volume EBS è collegato all'istanza. EC2
3. Esegui un'operazione in grado di rendere la chiave KMS inutilizzabile. Ciò non ha alcun effetto immediato sull' EC2 istanza o sul volume EBS. Amazon EC2 utilizza la chiave dati, non la chiave KMS, per crittografare tutto il disco I/O mentre il volume è collegato all'istanza.
4. Tuttavia, quando il volume EBS crittografato viene scollegato dall' EC2 istanza, Amazon EBS rimuove la chiave dati dall'hardware Nitro. La prossima volta che il volume EBS crittografato viene collegato a un' EC2 istanza, l'allegato non riesce perché Amazon EBS non può utilizzare la chiave KMS per decrittografare la chiave dati crittografata del volume. Per utilizzare di nuovo il volume EBS, devi rendere utilizzabile la chiave KMS.

Genera coppie di chiavi di dati

Una chiave KMS asimmetrica rappresenta una coppia di chiavi di dati. Le coppie di chiavi di dati sono chiavi di dati asimmetriche costituite da una chiave pubblica e una chiave privata correlate matematicamente. Sono progettati per l'uso nella crittografia e decrittografia lato client, nella firma e nella verifica all'esterno o per stabilire un segreto condiviso tra AWS KMS due peer.

A differenza delle coppie di chiavi di dati generate da strumenti come OpenSSL AWS KMS , protegge la chiave privata in ogni coppia di chiavi dati con una chiave KMS a crittografia simmetrica specificata dall'utente. AWS KMS Tuttavia, AWS KMS non archivia, gestisce o tiene traccia delle coppie di chiavi di dati né esegue operazioni crittografiche con coppie di chiavi di dati. È necessario utilizzare e gestire le coppie di chiavi di dati al di fuori di AWS KMS.

Argomenti

- [Creare una coppia di chiave di dati](#)
- [Come funzionano le operazioni crittografiche con coppie di chiavi di dati](#)

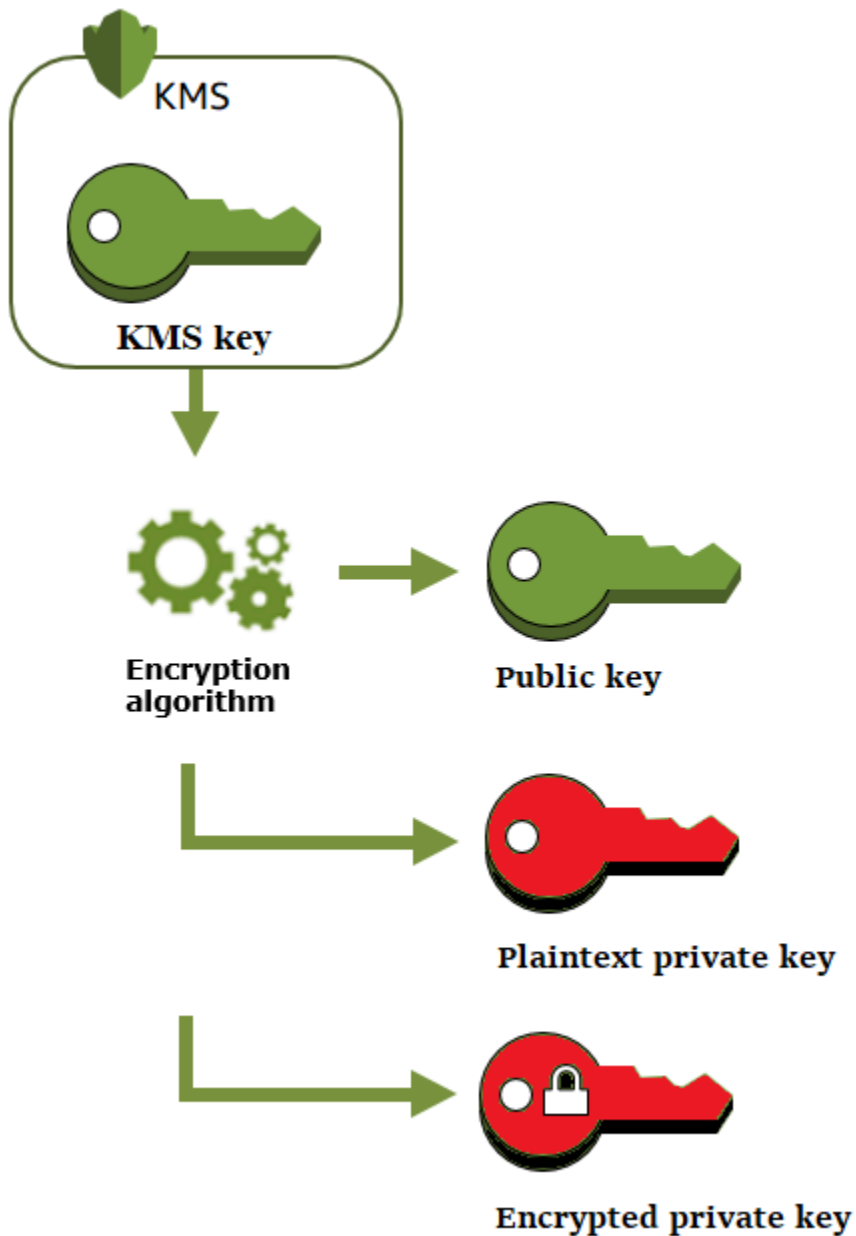
Creare una coppia di chiave di dati

Per creare una coppia di chiavi di dati, chiama le [GenerateDataKeyPairWithoutPlaintext](#) operazioni [GenerateDataKeyPair](#). Specifica la [chiave KMS di crittografia simmetrica](#) che desideri utilizzare per crittografare la chiave privata.

`GenerateDataKeyPair` restituisce una chiave pubblica di testo normale, una chiave privata di testo normale e una chiave privata crittografata. Utilizza questa operazione quando è necessaria una chiave privata di testo normale immediatamente, ad esempio per generare una firma digitale.

`GenerateDataKeyPairWithoutPlaintext` restituisce una chiave pubblica di testo normale e una chiave privata crittografata, ma non una chiave privata di testo normale. Utilizza questa operazione quando non è necessaria una chiave privata di testo normale, ad esempio quando si esegue la crittografia con una chiave pubblica. Successivamente, quando è necessaria una chiave privata di testo normale per decriptare i dati, è possibile chiamare l'operazione [Decrittografa](#).

L'immagine seguente mostra l'operazione `GenerateDataKeyPair`. L'operazione `GenerateDataKeyPairWithoutPlaintext` omette la chiave privata di testo normale.



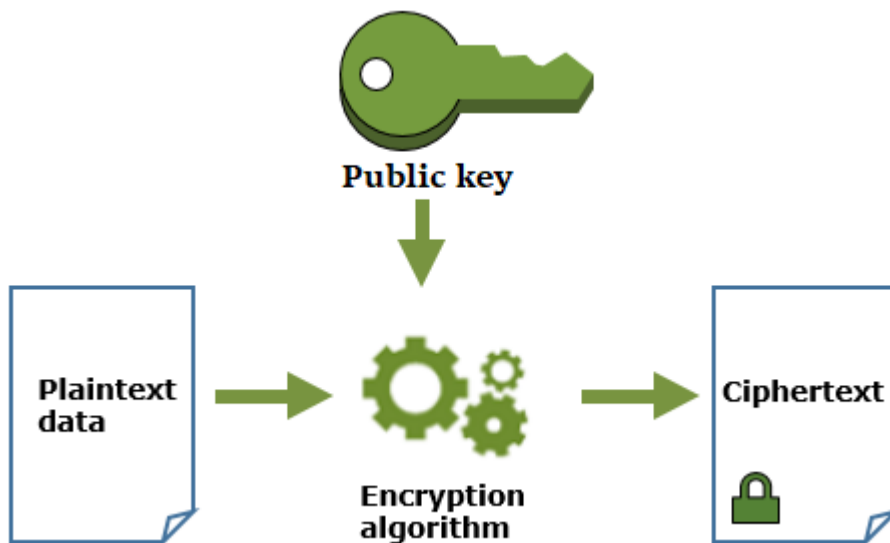
Come funzionano le operazioni crittografiche con coppie di chiavi di dati

I seguenti argomenti spiegano quali operazioni crittografiche è possibile eseguire con le coppie di chiavi di dati generate da un'[GenerateDataKeyPairWithoutPlaintext](#) operazione [GenerateDataKeyPair](#) o e come funzionano.

Crittografia dei dati con una coppia di chiavi di dati

Quando si esegue la crittografia con una coppia di chiavi di dati, si utilizza la chiave pubblica della coppia per crittografare i dati e la chiave privata della stessa coppia per decriptare i dati. In genere, usi le coppie di chiavi di dati quando molte parti devono crittografare i dati che solo la parte con la chiave privata può decriptare.

Le parti con la chiave pubblica utilizzano tale chiave per crittografare i dati, come mostrato nel diagramma seguente.

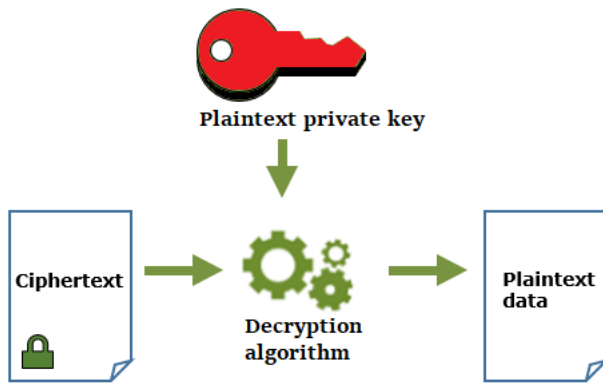


Decrittografia dei dati con una coppia di chiave di dati

Per decriptare i dati, utilizzare la chiave privata nella coppia di chiavi di dati. Affinché l'operazione abbia esito positivo, le chiavi pubbliche e private devono essere della stessa coppia di chiavi di dati ed è necessario utilizzare lo stesso algoritmo di crittografia.

Per decriptare la chiave privata crittografata, passarla all'operazione [Decrittografa](#). Utilizza la chiave privata di testo normale per decriptare i dati. Quindi rimuovi la chiave privata di testo normale dalla memoria il prima possibile.

Il diagramma seguente mostra come utilizzare la chiave privata in una coppia di chiavi di dati per decriptare il testo cifrato.



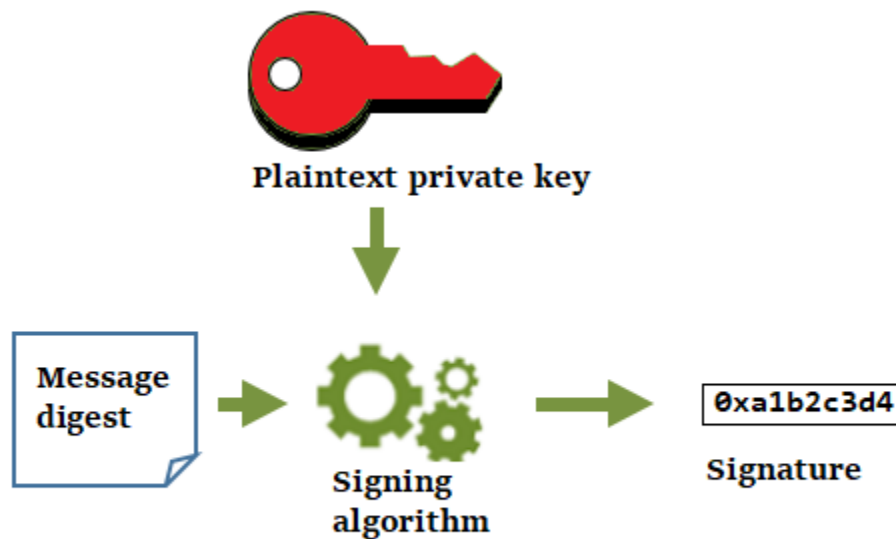
Firmare messaggi con una coppia di chiavi di dati

Per generare una firma crittografica per un messaggio, utilizzare la chiave privata nella coppia di chiavi di dati. Chiunque abbia la chiave pubblica può utilizzarla per verificare che il messaggio sia stato firmato con la chiave privata e che non sia cambiato da quando è stato firmato.

Se crittografi la tua chiave privata, passa la chiave privata crittografata all'operazione [Decrypt](#). AWS KMS utilizza la chiave KMS per decrittografare la chiave dati e quindi restituisce la chiave privata in testo non crittografato. Utilizza la chiave privata di testo normale per generare la firma. Quindi rimuovi la chiave privata di testo normale dalla memoria il prima possibile.

Per firmare un messaggio, creare un messaggio digest utilizzando una funzione hash di crittografia, ad esempio il comando [dgst](#) in OpenSSL. Quindi, passa la tua chiave privata di testo normale all'algoritmo di firma. Il risultato è una firma che rappresenta i contenuti del messaggio. (Potrebbe essere possibile firmare messaggi più brevi senza prima creare un digest. La dimensione massima del messaggio varia in base allo strumento di firma utilizzato.)

Il diagramma seguente mostra come utilizzare la chiave privata in una coppia di chiavi di dati per firmare un messaggio.

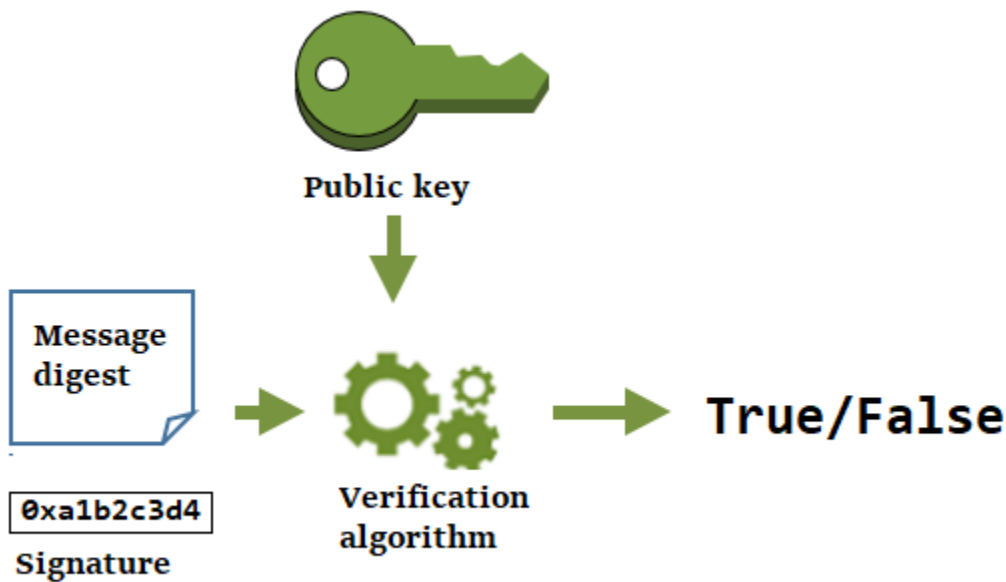


Verificare una firma con una coppia di chiavi di dati

Chiunque abbia la chiave pubblica nella coppia di chiavi di dati può utilizzarla per verificare la firma generata con la chiave privata. La verifica conferma che un utente autorizzato ha firmato il messaggio con la chiave privata e l'algoritmo di firma specificati e che il messaggio non è cambiato da quando è stato firmato.

Per avere successo, la parte che verifica la firma deve generare lo stesso tipo di digest, utilizzare lo stesso algoritmo e utilizzare la chiave pubblica corrispondente alla chiave privata utilizzata per firmare il messaggio.

Nel diagramma seguente viene illustrato come utilizzare la chiave pubblica in una coppia di chiavi di dati per verificare una firma del messaggio.



Ricava un segreto condiviso con coppie di chiavi di dati

L'accordo chiave consente a due peer, ciascuno dotato di una coppia di chiavi pubblica-privata a curva ellittica, di stabilire un segreto condiviso su un canale non sicuro. Per [ricavare un segreto condiviso](#), i due peer devono scambiarsi le proprie chiavi pubbliche su un canale di comunicazione non sicuro (come Internet). Quindi, ciascuna parte utilizza la propria chiave privata e la chiave pubblica del proprio pari per calcolare lo stesso segreto condiviso utilizzando un algoritmo di accordo chiave. È possibile utilizzare il valore segreto condiviso per derivare una chiave simmetrica in grado di crittografare e decrittografare i dati inviati tra i due peer o che può generare e verificare. HMACs

Note

AWS KMS consiglia vivamente di verificare che la chiave pubblica ricevuta provenga dalla parte prevista prima di utilizzarla per ricavare un segreto condiviso.

Esegui operazioni offline con chiavi pubbliche

In una chiave KMS asimmetrica, la chiave privata viene creata e non viene mai crittografata. AWS KMS Per utilizzare la chiave privata, è necessario chiamare. AWS KMS È possibile utilizzare la chiave pubblica interna AWS KMS chiamando le operazioni AWS KMS API. In alternativa, puoi [scaricare la chiave pubblica](#) e condividerla per utilizzarla all'esterno di AWS KMS.

Potresti condividere una chiave pubblica per consentire ad altri di AWS KMS crittografare i dati, ma puoi decrittografarla solo con la tua chiave privata. Oppure per consentire ad altri di verificare esternamente a AWS KMS una firma digitale generata con la chiave privata. Oppure, per condividere la tua chiave pubblica con un altro utente per ricavare un segreto condiviso.

Quando utilizzate la chiave pubblica contenuta nella vostra chiave KMS asimmetrica all'interno AWS KMS, beneficate dell'autenticazione, dell'autorizzazione e della registrazione che fanno parte di ogni operazione. AWS KMS Si riduce anche il rischio di crittografare dati che non possono essere decrittati. Queste funzionalità non sono efficaci al di fuori di. AWS KMS Per informazioni dettagliate, consultare [Considerazioni speciali per il download delle chiavi pubbliche](#).

Tip

Ti interessano le chiavi di dati o le chiavi SSH? In questo argomento viene descritto come gestire le chiavi asimmetriche in AWS Key Management Service, dove la chiave privata non è esportabile. Per le coppie di chiavi di dati esportabili in cui la chiave privata è protetta da una chiave KMS con crittografia simmetrica, vedere. [GenerateDataKeyPair Per assistenza su come scaricare la chiave pubblica associata a un' EC2 istanza Amazon, consulta Recupero della chiave pubblica nella Amazon User Guide e nella Amazon EC2 User Guide. EC2](#)

Argomenti

- [Considerazioni speciali per il download delle chiavi pubbliche](#)
- [Scarica la chiave pubblica](#)
- [Esempi di operazioni offline](#)

Considerazioni speciali per il download delle chiavi pubbliche

Per proteggere le tue chiavi KMS, AWS KMS fornisce controlli di accesso, crittografia autenticata e registri dettagliati di ogni operazione. AWS KMS consente inoltre di impedire l'uso delle chiavi KMS, temporaneamente o permanentemente. Infine, AWS KMS le operazioni sono progettate per ridurre al minimo il rischio di crittografia dei dati che non possono essere decrittografati. Queste funzionalità non sono disponibili quando si utilizzano chiavi pubbliche scaricate all'esterno di AWS KMS.

Autorizzazione

[Le policy chiave](#) e [le policy IAM](#) che controllano l'accesso alla chiave KMS interna non AWS KMS hanno alcun effetto sulle operazioni eseguite all'esterno di AWS. Qualsiasi utente in grado di ottenere la chiave pubblica può utilizzarla all'esterno di AWS KMS anche se non dispone dell'autorizzazione per crittografare i dati o verificare le firme con la chiave KMS.

Limitazioni d'uso delle chiavi

Le restrizioni sull'utilizzo delle chiavi non sono efficaci al di fuori di AWS KMS. Se si richiama l'operazione [Encrypt](#) con una chiave KMS con un KeyUsage di SIGN_VERIFY, l'operazione AWS KMS ha esito negativo. Tuttavia, se si crittografano i dati all'esterno di AWS KMS con una chiave pubblica proveniente da una chiave KMS con un KeyUsage di SIGN_VERIFY o KEY_AGREEMENT, i dati non possono essere decrittografati.

Restrizioni sugli algoritmi

Le restrizioni sugli algoritmi di crittografia e firma supportati non sono efficaci al di fuori di AWS KMS. Se si crittografano i dati con la chiave pubblica proveniente da una chiave KMS esterna e si utilizza un algoritmo di crittografia di AWS KMS che non supporta, i dati AWS KMS non possono essere decrittografati.

Disattivazione ed eliminazione di chiavi KMS

Le azioni che puoi intraprendere per impedire l'uso della chiave KMS in un'operazione crittografica interna non impediscono a nessuno di utilizzare la chiave pubblica all'esterno di AWS KMS. Ad esempio, la disattivazione di una chiave KMS, la pianificazione dell'eliminazione di una chiave KMS, l'eliminazione di una chiave KMS o l'eliminazione del materiale di chiave da una chiave KMS non hanno alcun effetto su una chiave pubblica esterna a AWS KMS. Se elimini una chiave KMS asimmetrica o elimini o perdi il relativo materiale chiave, i dati crittografati con una chiave pubblica esterna non sono recuperabili. AWS KMS

Registrazione

AWS CloudTrail i registri che registrano ogni AWS KMS operazione, inclusa la richiesta, la risposta, la data, l'ora e l'utente autorizzato, non registrano l'uso della chiave pubblica all'esterno di AWS KMS.

Verifica offline con coppie di SM2 chiavi (solo regioni della Cina)

Per verificare una firma all'esterno AWS KMS con una chiave SM2 pubblica, devi specificare l'ID distintivo. Per impostazione predefinita, AWS KMS viene utilizzato 1234567812345678 come ID distintivo. Per ulteriori informazioni, consulta [Verifica offline con coppie di SM2 chiavi \(solo regioni della Cina\)](#).

Scarica la chiave pubblica

È possibile scaricare la chiave pubblica da una coppia di chiavi KMS asimmetrica nella console o AWS KMS utilizzando l'operazione. [GetPublicKey](#) Per scaricare la chiave pubblica, è necessario disporre dell'`kms:GetPublicKey` autorizzazione sulla chiave KMS asimmetrica.

[La chiave pubblica AWS KMS restituita è una chiave pubblica X.509 con codifica DER, nota anche come SubjectPublicKeyInfo \(SPKI\), come definita in RFC 5280.](#) Quando si utilizza l'API HTTP o, il valore è codificato in Base64. AWS CLI Altrimenti, non è codificato in Base64.

Per scaricare la chiave pubblica da una coppia di chiavi KMS asimmetrica, sono necessarie le autorizzazioni. `kms:GetPublicKey` Per ulteriori informazioni sulle AWS KMS autorizzazioni, consulta. [Riferimento per le autorizzazioni](#)

Utilizzo della console AWS KMS

Puoi utilizzare il Console di gestione AWS per visualizzare, copiare e scaricare la chiave pubblica da una chiave KMS asimmetrica del tuo Account AWS. Per scaricare la chiave pubblica da una chiave KMS asimmetrica in diversi modi, utilizza l'API. Account AWS AWS KMS

1. [Accedi a Console di gestione AWS e apri la console AWS Key Management Service \(AWS KMS\) su /kms. `https://console.aws.amazon.com`](#)
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
4. Scegli l'alias o l'ID chiave di una chiave KMS asimmetrica.

5. Scegli la tab Configurazione crittografica. Registra i valori dei campi Specifica della chiave, Utilizzo della chiave e Algoritmi di crittografia o Algoritmi di firma. Dovrai utilizzare questi valori per utilizzare la chiave pubblica all'esterno di AWS KMS. Assicurarsi di condividere queste informazioni quando si condivide la chiave pubblica.
6. Scegliere la scheda Public key (Chiave pubblica).
7. Per copiare la chiave pubblica negli Appunti, scegliere Copy (Copia). Per scaricare la chiave pubblica in un file, scegliere Download (Scarica).

Utilizzando l' AWS KMS API

L'[GetPublicKey](#) operazione restituisce la chiave pubblica in una chiave KMS asimmetrica. Restituisce inoltre informazioni critiche necessarie per utilizzare correttamente la chiave pubblica all'esterno AWS KMS, inclusi gli algoritmi di utilizzo della chiave e di crittografia. Assicurati di salvare questi valori e di condividerli ogni volta che condividi la chiave pubblica.

Gli esempi in questa sezione utilizzano [AWS Command Line Interface \(AWS CLI\)](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

Per specificare una chiave KMS, utilizza l'[ID chiave](#), l'[ARN di chiave](#), il [nome dell'alias](#) o l'[ARN di alias](#). Quando utilizzi un nome alias, aggiungi il prefisso alias/. Per specificare una chiave KMS in un'altra chiave Account AWS, è necessario utilizzare la relativa chiave ARN o l'alias ARN.

Prima di eseguire questo comando, sostituisci il nome alias di esempio con un identificatore valido per la chiave KMS. Per eseguire questo comando, è necessario disporre delle autorizzazioni `kms:GetPublicKey` per la chiave KMS.

```
$ aws kms get-public-key --key-id alias/example_RSA_3072

{
  "KeySpec": "RSA_3072",
  "KeyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyUsage": "ENCRYPT_DECRYPT",
  "EncryptionAlgorithms": [
    "RSAES_OAEP_SHA_1",
    "RSAES_OAEP_SHA_256"
  ],
  "PublicKey": "MIIBojANBgkqhkiG..."
}
```

Esempi di operazioni offline

Dopo aver [scaricato la chiave pubblica](#) della tua coppia di chiavi KMS asimmetrica, puoi condividerla con altri e utilizzarla per eseguire operazioni offline.

AWS CloudTrail i registri che registrano ogni AWS KMS operazione, inclusa la richiesta, la risposta, la data, l'ora e l'utente autorizzato, non registrano l'uso della chiave pubblica all'esterno di AWS KMS.

Questo argomento fornisce esempi di operazioni offline e dettagli sugli strumenti AWS KMS forniti per semplificare le operazioni offline.

Argomenti

- [Ricevere segreti condivisi offline](#)
- [Verifica offline con coppie di chiavi ML-DSA](#)
- [Verifica offline con coppie di SM2 chiavi \(solo regioni della Cina\)](#)

Ricevere segreti condivisi offline

È possibile [scaricare la chiave pubblica](#) della coppia di chiavi ECC per utilizzarla in operazioni offline, ovvero operazioni esterne a AWS KMS.

La seguente procedura dettagliata di [OpenSSL](#) illustra un metodo per derivare un segreto condiviso oltre all'utilizzo della chiave pubblica di una coppia AWS KMS di chiavi ECC KMS e di una chiave privata creata con OpenSSL.

1. Crea una coppia di chiavi ECC in OpenSSL e preparala per l'uso con AWS KMS

```
// Create an ECC key pair in OpenSSL and save the private key in
openssl_ecc_key_priv.pem
export OPENSSL_CURVE_NAME="P-256"
export KMS_CURVE_NAME="ECC_NIST_P256"

export OPENSSL_KEY1_PRIV_PEM="openssl_ecc_key1_priv.pem"
openssl ecparam -name ${OPENSSL_CURVE_NAME} -genkey -out ${OPENSSL_KEY1_PRIV_PEM}

// Derive the public key from the private key
export OPENSSL_KEY1_PUB_PEM="openssl_ecc_key1_pub.pem"
openssl ec -in ${OPENSSL_KEY1_PRIV_PEM} -pubout -outform pem \
    -out ${OPENSSL_KEY1_PUB_PEM}
```

```
// View the PEM file containing the public key and extract the public key as a
// Base64 encoded string into OPENSSL_KEY1_PUB_BASE64 for use with AWS KMS
export OPENSSL_KEY1_PUB_BASE64=`cat ${OPENSSL_KEY1_PUB_PEM} | \
    tee /dev/stderr | grep -v "PUBLIC KEY" | tr -d "\n"`
```

2. Crea una coppia di chiavi ECC Key Agreement AWS KMS e preparala per l'uso con OpenSSL.

```
// Create a KMS key on the same curve as the key pair from step 1
// with a key usage of KEY_AGREEMENT
// Save its ARN in KMS_KEY1_ARN.
export KMS_KEY1_ARN=`aws kms create-key --key-spec ${KMS_CURVE_NAME} \
    --key-usage KEY_AGREEMENT | tee /dev/stderr | jq -r .KeyMetadata.Arn`

// Download the public key and save the Base64-encoded version in KMS_KEY1_PUB_BASE64

export KMS_KEY1_PUB_BASE64=`aws kms get-public-key --key-id ${KMS_KEY1_ARN} | \
    tee /dev/stderr | jq -r .PublicKey`

// Create a PEM file for the public KMS key for use with OpenSSL
export KMS_KEY1_PUB_PEM="aws_kms_ecdh_key1_pub.pem"
echo "-----BEGIN PUBLIC KEY-----" > ${KMS_KEY1_PUB_PEM}
echo ${KMS_KEY1_PUB_BASE64} | fold -w 64 >> ${KMS_KEY1_PUB_PEM}
echo "-----END PUBLIC KEY-----" >> ${KMS_KEY1_PUB_PEM}
```

3. Deriva un segreto condiviso in OpenSSL utilizzando la chiave privata in OpenSSL e la chiave KMS pubblica.

```
export OPENSSL_SHARED_SECRET1_BIN="openssl_shared_secret1.bin"
openssl pkeyutl -derive -inkey ${OPENSSL_KEY1_PRIV_PEM} \
    -peerkey ${KMS_KEY1_PUB_PEM} -out ${OPENSSL_SHARED_SECRET1_BIN}
```

Verifica offline con coppie di chiavi ML-DSA

AWS KMS supporta una variante con copertura della firma ML-DSA, come descritto nella sezione 3.4 dello [standard Federal Information Processing Standards \(FIPS\) 204](#) per messaggi fino a 4 KB di byte.

Per firmare messaggi di dimensioni superiori a 4 KB, si esegue la fase di preelaborazione dei messaggi all'esterno di AWS KMS. Questo passaggio di hashing crea un messaggio a 64 byte rappresentativo μ , come definito in NIST FIPS 204, sezione 6.2.

AWS KMS ha un tipo di messaggio chiamato `EXTERNAL_MU` per messaggi di dimensioni superiori a 4 KB. Quando lo usi al posto del tipo di `RAW` messaggio, AWS KMS:

- Presuppone che tu abbia già eseguito la fase di hashing
- Salta il processo di hashing interno
- Funziona con messaggi di qualsiasi dimensione

Quando si verifica un messaggio, il metodo utilizzato dipende dalla limitazione delle dimensioni del sistema o della libreria esterna e dal fatto che supporti il messaggio rappresentativo a 64 byte μ :

- Se il messaggio è più piccolo del limite di dimensione, utilizza il tipo di messaggio. `RAW`
- Se il messaggio è più grande del limite di dimensione, utilizzate il simbolo μ nel sistema esterno.

Le sezioni seguenti mostrano come firmare i messaggi utilizzando AWS KMS e verificare i messaggi utilizzando OpenSSL. Forniamo esempi per messaggi inferiori e superiori al limite di 4 KB di dimensione dei messaggi imposto da. AWS KMS OpenSSL non impone limiti alla dimensione dei messaggi per la verifica.

Per entrambi gli esempi, per prima cosa ottieni la chiave pubblica da. AWS KMS Utilizzando il seguente comando AWS CLI :

```
aws kms get-public-key \
  --key-id _<1234abcd-12ab-34cd-56ef-1234567890ab>_ \
  --output text \
  --query PublicKey | base64 --decode > public_key.der
```

Dimensione del messaggio inferiore a 4 KB

Per messaggi inferiori a 4 KB, utilizza il tipo di `RAW` messaggio con AWS KMS. Sebbene sia possibile utilizzarlo `EXTERNAL_MU`, non è necessario per i messaggi entro il limite di dimensione.

Usa il seguente AWS CLI comando per firmare il messaggio:

```
aws kms sign \
  --key-id _<1234abcd-12ab-34cd-56ef-1234567890ab>_ \
  --message 'your message' \
  --message-type RAW \
  --signing-algorithm ML_DSA_SHAKE_256 \
  --output text \
```

```
--query Signature | base64 --decode > ExampleSignature.bin
```

Per verificare questo messaggio utilizzando OpenSSL, usa il seguente comando:

```
echo -n 'your message' | ./openssl dgst -verify public_key.der -signature
ExampleSignature.bin
```

Dimensione del messaggio superiore a 4 KB

Per firmare messaggi di dimensioni superiori a 4 KB, utilizza il tipo di `EXTERNAL_MU` messaggio. Quando lo utilizzi `EXTERNAL_MU`, preimpostate l'hash del messaggio esternamente su un file rappresentativo di 64 byte μ , come definito nella sezione 6.2 del NIST FIPS 204, e lo passate alle operazioni di firma o verifica. Si noti che questo è diverso dal «MLDSA pre-hash» o HashML-DSA definito nella sezione 5.4 del NIST FIPS 204.

1. Innanzitutto, costruisci un prefisso per il messaggio. Il prefisso contiene un separatore di dominio, la lunghezza di qualsiasi contesto e il contesto. L'impostazione predefinita per il separatore di dominio e la lunghezza del contesto è zero.
2. Antepone il prefisso del messaggio al messaggio.
3. Utilizzare SHAKE256 per eseguire l'hash della chiave pubblica e anteporla al risultato del passaggio 2.
4. Infine, esegui l'hash del risultato del passaggio 3 per produrre un file da 64 byte. `EXTERNAL_MU`

L'esempio seguente utilizza OpenSSL 3.5 per costruire: `EXTERNAL_MU`

```
{
  openssl asn1parse -inform DER -in public_key.der -strparse 17 -noout -out - 2>/dev/
null |
  openssl dgst -provider default -shake256 -xoflen 64 -binary;
  printf '\x00\x00';
  echo -n "your message"
} | openssl dgst -provider default -shake256 -xoflen 64 -binary > mu.bin
```

Dopo aver creato il `mu.bin` file, chiama l' AWS KMS API con il seguente comando per firmare il messaggio:

```
aws kms sign \
  --key-id _<1234abcd-12ab-34cd-56ef-1234567890ab>_ \
  --message fileb://mu.bin \
```

```
--message-type EXTERNAL_MU \  
--signing-algorithm ML_DSA_SHAKE_256 \  
--output text \  
--query Signature | base64 --decode > ExampleSignature.bin
```

La firma risultante è la stessa del messaggio originale. RAW Puoi usare lo stesso comando OpenSSL 3.5 per verificare il messaggio:

```
echo -n 'your message' | ./openssl dgst -verify public_key.der -signature  
ExampleSignature.bin
```

Verifica offline con coppie di SM2 chiavi (solo regioni della Cina)

Per verificare una firma all'esterno AWS KMS con una chiave SM2 pubblica, devi specificare l'ID distintivo. Quando si passa un messaggio non elaborato [MessageType:RAW](#), all'API [Sign](#), AWS KMS utilizza l'ID distintivo predefinito1234567812345678, definito da OSCA nel 0009-2012. GM/T Non puoi specificare il tuo ID distintivo all'interno di AWS KMS.

Tuttavia, se stai generando un digest del messaggio all'esterno di AWS, puoi specificare il tuo ID distintivo, quindi passare il digest del messaggio, a to sign. [MessageType:DIGEST](#) AWS KMS A tale scopo, modifica il valore DEFAULT_DISTINGUISHING_ID in classe SM2OfflineOperationHelper. L'ID distintivo specificato può essere qualsiasi stringa lunga fino a 8.192 caratteri. Dopo aver AWS KMS firmato il digest del messaggio, è necessario il message digest o il messaggio e l'ID distintivo utilizzati per calcolare il digest per verificarlo offline.

Important

Il codice di riferimento SM2OfflineOperationHelper è progettato per essere compatibile con [Bouncy Castle](#) versione 1.68. Per assistenza con altre versioni, contatta [bouncycastle.org](#).

Classe SM2OfflineOperationHelper

Per aiutarvi con le operazioni offline con le SM2 chiavi, la SM2OfflineOperationHelper classe per Java dispone di metodi che eseguono le attività al posto vostro. È possibile utilizzare questa classe helper come modello per altri fornitori di crittografia.

All'interno AWS KMS, le conversioni di testo cifrato non elaborato e i calcoli del SM2 DSA Message Digest avvengono automaticamente. Non tutti i provider di crittografia implementano allo stesso modo. Alcune librerie, come [OpenSSL](#) versione 1.1.1 e successive, eseguono queste azioni automaticamente. AWS KMS ha confermato questo comportamento durante i test con OpenSSL versione 3.0. Utilizza la seguente classe `SM2OfflineOperationHelper` con librerie, come [Bouncy Castle](#), che richiedono di eseguire manualmente queste conversioni e calcoli.

La classe `SM2OfflineOperationHelper` fornisce metodi per le seguenti operazioni offline:

- Calcolo del digest del messaggio

Per generare un riepilogo dei messaggi offline da utilizzare per la verifica offline o da trasferire AWS KMS per firmare, utilizza questo metodo. `calculateSM2Digest` Il `calculateSM2Digest` metodo genera un digest dei messaggi con l'algoritmo di SM3 hashing. L'[GetPublicKey](#) API restituisce la chiave pubblica in formato binario. È necessario analizzare la chiave binaria in un file Java `PublicKey`. Fornisci il messaggio alla chiave pubblica analizzata. Il metodo combina automaticamente il tuo messaggio con l'ID distintivo predefinito, 1234567812345678, ma è possibile impostare il proprio ID distintivo modificando il valore `DEFAULT_DISTINGUISHING_ID`.

- Verify

Per verificare la firma offline, usa il metodo `offlineSM2DSAVerify`. Il metodo `offlineSM2DSAVerify` utilizza il digest del messaggio calcolato dall'ID distintivo specificato e il messaggio originale fornito per verificare la firma digitale. L'[GetPublicKey](#) API restituisce la chiave pubblica in formato binario. È necessario analizzare la chiave binaria in un file Java `PublicKey`. Fornisci alla chiave pubblica analizzata il messaggio originale e la firma che desideri verificare. Per maggiori dettagli, consulta [Verifica offline con coppie di SM2 chiavi](#).

- Encrypt

Per crittografare il testo normale offline utilizza il metodo `offlineSM2PKEEncrypt`. Questo metodo garantisce che il testo cifrato sia in un formato AWS KMS decifrabile. Il `offlineSM2PKEEncrypt` metodo cripta il testo non crittografato e quindi converte il testo cifrato non elaborato prodotto da PKE nel formato ASN.1. SM2 L'API restituisce la chiave pubblica in formato binario. [GetPublicKey](#) È necessario analizzare la chiave binaria in un file Java `PublicKey`. Fornisci alla chiave pubblica analizzata il testo normale che desideri crittografare.

Se non sei sicuro di dover eseguire la conversione, usa la seguente operazione OpenSSL per testare il formato del tuo testo criptato. Se l'operazione non riesce, è necessario convertire il testo criptato nel formato ASN.1.

```
openssl asn1parse -inform DER -in ciphertext.der
```

Per impostazione predefinita, la `SM2OfflineOperationHelper` classe utilizza l'ID distintivo predefinito quando genera i 1234567812345678 digest dei messaggi per SM2 le operazioni DSA.

```
package com.amazon.kms.utils;

import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import java.io.IOException;
import java.math.BigInteger;
import java.nio.ByteBuffer;
import java.nio.charset.StandardCharsets;
import java.security.InvalidKeyException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.NoSuchProviderException;
import java.security.PrivateKey;
import java.security.PublicKey;

import org.bouncycastle.crypto.CryptoException;
import org.bouncycastle.jce.interfaces.ECPublicKey;

import java.util.Arrays;

import org.bouncycastle.asn1.ASN1EncodableVector;
import org.bouncycastle.asn1.ASN1Integer;
import org.bouncycastle.asn1.DEROctetString;
import org.bouncycastle.asn1.DERSequence;
import org.bouncycastle.asn1.gm.GMNamedCurves;
import org.bouncycastle.asn1.x9.X9ECParameters;
import org.bouncycastle.crypto.CipherParameters;
import org.bouncycastle.crypto.params.ParametersWithID;
import org.bouncycastle.crypto.params.ParametersWithRandom;
import org.bouncycastle.crypto.signers.SM2Signer;
```



```

import org.bouncycastle.jcajce.provider.asymmetric.util.ECUtil;

public class SM2OfflineOperationHelper {
    // You can change the DEFAULT_DISTINGUISHING_ID value to set your own
    distinguishing ID,
    // the DEFAULT_DISTINGUISHING_ID can be any string up to 8,192 characters long.
    private static final byte[] DEFAULT_DISTINGUISHING_ID =
"1234567812345678".getBytes(StandardCharsets.UTF_8);
    private static final X9ECParameters SM2_X9EC_PARAMETERS =
GMNamedCurves.getByName("sm2p256v1");

    // ***calculateSM2Digest***
    // Calculate message digest
    public static byte[] calculateSM2Digest(final PublicKey publicKey, final byte[]
message) throws
        NoSuchProviderException, NoSuchAlgorithmException {
        final ECPublicKey ecPublicKey = (ECPublicKey) publicKey;

        // Generate SM3 hash of default distinguishing ID, 1234567812345678
        final int entlenA = DEFAULT_DISTINGUISHING_ID.length * 8;
        final byte [] entla = new byte[] { (byte) (entlenA & 0xFF00), (byte) (entlenA &
0x00FF) };
        final byte [] a = SM2_X9EC_PARAMETERS.getCurve().getA().getEncoded();
        final byte [] b = SM2_X9EC_PARAMETERS.getCurve().getB().getEncoded();
        final byte [] xg = SM2_X9EC_PARAMETERS.getG().getXCoord().getEncoded();
        final byte [] yg = SM2_X9EC_PARAMETERS.getG().getYCoord().getEncoded();
        final byte[] xa = ecPublicKey.getQ().getXCoord().getEncoded();
        final byte[] ya = ecPublicKey.getQ().getYCoord().getEncoded();
        final byte[] za = MessageDigest.getInstance("SM3", "BC")
            .digest(ByteBuffer.allocate(entla.length +
DEFAULT_DISTINGUISHING_ID.length + a.length + b.length + xg.length + yg.length +
xa.length +
ya.length).put(entla).put(DEFAULT_DISTINGUISHING_ID).put(a).put(b).put(xg).put(yg).put(xa).put
            .array());

        // Combine hashed distinguishing ID with original message to generate final
        digest
        return MessageDigest.getInstance("SM3", "BC")
            .digest(ByteBuffer.allocate(za.length +
message.length).put(za).put(message)
            .array());
    }

    // ***offlineSM2DSAVerify***

```

```

// Verify digital signature with SM2 public key
public static boolean offlineSM2DSAVerify(final PublicKey publicKey, final byte []
message,
    final byte [] signature) throws InvalidKeyException {
    final SM2Signer signer = new SM2Signer();
    CipherParameters cipherParameters =
ECUtil.generatePublicKeyParameter(publicKey);
    cipherParameters = new ParametersWithID(cipherParameters,
DEFAULT_DISTINGUISHING_ID);
    signer.init(false, cipherParameters);
    signer.update(message, 0, message.length);
    return signer.verifySignature(signature);
}

// ***offlineSM2PKEEncrypt***
// Encrypt data with SM2 public key
public static byte[] offlineSM2PKEEncrypt(final PublicKey publicKey, final byte []
plaintext) throws
    NoSuchPaddingException, NoSuchAlgorithmException, NoSuchProviderException,
InvalidKeyException,
    BadPaddingException, IllegalBlockSizeException, IOException {
    final Cipher sm2Cipher = Cipher.getInstance("SM2", "BC");
    sm2Cipher.init(Cipher.ENCRYPT_MODE, publicKey);

    // By default, Bouncy Castle returns raw ciphertext in the c1c2c3 format
    final byte [] cipherText = sm2Cipher.doFinal(plaintext);

    // Convert the raw ciphertext to the ASN.1 format before passing it to AWS KMS
    final ASN1EncodableVector asn1EncodableVector = new ASN1EncodableVector();
    final int coordinateLength = (SM2_X9EC_PARAMETERS.getCurve().getFieldSize() +
7) / 8 * 2 + 1;
    final int sm3HashLength = 32;
    final int xCoordinateInCipherText = 33;
    final int yCoordinateInCipherText = 65;
    byte[] coords = new byte[coordinateLength];
    byte[] sm3Hash = new byte[sm3HashLength];
    byte[] remainingCipherText = new byte[cipherText.length - coordinateLength -
sm3HashLength];

    // Split components out of the ciphertext
    System.arraycopy(cipherText, 0, coords, 0, coordinateLength);
    System.arraycopy(cipherText, cipherText.length - sm3HashLength, sm3Hash, 0,
sm3HashLength);

```

```
        System.arraycopy(cipherText, coordinateLength, remainingCipherText,
0,cipherText.length - coordinateLength - sm3HashLength);

        // Build standard SM2PKE ASN.1 ciphertext vector
        asn1EncodableVector.add(new ASN1Integer(new BigInteger(1,
Arrays.copyOfRange(coords, 1, xCoordinateInCipherText))));
        asn1EncodableVector.add(new ASN1Integer(new BigInteger(1,
Arrays.copyOfRange(coords, xCoordinateInCipherText, yCoordinateInCipherText))));
        asn1EncodableVector.add(new DEROctetString(sm3Hash));
        asn1EncodableVector.add(new DEROctetString(remainingCipherText));

        return new DERSequence(asn1EncodableVector).getEncoded("DER");
    }
}
```

Monitor AWS KMS keys

Il monitoraggio è una parte importante per comprendere la disponibilità, lo stato e l'utilizzo del tuo AWS KMS keys sistema AWS KMS e per mantenere l'affidabilità, la disponibilità e le prestazioni delle tue AWS soluzioni. Raccogliere i dati sul monitoraggio da tutte le parti della soluzione AWS consente un debug più facile di eventuali guasti in più punti. Prima di iniziare il monitoraggio delle chiavi KMS è tuttavia opportuno creare un piano di monitoraggio che includa le risposte alle seguenti domande:

- Quali sono gli obiettivi del monitoraggio?
- Di quali risorse si intende eseguire il monitoraggio?
- Con quale frequenza sarà eseguito il monitoraggio di queste risorse?
- Quali [strumenti di monitoraggio](#) utilizzerai?
- Chi eseguirà i processi di monitoraggio?
- Chi deve ricevere una notifica quando si verifica un problema?

Il passaggio successivo è quello di monitorare le tue chiavi KMS nel corso del tempo per stabilire una linea di base per l'uso normale di AWS KMS e le aspettative nel proprio ambiente. Quando monitori le chiavi KMS, archivia i dati di monitoraggio storici per poterli confrontare con i dati correnti, identificare i normali modelli di prestazioni e le anomalie e ideare metodi per risolvere i problemi.

Ad esempio, puoi monitorare l'attività e gli eventi delle AWS KMS API che influiscono sulle tue chiavi KMS. Quando i dati sono sopra o sotto il livello stabilito, potrebbe essere necessario indagare o intraprendere azioni correttive.

Per stabilire una baseline per modelli normali, devi monitorare gli elementi seguenti:

- AWS KMS Attività dell'API per le operazioni sul piano dati. Si tratta di [operazioni crittografiche](#) che utilizzano una chiave KMS, ad esempio [Decrypt](#), [Encrypt](#) e [ReEncryptGenerateDataKey](#)
- AWS KMS Attività delle API per le operazioni del piano di controllo che sono importanti per te. Queste operazioni gestiscono una chiave KMS e potresti voler monitorare quelle che modificano la disponibilità di una chiave KMS (ad esempio [ScheduleKeyDeletion](#), [CancelKeyDeletionDisableKeyEnableKeyImportKeyMaterial](#), e [DeleteImportedKeyMaterial](#)) o modificano il controllo di accesso di una chiave KMS (come [PutKeyPolicy](#) and). [RevokeGrant](#)
- Altre AWS KMS metriche (come il tempo che manca alla scadenza del [materiale chiave importato](#)) ed eventi (come la scadenza del materiale chiave importato o l'eliminazione o la rotazione delle chiavi di una chiave KMS).

Strumenti di monitoraggio

AWS fornisce vari strumenti che puoi utilizzare per monitorare le tue chiavi KMS. Alcuni di questi strumenti possono essere configurati in modo che eseguano automaticamente il monitoraggio, mentre altri richiedono l'intervento manuale. Si consiglia di automatizzare il più possibile i processi di monitoraggio.

Strumenti di monitoraggio automatici

Per controllare le chiavi KMS e segnalare l'eventuale presenza di problemi, puoi usare gli strumenti di monitoraggio automatici seguenti.

- **AWS CloudTrail Monitoraggio dei log:** condividi i file di CloudTrail registro tra account, monitora i file di registro in tempo reale inviandoli a CloudWatch Logs, scrivi applicazioni di elaborazione dei log con la [CloudTrail Processing Library](#) e verifica che i file di registro non siano cambiati dopo la consegna da parte di CloudTrail. Per ulteriori informazioni, consulta [Lavorare con i file di CloudTrail registro nella Guida](#) per l'AWS CloudTrail utente.
- **Amazon CloudWatch Alarms:** monitora una singola metrica in un periodo di tempo specificato ed esegui una o più azioni in base al valore della metrica rispetto a una determinata soglia in diversi periodi di tempo. L'azione è una notifica inviata a un argomento di Amazon Simple Notification Service (Amazon SNS) o a una politica di Amazon Auto EC2 Scaling. CloudWatch gli allarmi non richiamano azioni semplicemente perché si trovano in uno stato particolare; lo stato deve essere cambiato e mantenuto per un determinato numero di periodi. Per ulteriori informazioni, consulta [Monitora le chiavi KMS con Amazon CloudWatch](#).
- **Amazon EventBridge:** abbina gli eventi e li indirizza a una o più funzioni o flussi di destinazione per acquisire informazioni sullo stato e, se necessario, apportare modifiche o intraprendere azioni correttive. Per ulteriori informazioni, consulta [Monitora le chiavi KMS con Amazon EventBridge](#) la [Amazon EventBridge User Guide](#).
- **Amazon CloudWatch Logs:** monitora, archivia e accedi ai tuoi file di registro da AWS CloudTrail o altre fonti. Per ulteriori informazioni, consulta la [Amazon CloudWatch Logs User Guide](#).

Strumenti di monitoraggio manuali

Un'altra parte importante del monitoraggio delle chiavi KMS consiste nel monitorare manualmente gli elementi che gli CloudWatch allarmi e gli eventi non coprono. I AWS dashboard AWS KMS CloudWatch, AWS Trusted Advisor, e altri forniscono una at-a-glance panoramica dello stato dell'ambiente. AWS

È possibile [personalizzare](#) le pagine Chiavi gestite da AWS e Chiavi gestite dal cliente della [console AWS KMS](#) per visualizzare le seguenti informazioni su ciascuna chiave KMS:

- ID chiave
- Stato
- Data di creazione
- Data di scadenza (per le chiavi KMS con [materiale chiave importato](#))
- Origin
- ID dell'archivio delle chiavi personalizzate (per le chiavi KMS negli [archivi delle chiavi personalizzate](#))

Il [pannello di controllo della console CloudWatch](#) mostra quanto segue:

- Stato e allarmi attuali
- Grafici degli allarmi e delle risorse
- Stato di integrità dei servizi

Inoltre, è possibile utilizzare CloudWatch per effettuare le seguenti operazioni:

- Crea [pannelli di controllo personalizzati](#) per monitorare i servizi di interesse.
- Crea grafici dei dati dei parametri per la risoluzione di problemi e il rilevamento di tendenze.
- Cerca e sfoglia tutte le metriche AWS delle tue risorse
- Crea e modifica gli allarmi per ricevere le notifiche dei problemi.

AWS Trusted Advisor può aiutarti a monitorare AWS le tue risorse per migliorare prestazioni, affidabilità, sicurezza ed economicità. Quattro Trusted Advisor controlli sono disponibili per tutti gli utenti; più di 50 controlli sono disponibili per gli utenti con un piano di supporto Business o Enterprise. Per ulteriori informazioni, consulta [AWS Trusted Advisor](#).

Registrazione delle chiamate AWS KMS API con AWS CloudTrail

AWS KMS è integrato con [AWS CloudTrail](#), un servizio che registra tutte le chiamate effettuate AWS KMS da utenti, ruoli e altri AWS servizi. CloudTrail acquisisce tutte le chiamate API a AWS KMS come eventi, incluse le chiamate dalla AWS KMS console AWS KMS APIs, dai CloudFormation modelli, da AWS Command Line Interface (AWS CLI) e AWS Strumenti per PowerShell.

CloudTrail [registra tutte le AWS KMS operazioni, incluse le operazioni di sola lettura, come ListAliasesand GetKeyRotationStatus](#), le operazioni che gestiscono le chiavi KMS, come [and](#), e le operazioni crittografiche [PutKeyPolicy](#), come [CreateKey](#) [Decrypt](#). [GenerateDataKey](#)

Registra anche le operazioni interne che AWS KMS richiedono l'utente, ad esempio, e. [DeleteExpiredKeyMaterialDeleteKeySynchronizeMultiRegionKeyRotateKey](#)

CloudTrail registra tutte le operazioni riuscite e, in alcuni scenari, i tentativi di chiamata non riusciti, ad esempio quando al chiamante viene negato l'accesso a una risorsa. [Le operazioni tra account sulle chiavi KMS](#) vengono registrate sia nell'account del chiamante che nell'account del proprietario della chiave KMS. Tuttavia, AWS KMS le richieste tra più account che vengono rifiutate perché l'accesso è negato vengono registrate solo nell'account del chiamante.

Per motivi di sicurezza, alcuni campi vengono omessi dalle voci di AWS KMS registro, come il Plaintext parametro di una richiesta [Encrypt](#) e la risposta o qualsiasi operazione di [GetKeyPolicy](#) crittografia. Per semplificare la ricerca delle voci di CloudTrail registro per particolari chiavi KMS, AWS KMS aggiunge l'[ARN della chiave](#) KMS interessata al campo nelle voci di registro per AWS KMS alcune operazioni di gestione delle chiavi, anche quando l'operazione API non restituisce l'ARN della chiave. responseElements

Sebbene per impostazione predefinita, tutte le AWS KMS azioni vengano registrate come CloudTrail eventi, è possibile escludere AWS KMS le azioni da una traccia. CloudTrail Per informazioni dettagliate, vedi [Esclusione di AWS KMS eventi da un percorso](#).

Ulteriori informazioni:

- Per esempi di CloudTrail log di AWS KMS operazioni per piattaforme certificate, consulta. [Monitoraggio delle richieste attestate](#)

Argomenti

- [Ricerca delle voci di AWS KMS registro in CloudTrail](#)
- [Esclusione di AWS KMS eventi da un percorso](#)
- [Esempi di voci di AWS KMS registro](#)

Ricerca delle voci di AWS KMS registro in CloudTrail

Per cercare le voci di CloudTrail registro, usa la [CloudTrail console](#) o l'[CloudTrail LookupEvents](#) operazione. CloudTrail supporta numerosi [valori di attributo](#) per filtrare la ricerca, tra cui il nome dell'evento, il nome utente e l'origine dell'evento.

Per facilitare la ricerca delle voci di AWS KMS registro CloudTrail, AWS KMS compila i seguenti campi di immissione del CloudTrail registro.

Note

A partire da dicembre 2022, AWS KMS compila gli attributi Tipo di risorsa e Nome risorsa in tutte le operazioni di gestione che modificano una particolare chiave KMS. Questi valori degli attributi potrebbero essere nulli nelle CloudTrail voci precedenti per le seguenti operazioni: [CreateAlias](#), [CreateGrant](#), [DeleteAlias](#), [DeleteImportedKeyMaterial](#), [ImportKeyMaterial](#), [ReplicateKeyRetireGrantRevokeGrant](#), [UpdateAlias](#), [UpdatePrimaryRegion](#)

Attributo	Valore	Voci di log
Origine evento (EventSource)	kms.amazonaws.com	Tutte le operazioni.
Tipo di risorsa (ResourceType)	AWS::KMS::Key	Operazioni di gestione che modificano una determinata chiave KMS, ad esempio <code>CreateKey</code> e <code>EnableKey</code> , ma non <code>ListKeys</code> .
Nome risorsa (ResourceName)	ARN chiave (o ID chiave e ARN chiave)	Operazioni di gestione che modificano una determinata chiave KMS, ad esempio <code>CreateKey</code> e <code>EnableKey</code> , ma non <code>ListKeys</code> .

Per aiutarti a trovare le voci di registro per le operazioni di gestione su particolari chiavi KMS, AWS KMS registra l'ARN della chiave KMS interessata nell'elemento `responseElements.keyId` della voce di registro, anche quando l'operazione API non restituisce AWS KMS la chiave ARN.

Ad esempio, una chiamata riuscita all'[DisableKey](#) operazione non restituisce alcun valore nella risposta, ma invece di un valore nullo, il `responseElements.keyId` valore nella [voce di DisableKey registro](#) include la chiave ARN della chiave KMS disattivata.

Questa funzionalità è stata aggiunta a dicembre 2022 e influisce sulle seguenti voci di CloudTrail registro: [CreateAliasCreateGrant](#), [DeleteAlias](#), [DeleteKey](#), [DisableKey](#), [EnableKey](#), [EnableKeyRotation](#), [ImportKeyMaterial](#), [RotateKey](#), [SynchronizeMultiRegionKeyTagResourceUntagResource](#), [UpdateAlias](#), [UpdatePrimaryRegion](#)

Esclusione di AWS KMS eventi da un percorso

Per registrare l'uso e la gestione delle proprie AWS KMS risorse, la maggior parte AWS KMS degli utenti si affida agli eventi di un CloudTrail percorso. Il percorso può essere una preziosa fonte di dati per il controllo di eventi critici, come la creazione, la disabilitazione e l'eliminazione AWS KMS keys, la modifica delle politiche chiave e l'uso delle chiavi KMS da parte dei AWS servizi che operano per conto dell'utente. In alcuni casi, i metadati contenuti in una voce di CloudTrail registro, ad esempio il [contesto di crittografia](#) in un'operazione di crittografia, possono aiutare a evitare o risolvere errori.

Tuttavia, poiché AWS KMS può generare un gran numero di eventi, AWS CloudTrail consente di escludere AWS KMS gli eventi da una traccia. Questa impostazione per percorso esclude tutti AWS KMS gli eventi; non è possibile escludere eventi particolari. AWS KMS

Warning

L'esclusione di AWS KMS eventi da un CloudTrail registro può oscurare le azioni che utilizzano le chiavi KMS. Presta attenzione quando concedi alle entità principali l'autorizzazione `cloudtrail:PutEventSelectors` necessaria per eseguire questa operazione.

Per escludere AWS KMS eventi da un percorso:

- Nella CloudTrail console, utilizza l'impostazione degli eventi del servizio Log Key Management quando [crei un percorso](#) o [lo aggiorni](#). Per istruzioni, consulta [Logging Management Events with the Console di gestione AWS](#) nella Guida per l' AWS CloudTrail utente.
- Nell' CloudTrail API, usa l'[PutEventSelectors](#) operazione. Aggiungere l'attributo `ExcludeManagementEventSources` ai selettori di eventi con un valore `kms.amazonaws.com`. Per un esempio, vedi [Esempio: un percorso che non registra AWS Key Management Service gli eventi](#) nella Guida per l' AWS CloudTrail utente.

È possibile disattivare questa esclusione in qualsiasi momento modificando l'impostazione della console o i selettori di eventi per un trail. Il percorso inizierà quindi a registrare AWS KMS gli eventi.

Tuttavia, non può recuperare AWS KMS gli eventi che si sono verificati mentre l'esclusione era effettiva.

Quando si escludono AWS KMS eventi utilizzando la console o l'API, l'operazione CloudTrail PutEventSelectors API risultante viene registrata anche nei log. CloudTrail Se AWS KMS gli eventi non compaiono nei tuoi CloudTrail log, cerca un PutEventSelectors evento con l'ExcludeManagementEventSourcesattributo impostato su `kms.amazonaws.com`

Esempi di voci di AWS KMS registro

AWS KMS scrive voci nel CloudTrail registro quando si chiama un' AWS KMS operazione e quando un AWS servizio richiama un'operazione per conto dell'utente. AWS KMS scrive anche una voce quando richiama un'operazione per voi. Ad esempio, scrive una voce quando [elimina una chiave KMS](#) che hai pianificato per l'eliminazione.

I seguenti argomenti mostrano esempi di voci di CloudTrail registro relative AWS KMS alle operazioni.

Per esempi di voci di CloudTrail registro delle richieste AWS KMS provenienti da piattaforme attestate, vedere [Monitoraggio delle richieste attestate](#).

Argomenti

- [CancelKeyDeletion](#)
- [ConnectCustomKeyStore](#)
- [CreateAlias](#)
- [CreateCustomKeyStore](#)
- [CreateGrant](#)
- [CreateKey](#)
- [Decrittografia](#)
- [DeleteAlias](#)
- [DeleteCustomKeyStore](#)
- [DeleteExpiredKeyMaterial](#)
- [DeleteImportedKeyMaterial](#)
- [DeleteKey](#)

- [DescribeCustomKeyStores](#)
- [DescribeKey](#)
- [DisableKey](#)
- [DisableKeyRotation](#)
- [DisconnectCustomKeyStore](#)
- [EnableKey](#)
- [EnableKeyRotation](#)
- [Crittografia](#)
- [GenerateDataKey](#)
- [GenerateDataKeyPair](#)
- [GenerateDataKeyPairWithoutPlaintext](#)
- [GenerateDataKeyWithoutPlaintext](#)
- [GenerateMac](#)
- [GenerateRandom](#)
- [GetKeyPolicy](#)
- [GetKeyRotationStatus](#)
- [GetParametersForImport](#)
- [ImportKeyMaterial](#)
- [ListAliases](#)
- [ListGrants](#)
- [ListKeyRotations](#)
- [PutKeyPolicy](#)
- [ReEncrypt](#)
- [ReplicateKey](#)
- [RetireGrant](#)
- [RevokeGrant](#)
- [RotateKey](#)
- [RotateKeyOnDemand](#)

- [ScheduleKeyDeletion](#)
- [Sign](#)
- [SynchronizeMultiRegionKey](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateAlias](#)
- [UpdateCustomKeyStore](#)
- [UpdateKeyDescription](#)
- [UpdatePrimaryRegion](#)
- [VerifyMac](#)
- [Verifica](#)
- [Amazon EC2 , esempio uno](#)
- [Amazon EC2 , secondo esempio](#)

CancelKeyDeletion

L'esempio seguente mostra una voce di AWS CloudTrail registro generata chiamando l'[CancelKeyDeletion](#) operazione. Per informazioni sull'eliminazione AWS KMS keys, vedere [Eliminare un AWS KMS key](#).

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2020-07-27T21:53:17Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CancelKeyDeletion",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
```

```

    "requestParameters": {
      "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
      "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "e3452e68-d4b0-4ec7-a768-7ae96c23764f",
    "eventID": "d818bf03-6655-48e9-8b26-f279a07075fd",
    "readOnly": false,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }
}

```

ConnectCustomKeyStore

L'esempio seguente mostra una voce di AWS CloudTrail registro generata chiamando l'[ConnectCustomKeyStore](#) operazione. Per informazioni sulla connessione a un archivio delle chiavi personalizzate, consultare [Disconnetti un archivio AWS CloudHSM chiavi](#).

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2021-10-21T20:17:32Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "ConnectCustomKeyStore",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",

```

```

    "userAgent": "AWS Internal",
    "requestParameters": {
      "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "responseElements": null,
    "additionalEventData": {
      "customKeyStoreName": "ExampleKeyStore",
      "clusterId": "cluster-1a23b4cdefg"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
  }
}

```

CreateAlias

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[CreateAlias](#) operazione. L'elemento `resources` include i campi per l'alias e le risorse della chiave KMS. Per informazioni sulla creazione di alias in AWS KMS, vedere [Creare alias](#).

CloudTrail le voci di registro per questa operazione registrate a partire da dicembre 2022 includono l'ARN della chiave KMS interessata nel `responseElements.keyId` valore, anche se questa operazione non restituisce l'ARN della chiave.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2022-08-14T23:08:31Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateAlias",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",

```

```

    "userAgent": "AWS Internal",
    "requestParameters": {
      "aliasName": "alias/ExampleAlias",
      "targetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
      "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "caec1e0c-ce03-419e-bdab-6ab1f7c57c01",
    "eventID": "2dd6e784-8286-46a6-befd-d64e5a02fb28",
    "readOnly": false,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      },
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:alias/ExampleAlias"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
  }

```

CreateCustomKeyStore

L'esempio seguente mostra una voce di AWS CloudTrail registro generata chiamando l'[CreateCustomKeyStore](#) operazione su un AWS CloudHSM key store. Per informazioni sulla creazione di archivi delle chiavi personalizzate, consultare [Crea un archivio di AWS CloudHSM chiavi](#).

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",

```

```

    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2021-10-21T20:17:32Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateCustomKeyStore",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "customKeyStoreName": "ExampleKeyStore",
    "clusterId": "cluster-1a23b4cdefg"
  },
  "responseElements": {
    "customKeyStoreId": "cks-1234567890abcdef0"
  },
  "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
  "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333"
}

```

CreateGrant

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[CreateGrant](#) operazione. Per informazioni sulla creazione di sovvenzioni in AWS KMS, vedere [Sovvenzioni in AWS KMS](#).

CloudTrail le voci di registro per questa operazione registrate a partire da dicembre 2022 includono l'ARN della chiave KMS interessata nel `responseElements.keyId` valore, anche se questa operazione non restituisce l'ARN della chiave.

```

{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",

```



```

    "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:53:12Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "constraints": {
      "encryptionContextSubset": {
        "ContextKey1": "Value1"
      }
    },
    "operations": ["Encrypt",
"RetireGrant"],
    "granteePrincipal": "EX_PRINCIPAL_ID"
  },
  "responseElements": {
    "grantId": "f020fe75197b93991dc8491d6f19dd3cebb24ee62277a05914386724f3d48758",
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "f3c08808-63bc-11e4-bc2b-4198b6150d5c",
  "eventID": "5d529779-2d27-42b5-92da-91aaea1fc4b5",
  "readOnly": false,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}

```

CreateKey

Questi esempi mostrano le voci di AWS CloudTrail registro relative all'[CreateKey](#) operazione.

Una voce di CreateKey registro può derivare da una CreateKey richiesta o dall'CreateKey operazione relativa a una [ReplicateKey](#) richiesta.

L'esempio seguente mostra una voce di CloudTrail registro per un'[CreateKey](#) operazione che crea una chiave [KMS di crittografia simmetrica](#). Per informazioni sulla creazione di chiavi KMS, consulta [Crea una chiave KMS](#).

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2022-08-10T22:38:27Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "description": "",
    "origin": "EXTERNAL",
    "bypassPolicyLockoutSafetyCheck": false,
    "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "keySpec": "SYMMETRIC_DEFAULT",
    "keyUsage": "ENCRYPT_DECRYPT"
  },
  "responseElements": {
    "keyMetadata": {
      "AWSAccountId": "111122223333",
      "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "arn": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "creationDate": "Aug 10, 2022, 10:38:27 PM",
      "enabled": false,
      "description": "",
      "keyUsage": "ENCRYPT_DECRYPT",
      "keyState": "PendingImport",
      "origin": "EXTERNAL",
      "keyManager": "CUSTOMER",
      "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
      "keySpec": "SYMMETRIC_DEFAULT",
```

```

        "encryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "multiRegion": false
    }
},
"requestID": "1aef6713-0223-4ff7-9a6d-781360521930",
"eventID": "36327b37-f4f6-40a9-92ab-48064ec905a2",
"readOnly": false,
"resources": [
    {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management"
}

```

[L'esempio seguente mostra il CloudTrail registro di un'CreateKeyoperazione che crea una chiave KMS di crittografia simmetrica in un archivio di chiavi.AWS CloudHSM](#)

```

{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-14T17:39:50Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {

```

```

    "keyUsage": "ENCRYPT_DECRYPT",
    "bypassPolicyLockoutSafetyCheck": false,
    "origin": "AWS_CLOUDHSM",
    "keySpec": "SYMMETRIC_DEFAULT",
    "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "customKeyStoreId": "cks-1234567890abcdef0",
    "description": ""
  },
  "responseElements": {
    "keyMetadata": {
      "awsAccountId": "111122223333",
      "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
      "arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
      "creationDate": "Oct 14, 2021, 5:39:50 PM",
      "enabled": true,
      "description": "",
      "keyUsage": "ENCRYPT_DECRYPT",
      "keyState": "Enabled",
      "origin": "AWS_CLOUDHSM",
      "customKeyStoreId": "cks-1234567890abcdef0",
      "cloudHsmClusterId": "cluster-1a23b4cdefg",
      "keyManager": "CUSTOMER",
      "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
      "keySpec": "SYMMETRIC_DEFAULT",
      "encryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
      ],
      "multiRegion": false
    }
  },
  "additionalEventData": {
    "backingKey": "{\"backingKeyId\":\"backing-key-id\"}"
  },
  "requestID": "4f0b185c-588c-4767-9e90-c618f7e13cad",
  "eventID": "c73964b8-703d-49e4-bd9e-f773d0ee1e65",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
    }
  ]
}

```

```

],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management"
}

```

L'esempio seguente mostra il CloudTrail registro di un'CreateKeyoperazione che crea una chiave KMS di crittografia simmetrica in un archivio di chiavi esterno.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2022-09-07T22:37:45Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateKey",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "tags": [],
    "keyUsage": "ENCRYPT_DECRYPT",
    "description": "",
    "origin": "EXTERNAL_KEY_STORE",
    "multiRegion": false,
    "keySpec": "SYMMETRIC_DEFAULT",
    "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "bypassPolicyLockoutSafetyCheck": false,
    "customKeyId": "cks-1234567890abcdef0",
    "xksKeyId": "bb8562717f809024"
  },
  "responseElements": {
    "keyMetadata": {
      "awsAccountId": "111122223333",
      "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",

```

```

      "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "creationDate": "Dec 7, 2022, 10:37:45 PM",
      "enabled": true,
      "description": "",
      "keyUsage": "ENCRYPT_DECRYPT",
      "keyState": "Enabled",
      "origin": "EXTERNAL_KEY_STORE",
      "customKeyStoreId": "cks-1234567890abcdef0",
      "keyManager": "CUSTOMER",
      "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
      "keySpec": "SYMMETRIC_DEFAULT",
      "encryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
      ],
      "multiRegion": false,
      "xksKeyConfiguration": {
        "id": "bb8562717f809024"
      }
    },
    "requestID": "ba197c82-3ac7-487a-8ff4-7736bbbeb1316",
    "eventID": "838ad5f4-5fdd-4044-afd7-4dbd88c6af56",
    "readOnly": false,
    "resources": [
      {
        "accountId": "227179770375",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-east-1:227179770375:key/39c5eb22-
f37c-4956-92ca-89e8f8b57ab2"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
  }
}

```

Decrittografia

Questi esempi mostrano le voci di AWS CloudTrail registro relative all'operazione [Decrypt](#).

La voce di CloudTrail registro di un'Decryptoperazione include sempre il `encryptionAlgorithm` in `requestParameters` anche se l'algoritmo di crittografia non è stato specificato nella richiesta. Il testo cifrato nella richiesta e il testo normale nella risposta sono omessi.

Argomenti

- [Decrittografia con una chiave crittografica simmetrica standard](#)
- [Errore di decrittografia con una chiave crittografica simmetrica standard](#)
- [Decifrare con una chiave KMS in un archivio di chiavi AWS CloudHSM](#)
- [Decrittografia con una chiave KMS in un archivio delle chiavi esterne](#)
- [Errore di decrittografia con una chiave KMS in un archivio delle chiavi esterne](#)
- [Decrittografia con una chiave di crittografia simmetrica standard su una connessione TLS post-quantistica](#)

Decrittografia con una chiave crittografica simmetrica standard

Di seguito è riportato un esempio di voce di CloudTrail registro per un'Decryptoperazione con una chiave di crittografia simmetrica standard.

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2025-05-20T20:45:00Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "encryptionContext": {
```

```

        "Department": "Engineering",
        "Project": "Alpha"
    },
    "responseElements": null,
    "additionalEventData": {
        "keyMaterialId":
"123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0",
    },
    "requestID": "12345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES256-GCM-SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}

```

Errore di decrittografia con una chiave crittografica simmetrica standard

L'esempio seguente di voce di CloudTrail registro registra un'Decryptoperazione non riuscita con una chiave KMS di crittografia simmetrica standard. L'eccezione (`errorCode`) e il messaggio di errore (`errorMessage`) sono inclusi per aiutarti a risolvere l'errore.

In questo caso, la chiave KMS di crittografia simmetrica specificata nella richiesta Decrypt non corrispondeva alla chiave KMS utilizzata per crittografare i dati.

```

{
    "eventVersion": "1.08",
    "userIdentity": {

```



```

    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2022-11-24T18:57:43Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "errorCode": "IncorrectKeyException"
  "errorMessage": "The key ID in the request does not identify a CMK that can perform
this operation.",
  "requestParameters": {
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "encryptionContext": {
      "Department": "Engineering",
      "Project": "Alpha"
    }
  },
  "responseElements": null,
  "requestID": "22345126-30d5-4b28-98b9-9153da559963",
  "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}

```

Decifrare con una chiave KMS in un archivio di chiavi AWS CloudHSM

[L'esempio seguente di voce di CloudTrail registro registra un'Decryptoperazione con una chiave KMS in un archivio di chiavi.AWS CloudHSM](#) Tutte le voci di registro per le operazioni crittografiche con una chiave KMS in un archivio di chiavi personalizzato includono un `additionalEventData` campo con la `e.customKeyStoreId` `backingKeyId` Il valore restituito nel `backingKeyId` campo è l'attributo chiave `id` CloudHSM. `additionalEventData` non è specificato nella richiesta.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2021-10-26T23:41:27Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "requestParameters": {
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "encryptionContext": {
      "Department": "Development",
      "Purpose": "Test"
    }
  },
  "responseElements": null,
  "additionalEventData": {
    "customKeyStoreId": "cks-1234567890abcdef0"
  },
  "requestID": "e1b881f8-2048-41f8-b6cc-382b7857ec61",
  "eventID": "a79603d5-4cde-46fc-819c-a7cf547b9df4",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",

```

```

      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
  }
}

```

Decrittografia con una chiave KMS in un archivio delle chiavi esterne

L'esempio seguente di voce di CloudTrail registro registra un'Decryptoperazione con una chiave KMS in un archivio di chiavi [esterno](#). Oltre a `customKeyId`, il campo `additionalEventData` include l'[ID della chiave esterna](#) (`XksKeyId`). `additionalEventData` non è specificato nella richiesta.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2022-11-24T00:26:58Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "AWS Internal",
  "requestParameters": {
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
    "encryptionContext": {
      "Department": "Engineering",
      "Purpose": "Test"
    }
  },
  "responseElements": null,
  "additionalEventData": {

```

```

    "customKeyStoreId": "cks-9876543210fedcba9",
    "xksKeyId": "abc01234567890fe"
  },
  "requestID": "f1b881f8-2048-41f8-b6cc-382b7857ec61",
  "eventID": "b79603d5-4cde-46fc-819c-a7cf547b9df4",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}

```

Errore di decrittografia con una chiave KMS in un archivio delle chiavi esterne

L'esempio seguente di voce di CloudTrail registro registra una richiesta non riuscita per un'Decryptoperazione con una chiave KMS in un archivio di [chiavi esterno](#). CloudWatch registra le richieste che hanno esito negativo, oltre a quelle riuscite. Quando si registra un errore, la voce di CloudTrail registro include l'eccezione (ErrorCode) e il relativo messaggio di errore (ErrorMessage).

Se la richiesta non riuscita ha raggiunto il proxy dell'archivio delle chiavi esterne, come in questo esempio, puoi utilizzare il valore `requestId` per associare la richiesta non riuscita a una richiesta corrispondente registrata dal proxy, se l'operazione è consentita.

Per informazioni sulle richieste Decrypt negli archivi delle chiavi esterne, consulta [Errori di decrittografia](#).

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  }
}

```

```

},
"eventTime": "2022-11-24T00:26:58Z",
"eventSource": "kms.amazonaws.com",
"eventName": "Decrypt",
"awsRegion": "us-west-2",
"sourceIPAddress": "AWS Internal",
"userAgent": "AWS Internal",
"errorCode": "KMSInvalidStateException",
"errorMessage": "The external key store proxy rejected the request because the
specified ciphertext or additional authenticated data is corrupted, missing, or
otherwise invalid.",
"requestParameters": {
  "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
  "keyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
  "encryptionContext": {
    "Department": "Engineering",
    "Purpose": "Test"
  }
},
"responseElements": null,
"additionalEventData": {
  "customKeyStoreId": "cks-9876543210fedcba9",
  "xksKeyId": "abc01234567890fe"
},
"requestID": "f1b881f8-2048-41f8-b6cc-382b7857ec61",
"eventID": "b79603d5-4cde-46fc-819c-a7cf547b9df4",
"readOnly": true,
"resources": [
  {
    "accountId": "111122223333",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management"
}

```

Decrittografa con una chiave di crittografia simmetrica standard su una connessione TLS post-quantistica

Di seguito è riportato un esempio di CloudTrail registrazione per un'Decryptoperazione con una chiave di crittografia simmetrica standard su una connessione TLS post-quantistica. Il campo KeyExchange nella sezione menziona. tlsDetails X25519MLKEM768 [Si tratta di un algoritmo ibrido che combina Elliptic Curve Diffie-Hellman \(ECDH\) su Curve 25519, un classico algoritmo di scambio di chiavi utilizzato oggi in TLS, con il Module-Lattice-Based Key-Encapsulation Mechanism \(ML-KEM\), un algoritmo di crittografia e definizione delle chiavi a chiave pubblica che il National Institute for Standards and Technology \(NIST\) ha designato come primo algoritmo standard di accordo di chiavi post-quantistiche.](#)

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2025-11-12T15:16:26Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "aws-sdk-java/2.30.22 md/io#async md/http#AwsCommonRuntime ...",
  "requestParameters": {
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "encryptionContext": {
      "Department": "Engineering",
      "Project": "Alpha"
    }
  },
  "responseElements": null,
  "additionalEventData": {
    "keyMaterialId":
    "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
  },
}
```

```

    "requestID": "12345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
      "tlsVersion": "TLSv1.3",
      "cipherSuite": "TLS_AES_256_GCM_SHA384",
      "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com",
      "keyExchange": "X25519MLKEM768"
    }
  }
}

```

DeleteAlias

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[DeleteAlias](#) operazione. Per informazioni sull'eliminazione di archivi, consulta [Eliminare un alias](#).

CloudTrail le voci di registro per questa operazione registrate a partire da dicembre 2022 includono l'ARN della chiave KMS interessata nel `responseElements.keyId` valore, anche se questa operazione non restituisce l'ARN della chiave.

```

{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice",
    "sessionContext": {
      "attributes": {

```

```

        "mfaAuthenticated": "false",
        "creationDate": "2014-11-04T00:52:27Z"
    }
},
"eventTime": "2014-11-04T00:52:27Z",
"eventSource": "kms.amazonaws.com",
"eventName": "DeleteAlias",
"awsRegion": "us-east-1",
"sourceIPAddress": "192.0.2.0",
"userAgent": "AWS Internal",
"requestParameters": {
    "aliasName": "alias/my_alias"
},
"responseElements": {
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
},
"requestID": "d9542792-63bc-11e4-bc2b-4198b6150d5c",
"eventID": "12f48554-bb04-4991-9cfc-e7e85f68eda0",
"readOnly": false,
"resources": [{
    "ARN": "arn:aws:kms:us-east-1:111122223333:alias/my_alias",
    "accountId": "111122223333"
},
{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId": "111122223333"
}],
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}

```

DeleteCustomKeyStore

L'esempio seguente mostra una voce di AWS CloudTrail registro generata chiamando l'[DeleteCustomKeyStore](#) operazione. Per informazioni sulla creazione di archivi delle chiavi personalizzate, consultare [Eliminare un archivio AWS CloudHSM chiavi](#).

```

{
    "eventVersion": "1.08",
    "userIdentity": {

```



```

    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2021-10-21T20:17:32Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DeleteCustomKeyStore",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "customKeyId": "cks-1234567890abcdef0"
  },
  "responseElements": null,
  "additionalEventData": {
    "customKeyName": "ExampleKeyStore",
    "clusterId": "cluster-1a23b4cdefg"
  },
  "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
  "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333"
}

```

DeleteExpiredKeyMaterial

Quando importi materiale chiave in una AWS KMS key (chiave KMS), puoi impostare una data e un'ora di scadenza per quel materiale chiave. AWS KMS registra una voce nel CloudTrail registro quando [importi il materiale chiave](#) (con le impostazioni di scadenza) e quando AWS KMS elimini il materiale chiave scaduto. Per ulteriori informazioni sulla creazione di chiavi KMS con materiale chiave importato, consulta [Importazione di materiale chiave per le AWS KMS chiavi](#).

L'esempio seguente mostra una voce di AWS CloudTrail registro generata quando si AWS KMS elimina il materiale chiave scaduto.

```

{
  "eventVersion": "1.11",

```

```

    "userIdentity": {
      "accountId": "111122223333",
      "invokedBy": "AWS Internal"
    },
    "eventTime": "2025-05-22T19:55:11Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteExpiredKeyMaterial",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "cfa932fd-0d3a-4a76-a8b8-616863a2b547",
    "readOnly": false,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
      "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "keyMaterialId": "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "eventCategory": "Management"
  }

```

DeleteImportedKeyMaterial

Se importi materiale chiave in una chiave KMS, puoi eliminare il materiale chiave importato in qualsiasi momento utilizzando l'[DeleteImportedKeyMaterial](#) operazione. Quando elimini il materiale della chiave importato, lo stato della chiave KMS cambia in PendingImport e la chiave KMS non potrà essere utilizzata in alcuna operazione di crittografia. Per informazioni dettagliate, consultare [Eliminare il materiale chiave importato](#).

L'esempio seguente mostra una voce di AWS CloudTrail registro generata per l'[DeleteImportedKeyMaterial](#) operazione.

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2025-05-20T20:45:08Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DeleteImportedKeyMaterial",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "keyMaterialId":
"123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
  },
  "responseElements": {
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "keyMaterialId":
"123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
  },
  "requestID": "dcf0e82f-dad0-4622-a378-a5b964ad42c1",
  "eventID": "2afbb991-c668-4641-8a00-67d62e1fecbd",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management",
  "tlsDetails": {
```

```

        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES256-GCM-SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}

```

DeleteKey

Questi esempi mostrano la voce di AWS CloudTrail registro generata quando viene eliminata una chiave KMS. Per eliminare una chiave KMS, si utilizza l'[ScheduleKeyDeletion](#) operazione. Dopo la scadenza del periodo di attesa specificato, AWS KMS elimina la chiave KMS e registra una voce come la seguente nel CloudTrail registro per registrare l'evento.

CloudTrail le voci di registro per questa operazione registrate a partire da dicembre 2022 includono l'ARN della chiave KMS interessata nel `responseElements.keyId` valore, anche se questa operazione non restituisce l'ARN della chiave.

Per un esempio della voce di CloudTrail registro relativa all'`ScheduleKeyDeletion` operazione, vedere [ScheduleKeyDeletion](#). Per informazioni sull'eliminazione delle chiavi KMS, consulta [Eliminare un AWS KMS key](#).

L'esempio seguente di voce di CloudTrail registro registra `DeleteKey` l'operazione di una chiave KMS contenente materiale chiave. AWS KMS

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "accountId": "111122223333",
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2020-07-31T00:07:00Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DeleteKey",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "b25f9cda-74e1-4458-847b-4972a0bf9668",
  "readOnly": false,
  "resources": [
    {

```

```

        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
],
"eventType": "AwsServiceEvent",
"recipientAccountId": "111122223333",
"managementEvent": true,
"eventCategory": "Management"
}

```

La seguente voce di CloudTrail registro registra DeleteKey l'operazione di una chiave KMS in un archivio di [chiavi AWS CloudHSM personalizzato](#).

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "accountId": "111122223333",
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2021-10-26T23:41:27Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DeleteKey",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": {
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "additionalEventData": {
    "customKeyStoreId": "cks-1234567890abcdef0",
    "clusterId": "cluster-1a23b4cdefg",
    "backingKeys": "[{\"backingKeyId\": \"backing-key-id\"}]",
    "backingKeysDeletionStatus": "[{\"backingKeyId\": \"backing-key-id\",
\"deletionStatus\": \"SUCCESS\"}]"]
  },
  "eventID": "1234585c-4b0c-4340-ab11-662414b79239",
  "readOnly": false,
  "resources": [
    {

```

```

        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
],
"eventType": "AwsServiceEvent",
"recipientAccountId": "111122223333",
"managementEvent": true,
"eventCategory": "Management"
}

```

DescribeCustomKeyStores

L'esempio seguente mostra una voce di AWS CloudTrail registro generata chiamando [l'DescribeCustomKeyStores](#) operazione. Per informazioni sulla visualizzazione degli archivi delle chiavi personalizzate, consultare [Visualizza un archivio di AWS CloudHSM chiavi](#).

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2021-10-21T20:17:32Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DescribeCustomKeyStores",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "customKeyId": "cks-1234567890abcdef0"
  },
  "responseElements": null,
  "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
  "eventID": "2ea1735f-628d-43e3-b2ee-486d02913a78",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,

```

```
"recipientAccountId": "111122223333"
}
```

DescribeKey

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[DescribeKey](#) operazione. AWS KMS registra una voce come la seguente quando si richiama l'[DescribeKey](#) operazione o si [visualizzano le chiavi KMS](#) nella AWS KMS console. Questa chiamata è il risultato della visualizzazione di una chiave nella console di AWS KMS gestione.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2022-09-26T18:01:36Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DescribeKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": null,
  "requestID": "12345126-30d5-4b28-98b9-9153da559963",
  "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

```
}
```

DisableKey

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[DisableKey](#) operazione. Per informazioni sull'attivazione e la disabilitazione di AWS KMS keys in AWS KMS, vedere [Attivazione e disattivazione delle chiavi](#).

CloudTrail le voci di registro per questa operazione registrate a partire da dicembre 2022 includono l'ARN della chiave KMS interessata nel `responseElements.keyId` valore, anche se questa operazione non restituisce l'ARN della chiave.

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:52:43Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DisableKey",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "12345126-30d5-4b28-98b9-9153da559963",
  "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
  "readOnly": false,
  "resources": [{
    "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId": "111122223333"
  }],
}
```



```
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}
```

DisableKeyRotation

L'esempio seguente mostra una voce di AWS CloudTrail registro generata chiamando l'[DisableKeyRotation](#) operazione. Per ulteriori informazioni sulla rotazione automatica delle chiavi, consulta [Ruotare AWS KMS keys](#).

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2022-09-01T19:31:39Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DisableKeyRotation",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": null,
  "requestID": "d6a9351a-ed6e-4581-88d1-2a9a8a538497",
  "eventID": "6313164c-83aa-4cc3-9e1a-b7c426f7a5b1",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
```

```
"managementEvent": true,  
"recipientAccountId": "111122223333",  
"eventCategory": "Management"  
}
```

DisconnectCustomKeyStore

L'esempio seguente mostra una voce di AWS CloudTrail registro generata chiamando l'[DisconnectCustomKeyStore](#) operazione. Per informazioni sulla disconnessione di un archivio delle chiavi personalizzate, consultare [Disconnetti un archivio AWS CloudHSM chiavi](#).

```
{  
  "eventVersion": "1.08",  
  "userIdentity": {  
    "type": "IAMUser",  
    "principalId": "EX_PRINCIPAL_ID",  
    "arn": "arn:aws:iam::111122223333:user/Alice",  
    "accountId": "111122223333",  
    "accessKeyId": "EXAMPLE_KEY_ID",  
    "userName": "Alice"  
  },  
  "eventTime": "2021-10-21T20:17:32Z",  
  "eventSource": "kms.amazonaws.com",  
  "eventName": "DisconnectCustomKeyStore",  
  "awsRegion": "us-east-1",  
  "sourceIPAddress": "192.0.2.0",  
  "userAgent": "AWS Internal",  
  "requestParameters": {  
    "customKeyId": "cks-1234567890abcdef0"  
  },  
  "responseElements": null,  
  "additionalEventData": {  
    "customKeyName": "ExampleKeyStore",  
    "clusterId": "cluster-1a23b4cdefg"  
  },  
  "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",  
  "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",  
  "readOnly": false,  
  "eventType": "AwsApiCall",  
  "managementEvent": true,  
  "recipientAccountId": "111122223333"  
}
```

EnableKey

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[EnableKey](#) operazione. Per informazioni sull'attivazione e la disattivazione di AWS KMS keys in AWS KMS, vedere..

[Attivazione e disattivazione delle chiavi](#)

CloudTrail le voci di registro per questa operazione registrate a partire da dicembre 2022 includono l'ARN della chiave KMS interessata nel `responseElements.keyId` valore, anche se questa operazione non restituisce l'ARN della chiave.

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:52:20Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "EnableKey",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "d528a6fb-63bc-11e4-bc2b-4198b6150d5c",
  "eventID": "be393928-3629-4370-9634-567f9274d52e",
  "readOnly": false,
  "resources": [{
    "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

```
}
```

EnableKeyRotation

L'esempio seguente mostra una voce di AWS CloudTrail registro di una chiamata all'[EnableKeyRotation](#) operazione. Per un esempio della voce di CloudTrail registro che viene scritta quando la chiave viene ruotata, vedete [RotateKey](#). Per informazioni sulla rotazione di AWS KMS keys, consulta [Ruotare AWS KMS keys](#).

Note

[rotation-period](#) È un parametro di richiesta opzionale. Se non specificate un periodo di rotazione quando abilitate la rotazione automatica delle chiavi, il valore predefinito è 365 giorni.

CloudTrail le voci di registro per questa operazione registrate a partire da dicembre 2022 includono l'ARN della chiave KMS interessata nel `responseElements.keyId` valore, anche se questa operazione non restituisce l'ARN della chiave.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2020-07-25T23:41:56Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "EnableKeyRotation",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "rotationPeriodInDays": 180
  },
  "responseElements": {
```

```

    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "81f5b794-452b-4d6a-932b-68c188165273",
  "eventID": "fefc43a7-8e06-419f-bcab-b3bf18d6a401",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}

```

Crittografa

L'esempio seguente mostra una voce di AWS CloudTrail registro per l'operazione [Encrypt](#).

```

{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2025-05-20T20:46:16Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Encrypt",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "encryptionContext": {
      "Department": "Engineering"
    }
  },

```

```

    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
  },
  "responseElements": null,
  "additionalEventData": {
    "keyMaterialId":
"123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
  },
  "requestID": "f3423043-63bc-11e4-bc2b-4198b6150d5c",
  "eventID": "91235988-eb87-476a-ac2c-0cdc244e6dca",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333"
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_256_GCM_SHA384",
    "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
  }
}

```

GenerateDataKey

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[GenerateDataKey](#) operazione.

```

{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },

```

```

    "eventTime": "2025-05-20T20:46:16Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
      "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "keySpec": "AES_256",
      "encryptionContext": {
        "Department": "Engineering",
        "Project": "Alpha"
      }
    },
    "responseElements": null,
    "additionalEventData": {
      "keyMaterialId":
"123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0",
    },
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [{
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
    "tlsDetails": {
      "tlsVersion": "TLSv1.3",
      "cipherSuite": "TLS_AES_256_GCM_SHA384",
      "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
    }
  }
}

```

GenerateDataKeyPair

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[GenerateDataKeyPair](#) operazione. In questo esempio viene registrata un'operazione che genera una coppia di chiavi RSA crittografate con una AWS KMS key di crittografia simmetrica.

```

{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2025-05-20T20:46:16Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKeyPair",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyPairSpec": "RSA_3072",
    "encryptionContext": {
      "Project": "Alpha"
    }
  },
  "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
},
"responseElements": null,
"additionalEventData": {
  "keyMaterialId":
"123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
},
"requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
"eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
"readOnly": true,
"resources": [
  {
    "accountId": "111122223333",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333"
"eventCategory": "Management",

```



```

    "tlsDetails": {
      "tlsVersion": "TLSv1.3",
      "cipherSuite": "TLS_AES_256_GCM_SHA384",
      "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
  }
}

```

GenerateDataKeyPairWithoutPlaintext

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[GenerateDataKeyPairWithoutPlaintext](#) operazione. In questo esempio viene registrata un'operazione che genera una coppia di chiavi RSA crittografata con una AWS KMS key di crittografia simmetrica.

```

{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2025-05-20T20:46:16Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKeyPairWithoutPlaintext",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyPairSpec": "RSA_4096",
    "encryptionContext": {
      "Index": "5"
    }
  },
  "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
},
"responseElements": null,
"additionalEventData": {
  "keyMaterialId":
"123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
},
"requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",

```

```

    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
    "readOnly": true,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
  },
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_256_GCM_SHA384",
    "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
  }
}

```

GenerateDataKeyWithoutPlaintext

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[GenerateDataKeyWithoutPlaintext](#) operazione.

```

{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2025-05-20T20:46:16Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKeyWithoutPlaintext",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {

```

```

    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "keySpec": "AES_256",
    "encryptionContext": {
      "Project": "Alpha"
    }
  },
  "responseElements": null,
  "additionalEventData": {
    "keyMaterialId":
"123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
  },
  "requestID": "d6b8e411-63bc-11e4-bc2b-4198b6150d5c",
  "eventID": "f7734272-9ec5-4c80-9f36-528ebbe35e4a",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333"
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_256_GCM_SHA384",
    "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
  }
}

```

GenerateMac

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[GenerateMac](#) operazione.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  }
}

```

```

    },
    "eventTime": "2022-12-23T19:26:54Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateMac",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
      "macAlgorithm": "HMAC_SHA_512",
      "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [
      {
        "accountId": "111122223333",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
  }
}

```

GenerateRandom

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[GenerateRandom](#) operazione. Poiché questa operazione non utilizza un AWS KMS key, il `resources` campo è vuoto.

```

{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },

```

```
"eventTime": "2014-11-04T00:52:37Z",
"eventSource": "kms.amazonaws.com",
"eventName": "GenerateRandom",
"awsRegion": "us-east-1",
"sourceIPAddress": "192.0.2.0",
"userAgent": "AWS Internal",
"requestParameters": null,
"responseElements": null,
"requestID": "df1e3de6-63bc-11e4-bc2b-4198b6150d5c",
"eventID": "239cb9f7-ae05-4c94-9221-6ea30eef0442",
"readOnly": true,
"resources": [],
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}
```

GetKeyPolicy

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[GetKeyPolicy](#) operazione. Per informazioni sulla visualizzazione della policy delle chiavi per una chiave KMS, consulta [Visualizza una politica chiave](#).

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:50:30Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GetKeyPolicy",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "policyName": "default"
  },
  "responseElements": null,
}
```

```

    "requestID": "93746dd6-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "4aa7e4d5-d047-452a-a5a6-2cce282a7e82",
    "readOnly": true,
    "resources": [{
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }

```

GetKeyRotationStatus

L'esempio seguente mostra una voce di AWS CloudTrail registro per l'[GetKeyRotationStatus](#) operazione. Per informazioni sulla rotazione automatica e su richiesta del materiale chiave per una chiave KMS, vedere. [Ruotare AWS KMS keys](#)

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2024-02-20T19:16:45Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GetKeyRotationStatus",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": null,
  "requestID": "12f9b7e8-49b9-4c1c-a7e3-34ac0cdf0467",
  "eventID": "3d082126-9e7d-4167-8372-a6cfcbed4be6",
  "readOnly": true,
  "resources": [
    {

```

```

        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management"
"tlsDetails": {
    "tlsVersion": "TLSv1.2",
    "cipherSuite": "ECDHE-RSA-AES256-GCM-SHA384",
    "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
}
}

```

GetParametersForImport

L'esempio seguente mostra una voce di AWS CloudTrail registro generata quando si utilizza l'[GetParametersForImport](#) operazione. Questa operazione restituisce la chiave pubblica e il token di importazione utilizzati durante l'importazione del materiale della chiave in una chiave KMS. La stessa CloudTrail voce viene registrata quando si utilizza l'GetParametersForImport operazione o si utilizza la AWS KMS console per [scaricare la chiave pubblica e importare il token](#).

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2020-07-25T23:58:23Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GetParametersForImport",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",

```

```

    "wrappingAlgorithm": "RSAES_OAEP_SHA_256",
    "wrappingKeySpec": "RSA_2048"
  },
  "responseElements": null,
  "requestID": "b5786406-e3c7-43d6-8d3c-6d5ef96e2278",
  "eventID": "4023e622-0c3e-4324-bdef-7f58193bba87",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}

```

ImportKeyMaterial

L'esempio seguente mostra una voce di AWS CloudTrail registro generata quando si utilizza l'[ImportKeyMaterial](#) operazione. La stessa CloudTrail voce viene registrata quando si utilizza l'ImportKeyMaterial operazione o si utilizza la AWS KMS console per [importare materiale chiave](#) in un AWS KMS key.

CloudTrail le voci di registro per questa operazione registrate a partire da dicembre 2022 includono l'ARN della chiave KMS interessata nel `responseElements.keyId` valore, anche se questa operazione non restituisce l'ARN della chiave.

```

{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2025-05-21T05:42:31Z",
  "eventSource": "kms.amazonaws.com",

```



```

    "eventName": "ImportKeyMaterial",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
      "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "validTo": "May 21, 2025, 5:47:45 AM",
      "expirationModel": "KEY_MATERIAL_EXPIRES",
      "importType": "NEW_KEY_MATERIAL",
      "keyMaterialDescription": "ExampleKeyMaterialA"
    },
    "responseElements": {
      "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "keyMaterialId": "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0",
    },
    "requestID": "89e10ee7-a612-414d-95a2-a128346969fd",
    "eventID": "c7abd205-a5a2-4430-bbfa-fc10f3e2d79f",
    "readOnly": false,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
      "tlsVersion": "TLSv1.3",
      "cipherSuite": "TLS_AES_256_GCM_SHA384",
      "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
  }
}

```

ListAliases

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[ListAliases](#) operazione. Poiché questa operazione non utilizza alcun alias particolare oppure AWS KMS key il resources

campo è vuoto. Per informazioni sulla visualizzazione degli alias in AWS KMS, vedere. [Trova il nome dell'alias e l'alias ARN per una chiave KMS](#)

```
{  
    "eventVersion": "1.02",  
    "userIdentity": {  
        "type": "IAMUser",  
        "principalId": "EX_PRINCIPAL_ID",  
        "arn": "arn:aws:iam::111122223333:user/Alice",  
        "accountId": "111122223333",  
        "accessKeyId": "EXAMPLE_KEY_ID",  
        "userName": "Alice"  
    },  
    "eventType": "AwsApiCall",  
    "eventName": "ListAliases",  
    "sourceIPAddress": "192.0.2.0",  
    "userAgent": "AWS Internal",  
    "requestParameters": {  
        "limit": 5,  
        "marker": ""  
    },  
    "responseElements": null,  
    "requestID": "bfef6c19-63bc-11e4-bc2b-4198b6150d5c",  
    "eventId": "a27dda7b-76f1-4ac3-8b40-42dfba77bcd6",  
    "readOnly": true,  
    "resources": [],  
    "recipientAccountId": "111122223333"  
}
```

ListGrants

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[ListGrant](#) operazione. Per informazioni sulle sovvenzioni in AWS KMS, vedere [Sovvenzioni in AWS KMS](#).

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
```

```

    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:52:49Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "ListGrants",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "marker":
"eyJncmFudElkIjoibWY4M2U2ZmM0YTY2NDgxYjQ2YzcyMTdhM2Y4YmQwMDFkZDNiYmQ1MGVlYTMyY2RmOWFiNWY1Nzc1N\n\
\u003d\u003d",
    "limit": 10
  },
  "responseElements": null,
  "requestID": "e5c23960-63bc-11e4-bc2b-4198b6150d5c",
  "eventID": "d24380f5-1b20-4253-8e92-dd0492b3bd3d",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}

```

ListKeyRotations

L'esempio seguente mostra una voce di AWS CloudTrail registro per l'[ListKeyRotations](#) operazione. Per informazioni sulla rotazione automatica e su richiesta del materiale chiave per una chiave KMS, vedere. [Ruotare AWS KMS keys](#)

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
```

```

    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2025-05-21T05:42:35Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "ListKeyRotations",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "includeKeyMaterial": "ALL_KEY_MATERIAL"
  },
  "responseElements": null,
  "requestID": "99c88d32-f2db-455e-8a9a-23855258a452",
  "eventID": "8ce0e74b-b9c7-45a2-96ef-83136d38068e",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.2",
    "cipherSuite": "ECDHE-RSA-AES256-GCM-SHA384",
    "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
  }
}

```

PutKeyPolicy

L'esempio seguente mostra una voce di AWS CloudTrail registro generata chiamando l'[PutKeyPolicy](#) operazione. Per informazioni sull'aggiornamento di una policy delle chiavi, consulta [Modificare una policy della chiave](#).

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2022-09-01T20:06:16Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "PutKeyPolicy",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "policyName": "default",
    "policy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Id\" : \"key-default-1\",\n  \"Statement\" : [ {\n    \"Sid\" : \"Enable IAM User Permissions\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" : {\n      \"AWS\" : \"arn:aws:iam::111122223333:root\"\n    },\n    \"Action\" : \"kms:*\",\n    \"Resource\" : \"*\"\n  } ]\n}",
    "bypassPolicyLockoutSafetyCheck": false
  },
  "responseElements": null,
  "requestID": "7bb906fa-dc21-4350-b65c-808ff0f72f55",
  "eventID": "c217db1f-903f-4a2f-8f88-9580182d6313",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}
```

ReEncrypt

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[ReEncrypt](#) operazione. Il `resources` campo di questa voce di registro ne specifica due AWS KMS keys, la chiave KMS di origine e la chiave KMS di destinazione, in quest'ordine.

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2025-05-22T19:34:55Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "ReEncrypt",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "sourceEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
    "sourceEncryptionContext": {
      "Project": "Alpha",
      "Department": "Engineering"
    },
    "destinationKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "destinationEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
    "destinationEncryptionContext": {
      "Level": "3A"
    }
  },
  "responseElements": null,
  "additionalEventData": {
    "destinationKeyMaterialId":
"96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068",
    "sourceKeyMaterialId":
"123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
  },
  "requestID": "03769fd4-acf9-4b33-adf3-2ab8ca73aadf",
  "eventID": "542d9e04-0e8d-4e05-bf4b-4bdeb032e6ec",
}
```

```

    "readOnly": true,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      },
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
  },
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_256_GCM_SHA384",
    "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
  }
}

```

ReplicateKey

L'esempio seguente mostra una voce di AWS CloudTrail registro generata chiamando l'operazione. [ReplicateKey](#) Una ReplicateKey richiesta genera un'ReplicateKeyoperazione e un'[CreateKey](#)operazione.

Per informazioni sulla replica di chiavi in più Regioni, consulta [Crea chiavi di replica multiregionali](#).

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  }
}

```

```

},
"eventTime": "2020-11-18T01:29:18Z",
"eventSource": "kms.amazonaws.com",
"eventName": "ReplicateKey",
"awsRegion": "us-east-1",
"sourceIPAddress": "192.0.2.0",
"userAgent": "AWS Internal",
"requestParameters": {
  "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "replicaRegion": "us-west-2",
  "bypassPolicyLockoutSafetyCheck": false,
  "description": ""
},
"responseElements": {
  "replicaKeyMetadata": {
    "awsAccountId": "111122223333",
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "creationDate": "Nov 18, 2020, 1:29:18 AM",
    "enabled": false,
    "description": "",
    "keyUsage": "ENCRYPT_DECRYPT",
    "keyState": "Creating",
    "origin": "AWS_KMS",
    "keyManager": "CUSTOMER",
    "keySpec": "SYMMETRIC_DEFAULT",
    "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "encryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "multiRegion": true,
    "multiRegionConfiguration": {
      "multiRegionKeyType": "REPLICA",
      "primaryKey": {
        "arn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "region": "us-east-1"
      },
      "replicaKeys": [
        {
          "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "region": "us-west-2"
        }
      ]
    }
  }
}

```



```

    }
  ]
}
},
  "replicaPolicy": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement\": [{\n    \"Effect\": \"Allow\",\n    \"Principal\": {\"AWS\": \"arn:aws:iam::123456789012:user/Alice\"},\n    \"Action\": \"kms:*\",\n    \"Resource\": \"*\"\n  }, {\n    \"Effect\": \"Allow\",\n    \"Principal\": {\"AWS\": \"arn:aws:iam::012345678901:user/Bob\"},\n    \"Action\": \"kms:CreateGrant\",\n    \"Resource\": \"*\"\n  }, {\n    \"Effect\": \"Allow\",\n    \"Principal\": {\"AWS\": \"arn:aws:iam::012345678901:user/Charlie\"},\n    \"Action\": \"kms:Encrypt\",\n    \"Resource\": \"*\"}\n  ]\n}",
  "requestID": "abcdef68-63bc-11e4-bc2b-4198b6150d5c",
  "eventID": "fedcba44-6773-4f96-8763-1993aec9ae6a",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}

```

RetireGrant

L'esempio seguente mostra una voce di AWS CloudTrail registro generata chiamando l'[RetireGrant](#) operazione. Per informazioni su come ritirare le concessioni, consulta [Ritirare e revocare le concessioni](#).

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",

```

```

    "userName": "Alice"
  },
  "eventTime": "2022-09-01T19:39:33Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "RetireGrant",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData": {
    "grantId": "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a"
  },
  "requestID": "1d274d57-5697-462c-a004-f25fcc29fa26",
  "eventID": "0771bcfb-3e24-4332-9ac8-e1c06563eecf",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}

```

RevokeGrant

L'esempio seguente mostra una voce di AWS CloudTrail registro generata chiamando l'[RevokeGrant](#) operazione. Per informazioni su come revocare le concessioni, consulta [Ritirare e revocare le concessioni](#).

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",

```

```

    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2022-09-01T19:35:17Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "RevokeGrant",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "grantId": "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a"
  },
  "responseElements": null,
  "requestID": "59d94c03-c5b7-428d-ae6e-f2c4b47d2917",
  "eventID": "07a23a39-6526-4ae2-b31e-d35fbe9e24ee",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}

```

RotateKey

Questi esempi mostrano le voci di AWS CloudTrail registro per le operazioni che ruotano. AWS KMS keys Per informazioni sulla rotazione delle chiavi KMS, consulta [Ruotare AWS KMS keys](#).

L'esempio seguente mostra una voce di CloudTrail registro per l'operazione che ruota una chiave KMS di crittografia simmetrica su cui è abilitata la rotazione automatica delle chiavi. Per informazioni sull'attivazione della rotazione automatica, vedere. [Ruotare AWS KMS keys](#)

Per un esempio della voce di CloudTrail registro che registra l'EnableKeyRotationoperazione, vedere[EnableKeyRotation](#).

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "accountId": "111122223333",
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2025-05-20T20:44:37Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "RotateKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a24b3967-ddad-417f-9b22-2332b918db06",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsServiceEvent",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "serviceEventDetails": {
    "rotationType": "AUTOMATIC",
    "keyOrigin": "AWS_KMS",
    "previousKeyMaterialId": "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0",
    "currentKeyMaterialId": "96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068",
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "eventCategory": "Management"
}
```

L'esempio seguente mostra una voce di CloudTrail registro per una rotazione su richiesta avviata dall'[RotateKeyOnDemand](#) operazione. Per informazioni sulla rotazione delle chiavi KMS di crittografia simmetrica su richiesta, vedere. [Esegui la rotazione dei tasti su richiesta](#)

Per un esempio della voce di CloudTrail registro che registra l'`RotateKeyOnDemand` operazione, vedere. [RotateKeyOnDemand](#)

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "accountId": "111122223333",
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2025-05-20T20:44:37Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "RotateKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "a24b3967-ddad-417f-9b22-2332b918db06",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsServiceEvent",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "serviceEventDetails": {
    "rotationType": "ON_DEMAND",
    "keyOrigin": "EXTERNAL",
    "previousKeyMaterialId": "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0",
    "currentKeyMaterialId": "96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068",
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "eventCategory": "Management"
}
```

RotateKeyOnDemand

L'esempio seguente mostra una voce di AWS CloudTrail registro per l'[RotateKeyOnDemand](#) operazione. Per un esempio della voce di CloudTrail registro che viene scritta quando la chiave viene ruotata, vedere [RotateKey](#). Per ulteriori informazioni sulla rotazione su richiesta del materiale chiave per una chiave KMS, consulta [Esegui la rotazione dei tasti su richiesta](#)

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2024-02-20T17:41:57Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "RotateKeyOnDemand",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "9e1dee86-eb84-42fd-8f25-e3fc7dbb32c8",
  "eventID": "00a09fbc-20d6-4a58-9b92-7da85984ab77",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
}
```

```

    "eventCategory": "Management",
    "tlsDetails": {
      "tlsVersion": "TLSv1.2",
      "cipherSuite": "ECDHE-RSA-AES256-GCM-SHA384",
      "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
    }
  }
}

```

ScheduleKeyDeletion

Questi esempi mostrano le voci di AWS CloudTrail registro relative all'[ScheduleKeyDeletion](#) operazione.

Per un esempio della voce di CloudTrail registro che viene scritta quando la chiave viene eliminata, vedere [DeleteKey](#). Per informazioni sull'eliminazione delle AWS KMS keys, consulta [Eliminare un AWS KMS key](#).

Nell'esempio seguente viene registrata una richiesta ScheduleKeyDeletion di una chiave KMS a Regione singola.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2021-03-23T18:58:30Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "ScheduleKeyDeletion",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "pendingWindowInDays": 20,
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",

```

```

    "keyState": "PendingDeletion",
    "deletionDate": "Apr 12, 2021 18:58:30 PM"
  },
  "requestID": "ee408f36-ea01-422b-ac14-b0f147c68334",
  "eventID": "3c4226b0-1e81-48a8-a333-7fa5f3cbd118",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}

```

Nell'esempio seguente viene registrata una richiesta `ScheduleKeyDeletion` di una chiave KMS multi-Regione con chiavi di replica.

Poiché AWS KMS non eliminerà una chiave multiregionale finché non verranno eliminate tutte le relative chiavi di replica, nel `responseElements` campo, l'`keyState` è `PendingReplicaDeletion` e il `deletionDate` campo vengono omessi.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2021-10-28T17:59:05Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "ScheduleKeyDeletion",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "pendingWindowInDays": 30,

```



```

    "keyId": "mrk-1234abcd12ab34cd56ef1234567890ab"
  },
  "responseElements": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab",
    "keyState": "PendingReplicaDeletion",
    "pendingWindowInDays": 30
  },
  "requestID": "12341411-d846-42a6-a476-b1cbe3011f89",
  "eventID": "abcda5f-396d-494c-9380-0c47860df5f1",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}

```

[L'esempio seguente registra una `ScheduleKeyDeletion` richiesta per una chiave KMS in un AWS CloudHSM archivio di chiavi personalizzato.](#)

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2021-10-26T23:25:25Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "ScheduleKeyDeletion",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",

```

```

    "userAgent": "AWS Internal",
    "requestParameters": {
      "keyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
      "pendingWindowInDays": 30
    },
    "responseElements": {
      "keyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
      "deletionDate": "Nov 2, 2021, 11:25:25 PM",
      "keyState": "PendingDeletion",
      "pendingWindowInDays": 30
    },
    "additionalEventData": {
      "customKeyStoreId": "cks-1234567890abcdef0",
      "clusterId": "cluster-1a23b4cdefg",
      "backingKeys": "[{\"backingKeyId\":\"backing-key-id\"}]"
    },
    "requestID": "abcd9f60-2c9c-4a0b-a456-d5d998f7f321",
    "eventID": "ca01996a-01b0-4edd-bbbb-25d7b6d1a6fa",
    "readOnly": false,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
  }

```

Sign

Questi esempi mostrano le voci di AWS CloudTrail registro per l'operazione [Sign](#).

L'esempio seguente mostra una voce di CloudTrail registro per un'operazione [Sign](#) che utilizza una chiave RSA KMS asimmetrica per generare una firma digitale per un file.

```
{
```

```

    "eventVersion": "1.08",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::111122223333:user/Alice",
      "accountId": "111122223333",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "userName": "Alice"
    },
    "eventTime": "2022-03-07T22:36:44Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Sign",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
      "messageType": "RAW",
      "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
      "signingAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
    },
    "responseElements": null,
    "requestID": "8d0b35e0-46cf-48b9-be99-bf2ebc9ab9fb",
    "eventID": "107b3cac-b125-4556-9702-12a2b9afc7f7",
    "readOnly": true,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
  }

```

SynchronizeMultiRegionKey

L'esempio seguente mostra una voce di AWS CloudTrail registro generata durante AWS KMS la sincronizzazione di una chiave [multiregionale](#). La sincronizzazione prevede chiamate interregionali per copiare le [proprietà condivise](#) di una chiave primaria multiregionale nelle relative chiavi di replica.

AWS KMS sincronizza periodicamente le chiavi multiregionali per assicurare che tutte le chiavi multiregione correlate abbiano lo stesso materiale chiave.

L'elemento della voce di CloudTrail registro include la chiave ARN della chiave primaria multiregione, inclusa la sua. Regione AWS Le chiavi di replica in più Regioni correlate e le relative Regioni non sono elencate in questa voce di log.

CloudTrail le voci di registro per questa operazione registrate a partire da dicembre 2022 includono l'ARN della chiave KMS interessata nel `responseElements.keyId` valore, anche se questa operazione non restituisce l'ARN della chiave.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "accountId": "111122223333",
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2020-11-18T02:04:37Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "SynchronizeMultiRegionKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "12345681-de97-42e9-bed0-b02ae1abd8dc",
  "eventID": "abcdec99-2b5c-4670-9521-ddb8f031e146",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}
```

```
}
```

TagResource

L'esempio seguente mostra una voce di AWS CloudTrail registro di una chiamata all'[TagResource](#) operazione per aggiungere un tag con una chiave di tag Department e un valore di tag pari a. IT

Per un esempio di una voce di UntagResource CloudTrail registro che viene scritta quando la chiave viene ruotata, vedete [UntagResource](#). Per informazioni sull'etichettatura AWS KMS keys, vedere. [Tag in AWS KMS](#)

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2020-07-01T21:19:25Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "TagResource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "tags": [
      {
        "tagKey": "Department",
        "tagValue": "IT"
      }
    ]
  },
  "responseElements": null,
  "requestID": "b942584a-f77d-4787-9feb-b9c5be6e746d",
  "eventID": "0a091b9b-0df5-4cf9-b667-6f2879532b8f",
  "readOnly": false,
  "resources": [
```

```
{
  "accountId": "111122223333",
  "type": "AWS::KMS::Key",
  "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
},
{
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

UntagResource

L'esempio seguente mostra una voce di AWS CloudTrail registro di una chiamata all'[UntagResource](#) operazione di eliminazione di un tag con una chiave di tag diDept.

CloudTrail le voci di registro per questa operazione registrate a partire da dicembre 2022 includono l'ARN della chiave KMS interessata nel `responseElements.keyId` valore, anche se questa operazione non restituisce l'ARN della chiave.

Per un esempio di voce di TagResource CloudTrail registro, vedere. [TagResource](#) Per informazioni sull'assegnazione di tag per AWS KMS keys, consulta [Tag in AWS KMS](#).

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2020-07-01T21:19:19Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "UntagResource",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",

```

```

        "tagKeys": [
            "Dept"
        ],
        "responseElements": {
            "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        "requestID": "cb1d507b-6015-47f4-812b-179713af8068",
        "eventID": "0b00f4b0-036e-411d-aa75-87eb4a35a4b3",
        "readOnly": false,
        "resources": [
            {
                "accountId": "111122223333",
                "type": "AWS::KMS::Key",
                "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
            }
        ],
        "eventType": "AwsApiCall",
        "recipientAccountId": "111122223333"
    }
}

```

UpdateAlias

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[UpdateAlias](#) operazione. L'elemento `resources` include i campi per l'alias e le risorse della chiave KMS. Per informazioni sulla creazione di alias in AWS KMS, vedere [Creare alias](#).

CloudTrail le voci di registro per questa operazione registrate a partire da dicembre 2022 includono l'ARN della chiave KMS interessata nel `responseElements.keyId` valore, anche se questa operazione non restituisce l'ARN della chiave.

```

{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },

```

```

    "eventTime": "2020-11-13T23:18:15Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UpdateAlias",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
      "aliasName": "alias/my_alias",
      "targetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
      "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "d9472f40-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "f72d3993-864f-48d6-8f16-e26e1ae8dff0",
    "readOnly": false,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:alias/my_alias"
      },
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }
}

```

UpdateCustomKeyStore

L'esempio seguente mostra una voce di AWS CloudTrail registro generata chiamando l'[UpdateCustomKeyStore](#) operazione per aggiornare l'ID del cluster per un archivio di chiavi personalizzato. Per informazioni sulla modifica degli archivi delle chiavi personalizzate, consultare [Modifica le impostazioni del AWS CloudHSM key store](#).

```

{
  "eventVersion": "1.08",

```



```

    "userIdentity": {
      "type": "IAMUser",
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::111122223333:user/Alice",
      "accountId": "111122223333",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UpdateCustomKeyStore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
      "customKeyStoreId": "cks-1234567890abcdef0",
      "clusterId": "cluster-1a23b4cdefg"
    },
    "responseElements": null,
    "additionalEventData": {
      "customKeyStoreName": "ExampleKeyStore",
      "clusterId": "cluster-1a23b4cdefg"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
  }
}

```

UpdateKeyDescription

L'esempio seguente mostra una voce di AWS CloudTrail registro generata chiamando l'[UpdateKeyDescription](#) operazione.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",

```

```

    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2022-09-01T19:22:40Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "UpdateKeyDescription",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "description": "New key description"
  },
  "responseElements": null,
  "requestID": "8c3c1f8b-336d-4896-b034-4eb9916bc9b3",
  "eventID": "f5f3d548-2e9e-4658-8427-9dcb5b1ea791",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}

```

UpdatePrimaryRegion

L'esempio seguente mostra le voci di AWS CloudTrail registro generate richiamando l'[UpdatePrimaryRegion](#) operazione su una [chiave multiregionale](#).

L'UpdatePrimaryRegion operazione scrive due voci di CloudTrail registro: una nella regione con la chiave primaria multiregionale convertita in una chiave di replica e una nella regione con una chiave di replica multiarea convertita in una chiave primaria.

CloudTrail le voci di registro per questa operazione registrate a partire da dicembre 2022 includono l'ARN della chiave KMS interessata nel `responseElements.keyId` valore, anche se questa operazione non restituisce l'ARN della chiave.

L'esempio seguente mostra una voce di CloudTrail registro relativa alla regione UpdatePrimaryRegion in cui la chiave multiregionale è passata da una chiave primaria a una chiave di replica (us-west-2). Il campo primaryRegion mostra la Regione che ora ospita la chiave primaria (ap-northeast-1).

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2021-03-10T20:23:37Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "UpdatePrimaryRegion",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
    "primaryRegion": "ap-northeast-1"
  },
  "responseElements": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "ee408f36-ea01-422b-ac14-b0f147c68334",
  "eventID": "3c4226b0-1e81-48a8-a333-7fa5f3cbd118",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
}
```

```
"recipientAccountId": "111122223333"
}
```

L'esempio seguente rappresenta la voce di CloudTrail registro relativa alla regione UpdatePrimaryRegion in cui la chiave Multi-region è passata da una chiave di replica a una chiave primaria (ap-northeast-1). Questa voce di log non identifica la Regione principale precedente.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice",
    "invokedBy": "kms.amazonaws.com"
  },
  "eventTime": "2021-03-10T20:23:37Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "UpdatePrimaryRegion",
  "awsRegion": "ap-northeast-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "arn:aws:kms:ap-northeast-1:111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab",
    "primaryRegion": "ap-northeast-1"
  },
  "responseElements": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "ee408f36-ea01-422b-ac14-b0f147c68334",
  "eventID": "091e6be5-737f-43c6-8431-e3679d6d0619",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "111122223333"
}
```

VerifyMac

L'esempio seguente mostra una voce di AWS CloudTrail registro relativa all'[VerifyMac](#) operazione.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2022-03-31T19:25:54Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "VerifyMac",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "macAlgorithm": "HMAC_SHA_384",
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": null,
  "requestID": "f35da560-edff-4d6e-9b40-fb306fa9ef1e",
  "eventID": "6b464487-6dea-44cd-84ad-225d7450c975",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}
```

Verifica

Questi esempi mostrano le voci di AWS CloudTrail registro relative all'operazione [Verify](#).

L'esempio seguente mostra una voce di CloudTrail registro per un'operazione [Verify](#) che utilizza una chiave RSA KMS asimmetrica per verificare una firma digitale.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2022-03-07T22:50:41Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Verify",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "signingAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256",
    "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "messageType": "RAW"
  },
  "responseElements": null,
  "requestID": "c73ab82a-af82-4750-ae2c-b6bb790e9c28",
  "eventID": "3b4331cd-5b7b-4de5-bf5f-82ec22f0dac0",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
```

```
"eventCategory": "Management"
}
```

Amazon EC2 , esempio uno

L'esempio seguente registra un principale IAM che crea un volume crittografato utilizzando la chiave di volume predefinita nella console di EC2 gestione Amazon.

L'esempio seguente mostra una voce di CloudTrail registro in cui l'utente Alice crea un volume crittografato con una chiave di volume predefinita nella console di EC2 gestione Amazon. Il record del file di EC2 registro include un volumeId campo con un valore di "vol-13439757". Il AWS KMS record contiene un encryptionContext campo con un valore di "aws:ebs:id": "vol-13439757". Analogamente, il principalId e l'accountId tra i due record corrispondono. I record riflettono il fatto che la creazione di un volume crittografato genera una chiave di dati utilizzata per crittografare il contenuto del volume.

```
{
  "Records": [
    {
      "eventVersion": "1.02",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
      },
      "eventTime": "2014-11-05T20:50:18Z",
      "eventSource": "ec2.amazonaws.com",
      "eventName": "CreateVolume",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "AWS Internal",
      "requestParameters": {
        "size": "10",
        "zone": "us-east-1a",
        "volumeType": "gp2",
        "encrypted": true
      },
      "responseElements": {
        "volumeId": "vol-13439757",
```

```

    "size": "10",
    "zone": "us-east-1a",
    "status": "creating",
    "createTime": 1415220618876,
    "volumeType": "gp2",
    "iops": 30,
    "encrypted": true
  },
  "requestID": "1565210e-73d0-4912-854c-b15ed349e526",
  "eventID": "a3447186-135f-4b00-8424-bc41f1a93b4f",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
},
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-05T20:50:19Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKeyWithoutPlaintext",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "&AWS; Internal",
  "requestParameters": {
    "encryptionContext": {
      "aws:ebs:id": "vol-13439757"
    },
    "numberOfBytes": 64,
    "keyId": "alias/aws/ebs"
  },
  "responseElements": null,
  "requestID": "create-123456789012-758241111-1415220618",
  "eventID": "4bd2a696-d833-48cc-b72c-05e61b608399",
  "readOnly": true,
  "resources": [
    {
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",

```



```

        "accountId": "111122223333"
      }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }
]
}

```

Amazon EC2 , secondo esempio

Nell'esempio seguente, un principale IAM che esegue un' EC2 istanza Amazon crea e monta un volume di dati crittografato con una chiave KMS. Questa azione genera più record di CloudTrail log. Per ulteriori informazioni su Amazon EBS e la crittografia, consulta [Requisiti per la crittografia Amazon EBS](#).

Quando il volume viene creato, Amazon EC2, agendo per conto del cliente, ottiene una chiave dati crittografata da AWS KMS (GenerateDataKeyWithoutPlaintext). Quindi crea un'autorizzazione (CreateGrant) che gli consente di decrittografare la chiave dati. Quando il volume è montato, Amazon EC2 chiama AWS KMS per decrittografare la chiave dati (Decrypt).

L'instanceId EC2 istanza Amazon, "i-81e2f56c", viene visualizzata nell'RunInstancesevento. Lo stesso ID dell'istanza qualifica l'elemento granteePrincipal dell'autorizzazione creato ("111122223333:aws:ec2-infrastructure:i-81e2f56c") e il ruolo assunto che è l'entità nella chiamata Decrypt ("arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/i-81e2f56c").

L'[ARN della chiave](#) KMS che protegge il volume di dati appare in tutte e tre le AWS KMS chiamate (CreateGrantGenerateDataKeyWithoutPlaintext, e). arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab Decrypt

```

{
  "Records": [
    {
      "eventVersion": "1.02",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",

```

```
      "userName": "Alice"
    },
    "eventTime": "2014-11-05T21:35:27Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "RunInstances",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
      "instancesSet": {
        "items": [
          {
            "imageId": "ami-b66ed3de",
            "minCount": 1,
            "maxCount": 1
          }
        ]
      },
      "groupSet": {
        "items": [
          {
            "groupId": "sg-98b6e0f2"
          }
        ]
      },
      "instanceType": "m3.medium",
      "blockDeviceMapping": {
        "items": [
          {
            "deviceName": "/dev/xvda",
            "ebs": {
              "volumeSize": 8,
              "deleteOnTermination": true,
              "volumeType": "gp2"
            }
          },
          {
            "deviceName": "/dev/sdb",
            "ebs": {
              "volumeSize": 8,
              "deleteOnTermination": false,
              "volumeType": "gp2",
              "encrypted": true
            }
          }
        ]
      }
    }
  }
}
```

```

    }
  ]
},
"monitoring": {
  "enabled": false
},
"disableApiTermination": false,
"instanceInitiatedShutdownBehavior": "stop",
"clientToken": "XdKUT141516171819",
"ebsOptimized": false
},
"responseElements": {
  "reservationId": "r-5ebc9f74",
  "ownerId": "111122223333",
  "groupSet": {
    "items": [
      {
        "groupId": "sg-98b6e0f2",
        "groupName": "launch-wizard-2"
      }
    ]
  },
  "instancesSet": {
    "items": [
      {
        "instanceId": "i-81e2f56c",
        "imageId": "ami-b66ed3de",
        "instanceState": {
          "code": 0,
          "name": "pending"
        },
        "amiLaunchIndex": 0,
        "productCodes": {

        },
        "instanceType": "m3.medium",
        "launchTime": 1415223328000,
        "placement": {
          "availabilityZone": "us-east-1a",
          "tenancy": "default"
        },
        "monitoring": {
          "state": "disabled"
        }
      }
    ]
  }
}

```

```

    "stateReason": {
      "code": "pending",
      "message": "pending"
    },
    "architecture": "x86_64",
    "rootDeviceType": "ebs",
    "rootDeviceName": "/dev/xvda",
    "blockDeviceMapping": {

    },
    "virtualizationType": "hvm",
    "hypervisor": "xen",
    "clientToken": "XdKUT1415223327917",
    "groupSet": {
      "items": [
        {
          "groupId": "sg-98b6e0f2",
          "groupName": "launch-wizard-2"
        }
      ]
    },
    "networkInterfaceSet": {

    },
    "ebsOptimized": false
  }
]
}
},
"requestID": "41c4b4f7-8bce-4773-bf0e-5ae3bb5cbce2",
"eventID": "cd75a605-2fee-4fda-b847-9c3d330ebaae",
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
},
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },

```

```

    "eventTime": "2014-11-05T21:35:35Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
      "constraints": {
        "encryptionContextSubset": {
          "aws:ebs:id": "vol-f67bafb2"
        }
      },
      "granteePrincipal": "111122223333:aws:ec2-infrastructure:i-81e2f56c",
      "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
      "grantId": "abcde1237f76e4ba7987489ac329bfba6ad343d6f7075dbd1ef191f0120514a"
    },
    "requestID": "41c4b4f7-8bce-4773-bf0e-5ae3bb5cbce2",
    "eventID": "c1ad79e3-0d3f-402a-b119-d5c31d7c6a6c",
    "readOnly": false,
    "resources": [
      {
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
      }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  },
  {
    "eventVersion": "1.02",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::111122223333:user/Alice",
      "accountId": "111122223333",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "userName": "Alice"
    },
    "eventTime": "2014-11-05T21:35:32Z",
    "eventSource": "kms.amazonaws.com",

```

```

    "eventName": "GenerateDataKeyWithoutPlaintext",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
      "encryptionContext": {
        "aws:ebs:id": "vol-f67bafb2"
      },
      "numberOfBytes": 64,
      "keyId": "alias/aws/ebs"
    },
    "responseElements": null,
    "requestID": "create-111122223333-758247346-1415223332",
    "eventID": "ac3cab10-ce93-4953-9d62-0b6e5cba651d",
    "readOnly": true,
    "resources": [
      {
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
      }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  },
  {
    "eventVersion": "1.02",
    "userIdentity": {
      "type": "AssumedRole",
      "principalId": "111122223333:aws:ec2-infrastructure:i-81e2f56c",
      "arn": "arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/
i-81e2f56c",
      "accountId": "111122223333",
      "accessKeyId": "",
      "sessionContext": {
        "attributes": {
          "mfaAuthenticated": "false",
          "creationDate": "2014-11-05T21:35:38Z"
        },
        "sessionIssuer": {
          "type": "Role",
          "principalId": "111122223333:aws:ec2-infrastructure",
          "arn": "arn:aws:iam::111122223333:role/aws:ec2-infrastructure",
          "accountId": "111122223333",

```

```

        "userName": "aws:ec2-infrastructure"
      }
    },
    "eventTime": "2014-11-05T21:35:47Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "requestParameters": {
      "encryptionContext": {
        "aws:ebs:id": "vol-f67bafb2"
      }
    },
    "responseElements": null,
    "requestID": "b4b27883-6533-11e4-b4d9-751f1761e9e5",
    "eventID": "edb65380-0a3e-4123-bbc8-3d1b7cfff49b0",
    "readOnly": true,
    "resources": [
      {
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
      }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }
]
}

```

Monitora le chiavi KMS con Amazon CloudWatch

Puoi monitorare i tuoi dati AWS KMS keys utilizzando [Amazon CloudWatch](#), un AWS servizio che raccoglie ed elabora dati grezzi AWS KMS trasformandoli in metriche leggibili quasi in tempo reale. Questi dati sono registrati per un periodo di due settimane, in modo da accedere a informazioni cronologiche e comprendere meglio l'utilizzo delle chiavi KMS e le loro modifiche nel corso del tempo.

Puoi usare Amazon CloudWatch per avvisarti di eventi importanti, come i seguenti.

- Il materiale chiave importato in una chiave KMS si avvicina alla data di scadenza.
- Viene ancora utilizzata una chiave KMS in attesa di eliminazione.

- Il materiale chiave di una chiave KMS è stato ruotato automaticamente.
- È stata eliminata una chiave KMS.

Puoi anche creare un CloudWatch allarme [Amazon](#) che ti avvisi quando la frequenza delle richieste raggiunge una determinata percentuale del valore di quota. Per i dettagli, consulta [Gestisci le tariffe di richiesta AWS KMS API utilizzando Service Quotas e Amazon CloudWatch](#) nel blog sulla AWS sicurezza.

AWS KMS metriche e dimensioni

AWS KMS predefinisce i CloudWatch parametri di Amazon per semplificare il monitoraggio dei dati critici e la creazione di allarmi. Puoi visualizzare le AWS KMS metriche utilizzando l' CloudWatch API Console di gestione AWS e Amazon.

Questa sezione elenca ogni AWS KMS metrica e le relative dimensioni e fornisce alcune linee guida di base per la creazione di CloudWatch allarmi basati su tali metriche e dimensioni.

Note

Nome del gruppo di dimensioni:

Per visualizzare una metrica nella CloudWatch console Amazon, nella sezione Metriche, seleziona il nome del gruppo di dimensioni. Quindi, puoi filtrare in base a Metric name (Nome parametro). Questo argomento include il nome del parametro e il nome del gruppo di dimensioni per ogni parametro AWS KMS .

Puoi visualizzare le AWS KMS metriche utilizzando l' CloudWatch API Console di gestione AWS e Amazon. Per ulteriori informazioni, consulta [Visualizza i parametri disponibili](#) nella Amazon CloudWatch User Guide.

Argomenti

- [SuccessfulRequest](#)
- [SecondsUntilKeyMaterialExpiration](#)
- [Cloud HSMKey StoreThrottle](#)
- [ExternalKeyStoreThrottle](#)
- [XksProxyCertificateDaysToExpire](#)
- [XksProxyCredentialAge](#)

- [XksProxyErrors](#)
- [XksExternalKeyManagerStates](#)
- [XksProxyLatency](#)

SuccessfulRequest

Il numero di richieste riuscite per operazioni crittografiche su una chiave KMS specifica. Utilizzando la `SuccessfulRequest` metrica, puoi applicare il filtro a livello di chiave all'utilizzo dell'API in. AWS KMS CloudWatch La Sum statistica di questa metrica definisce il numero totale di richieste riuscite durante il periodo.

Utilizza questa metrica per identificare quali chiavi KMS consumano la maggior parte della quota di richieste o contribuiscono maggiormente ai costi delle API. Puoi anche creare un CloudWatch allarme basato sulla `SuccessfulRequest` metrica per avvisarti di modelli di utilizzo anomali delle API AWS KMS . Questi avvisi possono aiutare a identificare flussi di lavoro inefficienti che potrebbero involontariamente superare le quote di richiesta o comportare costi imprevisti.

Dimensioni per **SuccessfulRequest**

Dimensione	Descrizione
KeyArn	Valore per ogni chiave KMS.
Operazione	Valore per ogni operazione AWS KMS API. Questa metrica si applica solo alle operazioni crittografiche.

Per quanto riguarda [ReEncrypt](#) le operazioni, la `SuccessfulRequest` metrica include le dimensioni per le chiavi KMS di origine e di destinazione.

Dimensione	Descrizione
SourceKeyArn	Valore per la chiave KMS che ha decrittografato il testo cifrato.
DestinationKeyArn	Valore per la chiave KMS che ha ricrittografato i dati.

Dimensione	Descrizione
Operazione	Valore per ogni operazione AWS KMS API, in questo caso, ReEncrypt

SecondsUntilKeyMaterialExpiration

Il numero di secondi rimanenti alla scadenza più antica del [materiale chiave importato in una chiave KMS](#). Questo parametro è valido solo per le chiavi KMS con un materiale della chiave importato (un'[origine del materiale della chiave](#) di EXTERNAL) e una data di scadenza.

Usa questa metrica per tenere traccia del tempo rimasto prima che scada il materiale chiave importato con la scadenza più antica. Quando tale periodo scende al di sotto di una soglia definita, è necessario reimportare il materiale chiave con una nuova data di scadenza per mantenere utilizzabile la chiave KMS. Il parametro `SecondsUntilKeyMaterialExpiration` è specifico per una chiave KMS. Non puoi utilizzare questo parametro per monitorare più chiavi KMS o chiavi KMS che potresti creare in futuro. Per informazioni sulla creazione di un CloudWatch allarme per monitorare questa metrica, consulta [Crea un CloudWatch allarme per la scadenza del materiale chiave importato](#).

La statistica più utile per questo parametro è `Minimum`, che indica la quantità minima di tempo rimanente per tutti i punti dati nel periodo statistico specificato. L'unica unità valida per questo periodo è `Seconds`.

Nome del gruppo di dimensioni: Per-Key Metrics (Parametri per chiave)

Dimensioni per `SecondsUntilKeyMaterialExpiration`

Dimensione	Descrizione; correlata a AWS
KeyId	Valore per ogni chiave KMS.

Quando [programmi l'eliminazione](#) di una chiave KMS, AWS KMS applica un periodo di attesa prima di procedere all'eliminazione. È possibile usare il periodo di attesa per assicurarsi che la chiave KMS non sia necessaria ora o in futuro. È inoltre possibile configurare un CloudWatch allarme per avvisare l'utente se una persona o un'applicazione tenta di utilizzare la chiave KMS in un'[operazione crittografica](#) durante il periodo di attesa. Se si riceve una notifica da parte di un allarme, è possibile annullare l'eliminazione della chiave KMS.

Per istruzioni, consultare [Crea un allarme che rileva l'uso di una chiave KMS in attesa di eliminazione](#).

CloudHSMKeyStoreThrottle

Il numero di richieste di operazioni crittografiche sulle chiavi KMS in ogni AWS CloudHSM archivio di chiavi che AWS KMS rallenta (risponde con un). `ThrottlingException` Questa metrica si applica solo agli archivi di chiavi. AWS CloudHSM

[La `CloudHSMKeyStoreThrottle` metrica si applica solo alle chiavi KMS in un AWS CloudHSM archivio di chiavi e solo alle richieste di operazioni crittografiche.](#) AWS KMS [limita queste richieste](#) quando il tasso di richiesta supera la quota di richieste dell'[archivio chiavi personalizzato per il tuo archivio di chiavi](#). AWS CloudHSM Questa metrica include anche la limitazione da parte del cluster. AWS CloudHSM

Nome del gruppo di dimensioni: Keystore Throttle Metrics (Parametri di limitazione del keystore)

Dimensione	Descrizione
CustomKeyStoreId	Valore per ogni archivio di chiavi AWS CloudHSM .
KmsOperation	Valore per ogni operazione AWS KMS API. Questa metrica si applica solo alle operazioni crittografiche sulle chiavi KMS in un AWS CloudHSM archivio di chiavi.
KeySpec	Valore per ogni tipo di chiave KMS. L'unica specifica chiave supportata per le chiavi KMS in un AWS CloudHSM archivio di chiavi è SYMMETRIC_DEFAULT.

ExternalKeyStoreThrottle

Il numero di richieste di operazioni crittografiche sulle chiavi KMS in ogni archivio di chiavi esterno che rallenta (risponde con un). AWS KMS `ThrottlingException` Questo parametro si applica solo agli [archivi delle chiavi esterne](#).

[La `ExternalKeyStoreThrottle` metrica si applica solo alle chiavi KMS in un archivio di chiavi esterno e solo alle richieste di operazioni crittografiche.](#) AWS KMS [limita queste richieste](#) quando il tasso di richiesta supera la [quota di richieste dell'archivio chiavi personalizzato per l'archivio di chiavi esterno](#). Questo parametro non include la limitazione (della larghezza di banda della rete) da parte del proxy dell'archivio delle chiavi esterne o del gestore delle chiavi esterne.

Utilizza questa metrica per rivedere e modificare il valore della quota di richieste di archiviazione chiavi personalizzate. Se questa metrica indica che spesso AWS KMS limita le richieste per queste chiavi KMS, potresti prendere in considerazione la possibilità di richiedere un aumento del valore della quota di richieste di archiviazione chiavi personalizzate. Per ricevere assistenza, consulta [Richiesta di un aumento di quota](#) nella Guida per l'utente delle Service Quotas.

Se ricevi frequentemente errori `KMSInvalidStateException` con un messaggio che descrive il rifiuto della richiesta "a causa di un tasso di richieste molto elevato" o "perché il proxy dell'archivio delle chiavi esterne non ha risposto in tempo", ciò potrebbe indicare che il gestore delle chiavi esterne o il proxy non è in grado di sostenere la frequenza di richieste corrente. Se possibile, riduci il tasso di richiesta. Potresti anche prendere in considerazione la possibilità di richiedere una riduzione del valore della quota di richiesta dell'archivio delle chiavi personalizzate. La riduzione di questo valore di quota potrebbe aumentare la limitazione (e il valore `ExternalKeyStoreThrottle` metrico), ma indica che ciò significa rifiutare rapidamente le richieste in eccesso prima che AWS KMS vengano inviate al proxy dell'archivio chiavi esterno o al gestore di chiavi esterno. Per richiedere una riduzione della quota, consulta la sezione [Centro Supporto AWS](#) e crea un caso.

Nome del gruppo di dimensioni: Keystore Throttle Metrics (Parametri di limitazione del keystore)

Dimensione	Descrizione
CustomKeyStoreId	Valore per ogni archivio delle chiavi esterne.
KmsOperation	Valore per ogni operazione API. AWS KMS Questa metrica si applica solo alle operazioni crittografiche sulle chiavi KMS in un archivio di chiavi esterno.
KeySpec	Valore per ogni tipo di chiave KMS. L'unica specifica della chiave supportata per le chiavi KMS in un archivio delle chiavi esterne è SYMMETRIC_DEFAULT.

XksProxyCertificateDaysToExpire

Il numero di giorni che mancano alla scadenza del certificato TLS per l'[endpoint proxy dell'archivio delle chiavi esterne](#) (`XksProxyUriEndpoint`). Questo parametro si applica solo agli [archivi delle chiavi esterne](#).

Usa questa metrica per creare un CloudWatch allarme che ti avvisi della scadenza imminente del tuo certificato TLS. Quando il certificato scade, AWS KMS non è possibile comunicare con il proxy

dell'archivio chiavi esterno. Tutti i dati protetti dalle chiavi KMS nell'archivio delle chiavi esterne diventano inaccessibili fino al rinnovo del certificato.

Dal momento che la scadenza di un certificato potrebbe impedirti di accedere alle risorse crittografate, un avviso relativo al certificato potrebbe risultare utile. Configura l'allarme in modo che l'organizzazione abbia la possibilità di rinnovare il certificato prima della sua scadenza.

Nome del gruppo di dimensioni: XKS Proxy Certificate Metrics (Parametri del certificato proxy XKS)

Dimensione	Descrizione
CustomKeyStoreId	Valore per ogni archivio delle chiavi esterne.
CertificateName	Nome del soggetto (CN) nel certificato TLS.

È possibile creare CloudWatch allarmi in base alle metriche degli archivi di chiavi esterni e alle chiavi KMS degli archivi di chiavi esterni. Per istruzioni, consultare [Monitora gli archivi di chiavi esterni](#).

XksProxyCredentialAge

Il numero di giorni trascorsi dell'associazione delle [credenziali di autenticazione proxy](#) (XksProxyAuthenticationCredential) all'archivio delle chiavi esterne. Questo conteggio inizia quando inserisci le credenziali di autenticazione come parte della creazione o dell'aggiornamento dell'archivio delle chiavi esterne. Questo parametro si applica solo agli [archivi delle chiavi esterne](#).

Questo valore è progettato per ricordarti l'età delle credenziali di autenticazione. Tuttavia, poiché il conteggio inizia dal momento in cui associ le credenziali all'archivio delle chiavi esterne e non dalla loro data di creazione, l'indicazione dell'età potrebbe non essere accurata.

Utilizza questa metrica per creare un CloudWatch allarme che ti ricordi di ruotare le credenziali di autenticazione proxy dell'archivio chiavi esterno.

Nome del gruppo di dimensioni: Per-Keystore Metrics (Parametri per keystore)

Dimensione	Descrizione
CustomKeyStoreId	Valore per ogni archivio delle chiavi esterne.

Puoi creare CloudWatch allarmi in base alle metriche per gli archivi di chiavi esterni e le chiavi KMS negli archivi di chiavi esterni. Per istruzioni, consultare [Monitora gli archivi di chiavi esterni](#).

XksProxyErrors

Il numero di eccezioni relative alle AWS KMS richieste al proxy dell'archivio chiavi [esterno](#). Questo conteggio include le eccezioni a cui restituisce il proxy dell'archivio chiavi esterno AWS KMS e gli errori di timeout che si verificano quando il proxy dell'archivio chiavi esterno non risponde AWS KMS entro l'intervallo di timeout di 250 millisecondi. Questo parametro si applica solo agli [archivi delle chiavi esterne](#).

Utilizza questo parametro per tenere traccia del tasso di errore delle chiavi KMS nell'archivio delle chiavi esterne. Rivela gli errori più frequenti, in modo da consentirti di assegnare priorità diverse agli interventi tecnici. Ad esempio, le chiavi KMS che generano alti tassi di errori irreversibili potrebbero indicare un problema con la configurazione dell'archivio delle chiavi esterne. Per visualizzare la configurazione dell'archivio delle chiavi esterne, consulta [Visualizza gli archivi di chiavi esterni](#). Per modificare le impostazioni dell'archivio delle chiavi esterne, consulta [Modifica delle proprietà dell'archivio chiavi esterno](#).

Nome del gruppo di dimensioni: XKS Proxy Error Metrics (Parametri di errore del proxy XKS)

Dimensione	Descrizione
CustomKeyStoreId	Valore per ogni archivio delle chiavi esterne.
KmsOperation	Valore per ogni operazione AWS KMS API che ha generato una richiesta al proxy XKS.
XksOperation	Valore per ogni operazione API proxy dell'archivio delle chiavi esterne .

Dimensione	Descrizione
KeySpec	Valore per ogni tipo di chiave KMS. L'unica specifica della chiave supportata per le chiavi KMS in un archivio delle chiavi esterne è SYMMETRIC_DEFAULT.
ErrorType	Valori: • Errori non irreversibili: possono essere temporanei, come gli errori di rete. • Errori irreversibili: possono indicare un problema con la configurazione dell'archivio delle chiavi personalizzate o con i componenti esterni. • N/D: la richiesta è andata a buon fine; nessun errore
Exception Name	Valori: • Nome dell'eccezione • Nessuno: la richiesta è andata a buon fine; nessun errore

È possibile creare CloudWatch allarmi in base alle metriche degli archivi di chiavi esterni e alle chiavi KMS degli archivi di chiavi esterni. Per istruzioni, consultare [Monitora gli archivi di chiavi esterni](#).

XksExternalKeyManagerStates

Conteggio del numero di [istanze del gestore delle chiavi esterne](#) in ciascuno dei seguenti stati di integrità: Active, Degraded e Unavailable. Le informazioni per questo parametro provengono dal proxy associato a ogni archivio delle chiavi esterne. Questo parametro si applica solo agli [archivi delle chiavi esterne](#).

Di seguito sono riportati gli stati di integrità delle istanze del gestore delle chiavi esterne associate a un archivio delle chiavi esterne. Ogni proxy dell'archivio delle chiavi esterne potrebbe utilizzare indicatori diversi per misurare gli stati di integrità del gestore delle chiavi esterne. Per ulteriori informazioni, consulta la documentazione del proxy dell'archivio delle chiavi esterne.

- Active: il gestore delle chiavi esterne è integro.
- Degraded: il gestore delle chiavi esterne non è integro, ma può comunque servire il traffico
- Unavailable: il gestore delle chiavi esterne non è in grado di servire il traffico.

Utilizza questa metrica per creare un CloudWatch allarme che ti avvisi in caso di istanze di key manager esterne danneggiate e non disponibili. Per determinare lo stato di ogni istanza, consulta i log del proxy dell'archivio delle chiavi esterne.

Nome del gruppo di dimensioni: XKS External Key Manager Metrics (Parametri del gestore chiavi esterne di XKS)

Dimensione	Descrizione
CustomKeyStoreId	Valore per ogni archivio delle chiavi esterne.
XksExternalKeyManagerState	Valore per ogni stato di integrità.

Puoi creare CloudWatch allarmi in base alle metriche per gli archivi di chiavi esterni e le chiavi KMS negli archivi di chiavi esterni. Per istruzioni, consultare [Monitora gli archivi di chiavi esterni](#).

XksProxyLatency

Il numero di millisecondi necessari a un proxy dell'archivio delle chiavi esterne per rispondere a una richiesta AWS KMS. Se la richiesta è scaduta, il valore registrato è il limite di timeout di 250 millisecondi. Questo parametro si applica solo agli [archivi delle chiavi esterne](#).

Utilizza questo parametro per valutare le prestazioni del proxy dell'archivio delle chiavi esterne e del gestore delle chiavi esterne. Ad esempio, se il proxy è spesso prossimo al timeout per le operazioni di crittografia e decrittografia, rivolgiti all'amministratore del proxy.

Le risposte lente potrebbero anche indicare che il gestore delle chiavi esterno non è in grado di gestire il traffico di richieste corrente. AWS KMS consiglia che il gestore delle chiavi esterno sia in grado di gestire fino a 1800 richieste di operazioni crittografiche al secondo. Se il gestore delle chiavi esterne non è in grado di gestire 1.800 richieste al secondo, prendi in considerazione la possibilità di richiedere una riduzione della [quota di richieste per le chiavi KMS in un archivio delle chiavi personalizzate](#). Le richieste relative alle operazioni di crittografia che utilizzano le chiavi KMS nell'archivio delle chiavi esterne anticiperanno rapidamente l'errore (fail fast) con un'[eccezione di limitazione](#) (della larghezza di banda della rete), anziché essere elaborate e successivamente rifiutate dal proxy o dal gestore delle chiavi esterne.

Nome del gruppo di dimensioni: XKS Proxy Latency Metrics (Parametri di latenza del proxy XKS)

Dimensione	Descrizione
CustomKeyStoreId	Valore per ogni archivio delle chiavi esterne.
KmsOperation	Valore per ogni operazione AWS KMS API che ha generato una richiesta al proxy XKS.
XksOperation	Valore per ogni operazione API proxy dell'archivio delle chiavi esterne .
KeySpec	Valore per ogni tipo di chiave KMS. L'unica specificata della chiave supportata per le chiavi KMS in un archivio delle chiavi esterne è SYMMETRIC_DEFAULT.

È possibile creare CloudWatch allarmi in base alle metriche degli archivi di chiavi esterni e alle chiavi KMS degli archivi di chiavi esterni. Per istruzioni, consulta [Monitora gli archivi di chiavi esterni](#).

Crea un CloudWatch allarme per la scadenza del materiale chiave importato

Puoi creare un CloudWatch avviso che ti avvisi quando il materiale chiave importato in una chiave KMS si avvicina alla scadenza. Ad esempio, l'allarme può avvisarti quando mancano meno di 30 giorni alla scadenza.

Quando [importi il materiale della chiave in una chiave KMS](#), puoi specificare una data e un'ora in cui tale materiale scade. Quando il materiale chiave scade, AWS KMS elimina il materiale chiave e la chiave KMS diventa inutilizzabile. Per utilizzare di nuovo la chiave KMS, devi [importare nuovamente lo stesso materiale della chiave](#). Tuttavia, se reimporti il materiale chiave prima della scadenza, è possibile evitare di interrompere i processi che utilizzano quella chiave KMS.

Questo allarme utilizza la [SecondsUntilKeyMaterialExpiresmetrica](#) AWS KMS pubblicata su CloudWatch per le chiavi KMS con materiale chiave importato che scade. Ogni allarme utilizza questo parametro per monitorare il materiale chiave importato per una determinata chiave KMS. Non puoi creare un singolo allarme per tutte le chiavi KMS con materiale chiave in scadenza o un allarme per le chiavi KMS che potresti creare in futuro.

Requisiti

Le seguenti risorse sono necessarie per un CloudWatch allarme che monitora la scadenza del materiale chiave importato.

- Una chiave KMS con materiale della chiave importato.
- Argomento Amazon SNS Per maggiori dettagli, consulta l'[argomento Creating an Amazon SNS](#) nella Amazon CloudWatch User Guide.

Creazione dell'allarme

Segui le istruzioni riportate in [Creare un CloudWatch allarme basato su una soglia statica](#) utilizzando i seguenti valori obbligatori. Per gli altri campi, accetta i valori predefiniti e immetti i nomi come richiesto.

Campo	Valore
Seleziona parametro	Scegli KMS, quindi scegli Per-Key Metrics (Parametri per chiave). Scegli la riga con la chiave KMS e il parametro <code>SecondsUntilKeyMaterialExpires</code>. Quindi, scegli Seleziona parametro. L'elenco Metrics (Parametri) visualizza il parametro <code>SecondsUntilKeyMaterialExpires</code> solo per le chiavi KMS con materiale della chiave importato. Se non disponi di chiavi KMS con queste proprietà nell'account e nella regione, questo elenco è vuoto.
Statistic	Minimo
Periodo	1 minuto
Tipo di soglia	Statico
Quando...	Ogni volta che <i>metric-name</i> è maggiore di 1

Crea CloudWatch allarmi per archivi di chiavi esterni

Puoi creare CloudWatch allarmi Amazon in base a parametri di archiviazione di chiavi esterne per avvisarti quando il valore di una metrica supera una soglia specificata. L'allarme può inviare il messaggio a un argomento di [Amazon Simple Notification Service \(Amazon SNS\)](#) o a una politica

di [Amazon Amazon EC2 Auto EC2 Scaling](#). Per informazioni dettagliate sugli CloudWatch allarmi, consulta [Using Amazon CloudWatch alarms](#) nella Amazon CloudWatch User Guide.

Prima di creare un CloudWatch allarme Amazon, è necessario un argomento su Amazon SNS. Per maggiori dettagli, consulta l'[argomento Creating an Amazon SNS](#) nella Amazon CloudWatch User Guide.

Argomenti

- [Crea un allarme per la scadenza del certificato](#)
- [Crea un allarme per il timeout della risposta](#)
- [Crea un allarme per errori ripetibili](#)
- [Crea un allarme per errori irreparabili](#)

Crea un allarme per la scadenza del certificato

Questo allarme utilizza la [XksProxyCertificateDaysToExpire](#) metrica AWS KMS pubblicata per CloudWatch registrare la scadenza prevista del certificato TLS associato all'endpoint proxy di archiviazione delle chiavi esterno. Non puoi creare un singolo allarme per tutti gli archivi delle chiavi esterne dell'account o un allarme per gli archivi delle chiavi esterne che è possibile creare in futuro.

Ti consigliamo di impostare l'allarme in modo che ti avvisi 10 giorni prima della scadenza del certificato, ma dovresti impostare la soglia più adatta alle tue esigenze.

Creazione dell'allarme

Segui le istruzioni riportate in [Creare un CloudWatch allarme basato su una soglia statica](#) utilizzando i seguenti valori obbligatori. Per gli altri campi, accetta i valori predefiniti e immetti i nomi come richiesto.

Campo	Valore
Seleziona parametro	Scegli KMS, quindi XKS Proxy Certificate Metrics (Parametri del certificato proxy XKS). Seleziona la casella di controllo accanto al valore <code>XksProxyCertificateName</code> da monitorare. Quindi, scegli Seleziona parametro.

Campo	Valore
Statistic	Minimo
Periodo	5 minuti
Tipo di soglia	Statico
Quando...	Ogni volta Lower che XksProxyCertificateDaysToExpire è così 10.

Crea un allarme per il timeout della risposta

Questo allarme utilizza la [XksProxyLatency](#) metrica AWS KMS pubblicata per registrare il numero di millisecondi necessari CloudWatch a un proxy di archiviazione chiavi esterno per rispondere a una richiesta. AWS KMS Non puoi creare un singolo allarme per tutti gli archivi delle chiavi esterne dell'account o un allarme per gli archivi delle chiavi esterne che è possibile creare in futuro.

AWS KMS si aspetta che il proxy dell'archivio chiavi esterno risponda a ogni richiesta entro 250 millisecondi. Ti consigliamo di impostare un allarme per avvisarti quando il proxy dell'archivio delle chiavi esterne impiega più di 200 millisecondi per rispondere, ma dovresti impostare la soglia più adatta alle tue esigenze.

Creazione dell'allarme

Segui le istruzioni in [Creare un CloudWatch allarme basato su una soglia statica](#) utilizzando i seguenti valori obbligatori. Per gli altri campi, accetta i valori predefiniti e immetti i nomi come richiesto.

Campo	Valore
Seleziona parametro	Scegli KMS, quindi XKS Proxy Latency Metrics (Parametri di latenza del proxy XKS). Seleziona la casella di controllo accanto al valore KmsOperation da monitorare. Quindi, scegli Seleziona parametro.
Statistic	Media

Campo	Valore
Periodo	5 minuti
Tipo di soglia	Statico
Quando...	Ogni volta Greater che XksProxyLatency è così200.

Crea un allarme per errori ripetibili

Questo allarme utilizza la [XksProxyErrors](#) metrica AWS KMS pubblicata su per CloudWatch registrare il numero di eccezioni relative AWS KMS alle richieste al proxy dell'archivio chiavi esterno. Non puoi creare un singolo allarme per tutti gli archivi delle chiavi esterne dell'account o un allarme per gli archivi delle chiavi esterne che è possibile creare in futuro.

Gli errori non irreversibili riducono la percentuale di affidabilità e possono indicare errori di rete. Ti consigliamo di impostare un allarme per avvisarti quando vengono registrati più di cinque errori non irreversibili in un minuto, ma dovresti impostare la soglia più adatta alle tue esigenze.

Segui le istruzioni riportate in [Creare un CloudWatch allarme basato su una soglia statica](#) utilizzando i seguenti valori obbligatori. Per gli altri campi, accetta i valori predefiniti e immetti i nomi come richiesto.

Campo	Valore
Seleziona parametro	Scegli la scheda Queries (Query). Scegli AWS/KMS per Namespace. Inserisci SUM(XksProxyErrors) in Metric name (Nome parametro). Inserisci ErrorType = Retryable in Filter by (Filtra per). Seleziona Esegui. Quindi, scegli Seleziona parametro.
Etichetta	<i>Retryable errors</i>
Periodo	1 minuto
Tipo di soglia	Statico

Campo	Valore
Quando...	Ogni volta che q1 è Greater di 5.

Crea un allarme per errori irreparabili

Questo allarme utilizza la [XksProxyErrors](#) metrica AWS KMS pubblicata su per registrare il numero di eccezioni relative CloudWatch alle AWS KMS richieste al proxy dell'archivio chiavi esterno. Non puoi creare un singolo allarme per tutti gli archivi delle chiavi esterne dell'account o un allarme per gli archivi delle chiavi esterne che è possibile creare in futuro.

Gli errori irreversibili possono indicare un problema con la configurazione dell'archivio delle chiavi esterne. Ti consigliamo di impostare un allarme per avvisarti quando vengono registrati più di cinque errori irreversibili in un minuto, ma dovresti impostare la soglia più adatta alle tue esigenze.

Segui le istruzioni riportate in [Creare un CloudWatch allarme basato su una soglia statica](#) utilizzando i seguenti valori obbligatori. Per gli altri campi, accetta i valori predefiniti e immetti i nomi come richiesto.

Campo	Valore
Seleziona parametro	Scegli la scheda Queries (Query). Scegli AWS/KMS per Namespace. Inserisci SUM(XksProxyErrors) in Metric name (Nome parametro). Inserisci ErrorType = Non-retryable in Filter by (Filtra per). Seleziona Esegui. Quindi, scegli Seleziona parametro.
Etichetta	<i>Non-retryable errors</i>
Periodo	1 minuto
Tipo di soglia	Statico
Quando...	Ogni volta che q1 è Greater di 5.

Monitora le chiavi KMS con Amazon EventBridge

Puoi utilizzare Amazon EventBridge (precedentemente Amazon CloudWatch Events) per avvisarti dei seguenti eventi importanti nel ciclo di vita delle tue chiavi KMS.

- Il materiale chiave in una chiave KMS è stato ruotato automaticamente o su richiesta.
- Il materiale chiave importante in una chiave KMS scaduta.
- Una chiave KMS che era stata pianificata per l'eliminazione è stata eliminata.

AWS KMS si integra con Amazon EventBridge per avvisarti di eventi importanti che influiscono sulle tue chiavi KMS. Ogni evento è rappresentato in [JSON \(JavaScriptObject Notation\)](#) e include il nome dell'evento, la data e l'ora in cui si è verificato l'evento e l'evento interessato. Puoi raccogliere questi eventi e stabilire regole che li indirizzino verso uno o più target come AWS Lambda funzioni, argomenti Amazon SNS, code Amazon SQS, flussi in Amazon Kinesis Data Streams o destinazioni integrate.

Per ulteriori informazioni sull'utilizzo EventBridge con altri tipi di eventi, inclusi quelli emessi AWS CloudTrail quando registra una richiesta read/write API, consulta la [Amazon EventBridge User Guide](#).

I seguenti argomenti descrivono gli EventBridge eventi che AWS KMS genera.

Rotazione KMS CMK

AWS KMS supporta la [rotazione automatica e su richiesta](#) del materiale chiave nelle chiavi KMS con crittografia simmetrica.

Ogni volta che AWS KMS ruota il materiale chiave, invia un evento a. `KMS CMK Rotation` EventBridge AWS KMS genera questo evento con il massimo impegno.

Di seguito è illustrato un esempio di questo evento.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "KMS CMK Rotation",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2025-05-23T03:11:54Z",
```

```

"region": "us-west-2",
"resources": [
  "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
],
"detail": {
  "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "key-origin": "AWS_KMS",
  "rotation-type": "ON_DEMAND",
  "previous-key-material-id":
"123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0",
  "current-key-material-id":
"96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068"
}
}

```

Scadenza del materiale chiave importato di KMS

Quando [importi il materiale chiave in una chiave KMS](#), è possibile specificare un'ora in cui tale materiale scade. Quando il materiale chiave scade, AWS KMS elimina il materiale chiave e invia un evento corrispondente KMS Imported Key Material Expiration a EventBridge AWS KMS genera questo evento con la massima diligenza possibile.

Di seguito è illustrato un esempio di questo evento.

```

{
  "version": "0",
  "id": "9da9af57-9253-4406-87cb-7cc400e43465",
  "detail-type": "KMS Imported Key Material Expiration",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2025-05-23T03:11:54Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "key-material-id":
"123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
  }
}

```


Eliminazione di KMS CMK

Quando [programmi l'eliminazione](#) di una chiave KMS, AWS KMS applica un periodo di attesa prima di procedere all'eliminazione. Al termine del periodo di attesa, AWS KMS elimina la chiave KMS e invia un KMS CMK Deletion evento a EventBridge AWS KMS garantisce questo EventBridge evento. A seguito dei tentativi, potrebbero generarsi più eventi in pochi secondi che eliminano la stessa chiave KMS.

Di seguito è illustrato un esempio di questo evento.

```
{
  "version": "0",
  "id": "e9ce3425-7d22-412a-a699-e7a5fc3fbc9a",
  "detail-type": "KMS CMK Deletion",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2025-05-23T03:11:54Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

Alias in AWS KMS

Un alias è un nome descrittivo per una AWS KMS key. Ad esempio, un alias ti consente di fare riferimento a una chiave KMS come `test-key` invece di `1234abcd-12ab-34cd-56ef-1234567890ab`.

[È possibile utilizzare un alias per identificare una chiave KMS nella AWS KMS console, nell'DescribeKeyoperazione e nelle operazioni crittografiche, come Encrypt e GenerateDataKey](#) Gli alias semplificano anche il riconoscimento delle [Chiave gestita da AWS](#). Gli alias per queste chiavi KMS hanno sempre il formato `aws/<service-name>`. Ad esempio, l'alias Chiave gestita da AWS per Amazon DynamoDB è `aws/dynamodb`. Puoi stabilire gli standard per alias simili dei progetti, ad esempio anteporre agli alias il nome di un progetto o di una categoria.

Puoi inoltre consentire e negare l'accesso alle chiavi KMS in base ai relativi alias senza modificare le policy o gestire le concessioni. Questa funzionalità fa parte del AWS KMS supporto per il controllo degli accessi [basato sugli attributi \(ABAC\)](#). Per informazioni dettagliate, vedi [Usa gli alias per controllare l'accesso alle chiavi KMS](#).

Gran parte del potere degli alias deriva dalla tua capacità di modificare la chiave KMS associata a un alias in qualsiasi momento. Gli alias possono rendere il tuo codice più facile da scrivere e gestire. Supponi, ad esempio, di utilizzare un alias per fare riferimento a una chiave KMS particolare e di voler modificare la chiave KMS. In tal caso, basta associare l'alias a un'altra chiave KMS. Non è necessario cambiare il codice.

Gli alias semplificano anche il riutilizzo dello stesso codice in diverse Regioni AWS. Crea alias con lo stesso nome in più regioni e associa ogni alias a una chiave KMS nella sua regione. Quando il codice viene eseguito in ogni regione, l'alias fa riferimento alla relativa chiave KMS associata in quella regione. Per vedere un esempio, consulta [Scopri come utilizzare gli alias nelle tue applicazioni](#).

[È possibile creare un alias per una chiave KMS nella AWS KMS console, utilizzando l>CreateAliasAPI o utilizzando il modello. AWS::KMS::Alias CloudFormation](#)

L' AWS KMS API fornisce il controllo completo degli alias in ogni account e regione. L'API include operazioni per creare un alias ([CreateAlias](#)), visualizzare nomi di alias e alias ARNs ([ListAliases](#)), modificare la chiave KMS associata a un alias ([UpdateAlias](#)) ed eliminare un alias ([DeleteAlias](#)).

Come funzionano gli alias

Scopri come gli alias funzionano in AWS KMS.

Un alias è una risorsa indipendente AWS

Un alias non è una proprietà di una chiave KMS. Le operazioni eseguite sull'alias non influiscono sulla relativa chiave KMS associata. Puoi creare un alias per una chiave KMS e quindi aggiornare l'alias in modo che venga associato a un'altra chiave KMS. È anche possibile eliminare l'alias senza alcun effetto sulla chiave KMS associata. Tuttavia, se si elimina una chiave KMS, vengono eliminati tutti gli alias associati a tale chiave KMS.

Se si specifica un alias come risorsa in una policy IAM, la policy fa riferimento all'alias e non alla chiave KMS associata.

Ogni alias ha due formati

Quando si crea un alias, si specifica il nome dell'alias. AWS KMS crea l'alias ARN per te.

- Un [ARN di alias](#) è un Amazon Resource Name (ARN) che identifica in modo univoco l'alias.

```
# Alias ARN
arn:aws:kms:us-west-2:111122223333:alias/<alias-name>
```

- Il [nome alias](#) è unico solo per un account e per una regione. Nell' AWS KMS API, il nome dell'alias è sempre preceduto da. `alias/` Tale prefisso viene omissso nella console. AWS KMS

```
# Alias name
alias/<alias-name>
```

Gli alias non sono segreti

Gli alias possono essere visualizzati in testo semplice nei log e in CloudTrail altri output. Non includere informazioni riservate o sensibili nel nome dell'alias.

Ogni alias è associato a una chiave KMS alla volta

L'alias e la relativa chiave KMS devono trovarsi nello stesso account e nella stessa regione.

È possibile associare un alias a qualsiasi [chiave gestita dal cliente](#) nella stessa regione. Account AWS Tuttavia, non hai l'autorizzazione per associare un alias a una [Chiave gestita da AWS](#).

Ad esempio, questo [ListAliases](#) output mostra che l'`test-keyalias` è associato esattamente a una chiave KMS di destinazione, rappresentata dalla proprietà. `TargetKeyId`

```
{
  "AliasName": "alias/test-key",
  "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
```

```
"TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
"CreationDate": 1593622000.191,  
"LastUpdatedDate": 1593622000.191  
}
```

Puoi associare più alias alla stessa chiave KMS

Ad esempio, puoi associare gli alias `test-key` e `project-key` alla stessa chiave KMS.

```
{  
  "AliasName": "alias/test-key",  
  "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",  
  "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
  "CreationDate": 1593622000.191,  
  "LastUpdatedDate": 1593622000.191  
},  
{  
  "AliasName": "alias/project-key",  
  "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/project-key",  
  "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
  "CreationDate": 1516435200.399,  
  "LastUpdatedDate": 1516435200.399  
}
```

L'alias deve essere univoco nell'account e nella regione

Ad esempio, è possibile avere un solo alias `test-key` in ogni account e regione. Gli alias rispettano la distinzione tra maiuscole e minuscole, ma gli alias che differiscono solo nella la distinzione tra maiuscole e minuscole sono molto inclini all'errore. Non è possibile modificare un nome alias. Tuttavia, puoi eliminare l'alias e creare un nuovo alias con il nome desiderato.

Puoi creare un alias con lo stesso nome in diverse regioni

Ad esempio, puoi avere un alias `finance-key` negli Stati Uniti orientali (Virginia settentrionale) e un alias `finance-key` in Europa (Francoforte). Ogni alias verrebbe associato a una chiave KMS nella rispettiva regione. Se il tuo codice fa riferimento a un nome alias come `alias/finance-key`, puoi eseguirlo in più regioni. In ogni regione utilizza una diversa chiave KMS. Per informazioni dettagliate, vedi [Scopri come utilizzare gli alias nelle tue applicazioni](#).

Puoi modificare la chiave KMS associata a un alias

È possibile utilizzare l'[UpdateAlias](#) operazione per associare un alias a una chiave KMS diversa. Ad esempio, se l'alias `finance-key` è associato alla chiave KMS

1234abcd-12ab-34cd-56ef-1234567890ab puoi aggiornarlo in modo che sia associato alla chiave KMS 0987dcba-09fe-87dc-65ba-ab0987654321.

Tuttavia, le chiavi KMS correnti e nuove devono essere dello stesso tipo (entrambe simmetriche, asimmetriche o HMAC) e devono avere lo stesso [utilizzo della chiave](#) (ENCRYPT_DECRYPT, SIGN_VERIFY o GENERATE_VERIFY_MAC). Questa restrizione impedisce errori nel codice che utilizza alias. Se è necessario associare un alias a un tipo di chiave diverso e sono stati attenuati i rischi, puoi eliminare e ricreare l'alias.

Alcune chiavi KMS non hanno alias

Quando crei una chiave KMS nella AWS KMS console, devi assegnarle un nuovo alias. Ma non è necessario un alias quando si utilizza l'[CreateKey](#) operazione per creare una chiave KMS. Inoltre, è possibile utilizzare l'[UpdateAlias](#) operazione per modificare la chiave KMS associata a un alias e l'[DeleteAlias](#) operazione per eliminare un alias. Di conseguenza, alcune chiavi KMS potrebbero avere diversi alias e altre potrebbero non averne alcuno.

AWS crea alias nel tuo account

AWS crea alias nel tuo account per. [Chiavi gestite da AWS](#) Questi alias hanno i nomi del modulo `alias/aws/<service-name>`, ad esempio `alias/aws/s3`.

Alcuni AWS alias non hanno una chiave KMS. Questi alias predefiniti sono generalmente associati a un messaggio Chiave gestita da AWS quando si inizia a utilizzare il servizio.

Utilizzare gli alias per identificare le chiavi KMS

È possibile utilizzare un [nome alias](#) o un [alias ARN](#) per identificare una chiave KMS nelle operazioni [crittografiche](#) e [DescribeKeyGetPublicKey](#) (Se l'opzione [La chiave KMS è in un Account AWS diverso](#), è necessario utilizzare il suo [ARN della chiave](#) o ARN dell'alias.) Gli alias non sono identificatori validi per le chiavi KMS in altre operazioni AWS KMS. Per informazioni sugli [identificatori di chiave](#) validi per ogni operazione AWS KMS API, consulta le descrizioni dei parametri nell'API Reference. *KeyId* AWS Key Management Service

Non puoi utilizzare un nome alias o un ARN di alias per [identificare una chiave KMS in una policy IAM](#). Per controllare l'accesso a una chiave KMS in base ai relativi alias, usa le chiavi di condizione [kms: RequestAlias](#) o [kms: ResourceAliases](#). Per informazioni dettagliate, vedi [ABAC per AWS KMS](#).

Controllo dell'accesso agli alias

Quando crei o modifichi un alias, l'operazione interessa l'alias e la relativa chiave KMS associata. Di conseguenza, i principali che gestiscono gli alias devono disporre dell'autorizzazione per chiamare l'operazione alias sull'alias e su tutte le chiavi KMS interessate. Puoi fornire queste autorizzazioni utilizzando le [policy delle chiavi](#), le [policy IAM](#) e le [concessioni](#).

Note

Presta attenzione quando concedi ai principali l'autorizzazione per gestire tag e alias. Modificando un tag o un alias puoi consentire o negare l'autorizzazione alla chiave gestita dal cliente. Per informazioni dettagliate, consulta [ABAC per AWS KMS](#) e [Usa gli alias per controllare l'accesso alle chiavi KMS](#).

Per informazioni sul controllo dell'accesso a tutte le AWS KMS operazioni, consulta. [Riferimento per le autorizzazioni](#)

Le autorizzazioni per la creazione e la gestione degli alias funzionano come descritto di seguito.

kms:CreateAlias

Per creare un alias, il principale richiede le seguenti autorizzazioni sia per l'alias che per la chiave KMS associata.

- `kms:CreateAlias` per l'alias. Fornisci questa autorizzazione in una policy IAM collegata al principale autorizzato a creare l'alias.

L'esempio di istruzione di policy seguente specifica un alias particolare in un elemento Resource. È tuttavia possibile elencare più alias ARNs o specificare uno schema di alias, ad esempio «test*». Puoi specificare il valore Resource di "*" in modo da consentire al principale di creare qualsiasi alias nell'account e nella regione. L'autorizzazione per creare un alias può anche essere inclusa in un'autorizzazione `kms:Create*` per tutte le risorse di un account e di una regione.

```
{
  "Sid": "IAMPolicyForAnAlias",
  "Effect": "Allow",
  "Action": [
    "kms:CreateAlias",
```

```

    "kms:UpdateAlias",
    "kms>DeleteAlias"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/test-key"
}

```

- `kms:CreateAlias` per la chiave KMS. Questa autorizzazione deve essere fornita in una policy delle chiavi o in una policy IAM delegata dalla policy delle chiavi.

```

{
  "Sid": "Key policy for 1234abcd-12ab-34cd-56ef-1234567890ab",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSAdminUser"},
  "Action": [
    "kms:CreateAlias",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}

```

È possibile utilizzare le chiavi di condizione per limitare le chiavi KMS che è possibile associare a un alias. Ad esempio, puoi usare la chiave [kms:KeySpec](#) condition per consentire al principale di creare alias solo su chiavi KMS asimmetriche. Per un elenco completo delle chiavi delle condizioni utilizzabili per limitare l'autorizzazione `kms:CreateAlias` sulle risorse delle chiavi KMS, vedi [AWS KMS autorizzazioni](#).

kms:ListAliases

Per elencare gli alias nell'account e nella regione, il principale deve disporre dell'autorizzazione `kms:ListAliases` in una policy IAM. Poiché questa policy non è correlata a una determinata chiave KMS o a una risorsa alias, il valore dell'elemento risorsa nella policy deve essere `"*"`.

Ad esempio, l'istruzione della policy IAM riportata di seguito consente al principale di elencare tutte le chiavi KMS e gli alias nell'account e nella regione.

JSON

```

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

```

```
"Effect": "Allow",
"Action": [
    "kms:ListKeys",
    "kms:ListAliases"
],
"Resource": "*"
}
```

kms:UpdateAlias

Per modificare la chiave KMS associata a un alias, il principale richiede tre elementi di autorizzazione: uno per l'alias, uno per la chiave KMS attuale e uno per la nuova chiave KMS.

Ad esempio, supponi di voler modificare l'alias `test-key` dalla chiave KMS con ID chiave `1234abcd-12ab-34cd-56ef-1234567890ab` alla chiave KMS con ID chiave `0987dcba-09fe-87dc-65ba-ab0987654321`. Per questo caso, includi le istruzioni di policy simile agli esempi riportati in questa sezione.

- `kms:UpdateAlias` per l'alias. Questa autorizzazione viene fornita in una policy IAM collegata al principale. La policy IAM seguente specifica un alias particolare. Ma puoi elencare più alias ARNs o specificare un modello di alias, ad esempio, `"test*"`. Puoi specificare il valore `Resource` di `"*"` in modo da consentire al principale di creare qualsiasi alias nell'account e nella regione.

```
{
  "Sid": "IAMPolicyForAnAlias",
  "Effect": "Allow",
  "Action": [
    "kms:UpdateAlias",
    "kms:ListAliases",
    "kms:ListKeys"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/test-key"
}
```

- `kms:UpdateAlias` per la chiave KMS che è attualmente associata all'alias. Questa autorizzazione deve essere fornita in una policy delle chiavi o in una policy IAM delegata dalla policy delle chiavi.

```
{
```



```

{
  "Sid": "Key policy for 1234abcd-12ab-34cd-56ef-1234567890ab",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSAdminUser"},
  "Action": [
    "kms:UpdateAlias",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}

```

- `kms:UpdateAlias` per la chiave KMS che l'operazione associa all'alias. Questa autorizzazione deve essere fornita in una policy delle chiavi o in una policy IAM delegata dalla policy delle chiavi.

```

{
  "Sid": "Key policy for 0987dcba-09fe-87dc-65ba-ab0987654321",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSAdminUser"},
  "Action": [
    "kms:UpdateAlias",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}

```

È possibile utilizzare chiavi di condizione per limitare una o entrambe le chiavi KMS in una operazione `UpdateAlias`. Ad esempio, puoi utilizzare una chiave [kms: ResourceAliases](#) condition per consentire al principale di aggiornare gli alias solo quando la chiave KMS di destinazione ha già un alias particolare. Per un elenco completo delle chiavi delle condizioni utilizzabili per limitare l'autorizzazione `kms:UpdateAlias` su una risorsa della chiave KMS, vedi [AWS KMS autorizzazioni](#).

kms:DeleteAlias

Per eliminare un alias, il principale richiede l'autorizzazione per l'alias e per la chiave KMS associata.

Come sempre, è necessario prestare attenzione quando si concedono le autorizzazioni per eliminare una risorsa. L'eliminazione di un alias non ha alcun effetto sulla chiave KMS associata. Anche se potrebbe causare errori nelle applicazioni che utilizzano l'alias, se si elimina erroneamente un alias, puoi ricrearlo.

- `kms:DeleteAlias` per l'alias. Fornisci questa autorizzazione in una policy IAM collegata al principale autorizzato a eliminare l'alias.

L'esempio di istruzione di policy seguente specifica l'alias in un elemento Resource. Ma puoi elencare più alias ARNs o specificare un modello di alias, ad esempio "test*", Puoi anche specificare un Resource valore di "*" per consentire al principale di eliminare qualsiasi alias nell'account e nella regione.

```
{
  "Sid": "IAMPolicyForAnAlias",
  "Effect": "Allow",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/test-key"
}
```

- kms:DeleteAlias per la chiave KMS associata. Questa autorizzazione deve essere fornita in una policy delle chiavi o in una policy IAM delegata dalla policy delle chiavi.

```
{
  "Sid": "Key policy for 1234abcd-12ab-34cd-56ef-1234567890ab",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/KMSAdminUser"
  },
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

Limitazione delle autorizzazioni alias

È possibile utilizzare le chiavi di condizione per limitare le autorizzazioni alias quando la risorsa è una chiave KMS. Ad esempio, la seguente policy IAM consente le operazioni alias sulle chiavi KMS in un

determinato account e nella regione. Tuttavia, utilizza la chiave [kms: KeyOrigin](#) condition per limitare ulteriormente le autorizzazioni alle chiavi KMS contenenti materiale chiave proveniente da. AWS KMS

Per un elenco completo delle chiavi di condizioni che è possibile utilizzare per limitare l'autorizzazione alias per una risorsa della chiave KMS, vedi [AWS KMS autorizzazioni](#).

```
{
  "Sid": "IAMPolicyKeyPermissions",
  "Effect": "Allow",
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms>DeleteAlias"
  ],
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "AWS_KMS"
    }
  }
}
```

Non è possibile utilizzare le chiavi di condizione in un'istruzione della policy delle chiavi in cui la risorsa è un alias. Per limitare gli alias che un principale può gestire, utilizzare il valore dell'elemento Resource dell'istruzione della policy IAM che controlla l'accesso all'alias. Ad esempio, le seguenti dichiarazioni politiche consentono al principale di creare, aggiornare o eliminare qualsiasi alias nella regione Account AWS and, a meno che l'alias non inizi con. Restricted

```
{
  "Sid": "IAMPolicyForAnAliasAllow",
  "Effect": "Allow",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms>DeleteAlias"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/*"
},
{
  "Sid": "IAMPolicyForAnAliasDeny",
  "Effect": "Deny",
  "Action": [
```

```
"kms:CreateAlias",
"kms:UpdateAlias",
"kms>DeleteAlias"
],
"Resource": "arn:aws:kms:us-west-2:111122223333:alias/Restricted*"
}
```

Creare alias

È possibile creare alias nella AWS KMS console o utilizzando le operazioni AWS KMS API.

L'alias deve essere una stringa da 1 a 256 caratteri. Può contenere solo caratteri alfanumerici, barre (/), trattini bassi (_) e trattini (-). Il nome alias per una [chiave gestita dal cliente](#) non può iniziare con `alias/aws/`. Il prefisso `alias/aws/` è riservato per [Chiave gestita da AWS](#).

È possibile creare un alias per una nuova chiave KMS o per una chiave KMS esistente. Puoi aggiungere un alias in modo che la chiave KMS venga utilizzata un particolare progetto o applicazione.

Puoi anche utilizzare un AWS CloudFormation modello per creare un alias per una chiave KMS. Per ulteriori informazioni, consulta [AWS::KMS::Alias](#) nella Guida per l'utente di AWS CloudFormation .

Utilizzo della console AWS KMS

Quando [crei una chiave KMS](#) nella AWS KMS console, devi creare un alias per la nuova chiave KMS. Per creare un alias per una chiave KMS esistente, utilizzare la scheda Alias nella pagina dei dettagli della chiave KMS.

1. [Accedi Console di gestione AWS e apri la console AWS Key Management Service \(AWS KMS\) in /kms. `https://console.aws.amazon.com`](#)
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente. Non è possibile gestire gli alias per o. Chiavi gestite da AWS Chiavi di proprietà di AWS
4. Nella tabella scegli l'alias o l'ID chiave della chiave KMS. Quindi, nella pagina dei dettagli della chiave KMS, scegliere la scheda Alias.

Se una chiave KMS dispone di più alias, la colonna Alias nella tabella mostra un alias e un riepilogo degli alias, ad esempio (+n più). Scegliendo il riepilogo alias puoi accedere direttamente alla scheda Alias nella pagina dei dettagli della chiave KMS.

5. Nella scheda Alias, scegli Crea alias. Immetti un nome alias e scegli Crea alias.

Important

Non includere informazioni riservate o sensibili in questo campo. Questo campo può essere visualizzato in testo semplice nei CloudTrail log e in altri output.

Note

Non aggiungere il prefisso alias/. La console lo aggiunge automaticamente. Se inserisci alias/ExampleAlias, il nome alias effettivo sarà alias/alias/ExampleAlias.

Utilizzo dell'API AWS KMS

Per creare un alias, usa l'[CreateAlias](#) operazione. A differenza del processo di creazione delle chiavi KMS nella console, l'[CreateKey](#) operazione non crea un alias per una nuova chiave KMS.

Important

Non includere informazioni riservate o sensibili in questo campo. Questo campo può essere visualizzato in testo semplice nei log e in CloudTrail altri output.

Puoi utilizzare l'operazione `CreateAlias` per creare un alias per una nuova chiave KMS senza alias. Puoi inoltre utilizzare l'operazione `CreateAlias` per aggiungere un alias a qualsiasi chiave KMS esistente o per ricreare un alias eliminato accidentalmente.

Nelle operazioni AWS KMS API, il nome dell'alias deve iniziare con `alias/` seguito da un nome, ad esempio. `alias/ExampleAlias` L'alias deve essere univoco nell'account e nella regione. Per trovare i nomi di alias già in uso, utilizzate l'[ListAliases](#) operazione. Il nome alias fa distinzione tra maiuscole e minuscole.

Il `TargetKeyId` può essere qualsiasi [chiave gestita dal cliente](#) nella stessa Regione AWS. Per identificare la chiave KMS utilizza [l'ID chiave](#) o [l'ARN di chiave](#). Non puoi utilizzare un altro alias.

Nell'esempio seguente viene creato l'alias `example-key` e viene associato alla chiave KMS specificata. Questi esempi utilizzano il AWS Command Line Interface (AWS CLI). Per esempi in più linguaggi di programmazione, consulta [Utilizzo CreateAlias con un AWS SDK o una CLI](#).

```
$ aws kms create-alias \
  --alias-name alias/example-key \
  --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

`CreateAlias` non restituisce alcun output. Per visualizzare il nuovo alias utilizza l'operazione `ListAliases`. Per informazioni dettagliate, vedi [Utilizzo dell'API AWS KMS](#).

Trova il nome dell'alias e l'alias ARN per una chiave KMS

Gli alias facilitano il riconoscimento delle chiavi KMS nella console. AWS KMS È possibile visualizzare gli alias per una chiave KMS nella AWS KMS console o utilizzando l'operazione [ListAliases](#). L'[DescribeKey](#) operazione, che restituisce le proprietà di una chiave KMS, non include gli alias.

Le seguenti procedure mostrano come visualizzare e identificare gli alias associati a una chiave KMS utilizzando la console e l' AWS KMS API. AWS KMS Gli esempi di AWS KMS API utilizzano il [AWS Command Line Interface \(AWS CLI\)](#), ma è possibile utilizzare qualsiasi linguaggio di programmazione supportato.

Utilizzo della AWS KMS console

La AWS KMS console visualizza gli alias associati alla chiave KMS.

1. [Apri la AWS KMS console in /kms. https://console.aws.amazon.com](#)
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Per visualizzare le chiavi nell'account creato e gestito dall'utente, nel riquadro di navigazione, seleziona Chiavi gestite dal cliente. Per visualizzare le chiavi dell'account che AWS crea e gestisce per te, nel riquadro di navigazione, scegli chiavi gestite.AWS
4. La colonna Alias mostra l'alias per ogni chiave KMS. Se una chiave KMS non dispone di un alias, nella colonna Alias compare un trattino (-).

Se una chiave KMS dispone di più alias, la colonna Alias ha anche un riepilogo degli alias, ad esempio (+n più). Ad esempio, la seguente chiave KMS ha due alias, uno dei quali è key-test.

Per trovare il nome e l'ARN alias di tutti gli alias della chiave KMS, utilizza la scheda Alias.

- Per andare direttamente alla scheda Alias, nella colonna Alias, scegli il riepilogo degli alias (+n più). Un riepilogo degli alias viene mostrato solo se la chiave KMS contiene più di un alias.
- In alternativa, scegliere l'ID chiave o alias della chiave KMS (che apre la pagina dei dettagli della chiave KMS) e scegli la scheda Alias. Le schede si trovano sotto la sezione Configurazione generale.

Customer managed keys (16)				Key actions ▼	Create key
Filter keys by aliases, key ID, or key type				< 1 2 >	⚙
☐	Aliases ▼	Key ID ▼	Status		
☐	key-test (+1 more)	1234abcd-12ab-34cd-56ef-1234567890ab	Enabled		
☐	-	1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d	Enabled		

5. La scheda Alias mostra il nome alias e l'ARN dell'alias di tutti gli alias per una chiave KMS. In questa scheda puoi anche creare ed eliminare alias per la chiave KMS.

Key policy

Cryptographic configuration

Key material

Tags

Public key

Aliases

Aliases

Info

Delete

Create new alias

Filter by Alias name

< 1 >

☐	Alias name	Alias ARN
☐	key-test	arn:aws:kms:us-east-1:111122223333:alias/key-test
☐	project-key	arn:aws:kms:us-east-1:111122223333:alias/project-key

Chiavi gestite da AWS

Puoi usare l'alias per riconoscere un Chiave gestita da AWS, come mostrato in questa Chiavi gestite da AWS pagina di esempio. Gli alias per le Chiavi gestite da AWS hanno sempre il formato: `aws/<service-name>`. Ad esempio, l'alias Chiave gestita da AWS per Amazon DynamoDB è. `aws/dynamodb`

AWS managed keys (9)	
Filter keys by alias or key ID	
Alias	
aws/dynamodb	
aws/ebs	
aws/lightsail	
aws/rds	
aws/s3	
aws/secretsmanager	
aws/ssm	
aws/workmail	
aws/xray	

Utilizzo dell'API AWS KMS

L'[ListAliases](#) operazione restituisce il nome dell'alias e l'alias ARN degli alias nell'account e nella regione. L'output include alias per Chiavi gestite da AWS e per le chiavi gestite dal cliente. Gli alias per Chiavi gestite da AWS hanno il formato `aws/<service-name>`, ad esempio `aws/dynamodb`.

La risposta potrebbe anche includere alias che non hanno alcun campo `TargetKeyId`. Si tratta di alias predefiniti che sono AWS stati creati ma non sono ancora associati a una chiave KMS.

```
$ aws kms list-aliases
{
  "Aliases": [
    {
      "AliasName": "alias/access-key",
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
      "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
      "CreationDate": 1516435200.399,
      "LastUpdatedDate": 1516435200.399
    },
    {
      "AliasName": "alias/ECC-P521-Sign",
```



```

    "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ECC-P521-Sign",
    "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": 1693622000.704,
    "LastUpdatedDate": 1693622000.704
  },
  {
    "AliasName": "alias/ImportedKey",
    "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ImportedKey",
    "TargetKeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
    "CreationDate": 1493622000.704,
    "LastUpdatedDate": 1521097200.235
  },
  {
    "AliasName": "alias/finance-project",
    "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/finance-project",
    "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "CreationDate": 1604958290.014,
    "LastUpdatedDate": 1604958290.014
  },
  {
    "AliasName": "alias/aws/dynamodb",
    "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/dynamodb",
    "TargetKeyId": "0987ab65-43cd-21ef-09ab-87654321cdef",
    "CreationDate": 1521097200.454,
    "LastUpdatedDate": 1521097200.454
  },
  {
    "AliasName": "alias/aws/ebs",
    "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/ebs",
    "TargetKeyId": "abcd1234-09fe-ef90-09fe-ab0987654321",
    "CreationDate": 1466518990.200,
    "LastUpdatedDate": 1466518990.200
  }
]
}

```

Per ottenere tutti gli alias associati a una particolare chiave KMS utilizza il parametro `KeyId` facoltativo dell'operazione `ListAliases`. Il parametro `KeyId` accetta [l'ID chiave](#) o [l'ARN della chiave](#) della chiave KMS.

In questo esempio vengono restituiti tutti gli alias associati alla chiave KMS `0987dcba-09fe-87dc-65ba-ab0987654321`.

```
$ aws kms list-aliases --key-id 0987dcba-09fe-87dc-65ba-ab0987654321
{
  "Aliases": [
    {
      "AliasName": "alias/access-key",
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
      "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
      "CreationDate": "2018-01-20T15:23:10.194000-07:00",
      "LastUpdatedDate": "2018-01-20T15:23:10.194000-07:00"
    },
    {
      "AliasName": "alias/finance-project",
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/finance-project",
      "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
      "CreationDate": 1604958290.014,
      "LastUpdatedDate": 1604958290.014
    }
  ]
}
```

Il parametro `KeyId` non accetta caratteri jolly, ma puoi utilizzare le caratteristiche del linguaggio di programmazione per filtrare la risposta.

Ad esempio, il AWS CLI comando seguente ottiene solo gli alias per. Chiavi gestite da AWS

```
$ aws kms list-aliases --query 'Aliases[?starts_with(AliasName, `alias/aws/`)]'
```

Il comando seguente ottiene solo l'alias `access-key`. Il nome `alias` fa distinzione tra maiuscole e minuscole.

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/access-key`]'
[
  {
    "AliasName": "alias/access-key",
    "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
    "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "CreationDate": "2018-01-20T15:23:10.194000-07:00",
    "LastUpdatedDate": "2018-01-20T15:23:10.194000-07:00"
  }
]
```

Aggiornare gli alias

Poiché un alias è una risorsa indipendente, puoi modificare la chiave KMS associata a un alias. Ad esempio, se l'`test-keyalias` è associato a una chiave KMS, puoi utilizzare l'[UpdateAlias](#) operazione per associarlo a una chiave KMS diversa. Questo è uno dei diversi modi per [ruotare manualmente una chiave KMS](#) senza modificare il materiale della chiave. È inoltre possibile aggiornare una chiave KMS in modo che un'applicazione che utilizzava una chiave KMS per nuove risorse utilizzi ora una diversa.

Non è possibile aggiornare un alias nella console. AWS KMS Inoltre, non puoi utilizzare `UpdateAlias` (o qualsiasi altra operazione) per modificare un nome di alias. Per modificare un nome di alias, elimina l'attuale alias e quindi crea un nuovo alias per la chiave KMS.

Quando aggiorni un alias, la chiave KMS corrente e la nuova chiave KMS devono essere dello stesso tipo (entrambe simmetriche, asimmetriche o HMAC). Devono anche avere lo stesso utilizzo della chiave (`ENCRYPT_DECRYPT` o `SIGN_VERIFY` o `GENERATE_VERIFY_MAC`). Questa restrizione impedisce errori di crittografia nel codice che utilizza alias.

L'esempio seguente inizia utilizzando l'[ListAliases](#) operazione per mostrare che l'`test-keyalias` è attualmente associato alla chiave KMS. `1234abcd-12ab-34cd-56ef-1234567890ab`

```
$ aws kms list-aliases --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
  "Aliases": [
    {
      "AliasName": "alias/test-key",
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
      "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "CreationDate": 1593622000.191,
      "LastUpdatedDate": 1593622000.191
    }
  ]
}
```

Successivamente, utilizza l'operazione `UpdateAlias` per modificare la chiave KMS associata con l'alias `test-key` alla chiave KMS `0987dcba-09fe-87dc-65ba-ab0987654321`. Non è necessario specificare la chiave KMS attualmente associata, ma solo la nuova chiave KMS ("di destinazione"). Il nome alias fa distinzione tra maiuscole e minuscole.

```
$ aws kms update-alias --alias-name 'alias/test-key' --target-key-id  
0987dcba-09fe-87dc-65ba-ab0987654321
```

Per verificare che l'alias sia ora associato alla chiave KMS di destinazione, utilizza nuovamente l'operazione `ListAliases`. Questo AWS CLI comando utilizza il `--query` parametro per ottenere solo l'`test-keyalias`. I campi `TargetKeyId` e `LastUpdatedDate` vengono aggiornati.

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/test-key`]'  
[  
  {  
    "AliasName": "alias/test-key",  
    "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",  
    "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",  
    "CreationDate": 1593622000.191,  
    "LastUpdatedDate": 1604958290.154  
  }  
]
```

Eliminare un alias

È possibile eliminare un alias nella AWS KMS console o utilizzando l'[DeleteAlias](#) operazione. Prima di eliminare un alias, assicurati che non sia in uso. Sebbene l'eliminazione di un alias non influisca sulla chiave KMS associata, potrebbe creare problemi per qualsiasi applicazione che lo utilizza. Se elimini un alias per errore, puoi creare un nuovo alias con lo stesso nome e associarlo alla stessa o a un'altra chiave KMS.

Se si elimina una chiave KMS, vengono eliminati tutti gli alias associati a tale chiave KMS.

Utilizzo della console AWS KMS

Per eliminare un alias nella AWS KMS console, utilizza la scheda `Alias` nella pagina dei dettagli della chiave KMS. È possibile eliminare più alias per una chiave KMS contemporaneamente.

1. [Accedi Console di gestione AWS e apri la console AWS Key Management Service \(AWS KMS\) in /kms. https://console.aws.amazon.com](https://console.aws.amazon.com/kms)
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli `Chiavi gestite dal cliente`. Non è possibile gestire gli alias per o. `Chiavi gestite da AWS` `Chiavi di proprietà di AWS`

4. Nella tabella scegli l'alias o l'ID chiave della chiave KMS. Quindi, nella pagina dei dettagli della chiave KMS, scegliere la scheda Alias.

Se una chiave KMS dispone di più alias, la colonna Alias nella tabella mostra un alias e un riepilogo degli alias, ad esempio (+n più). Scegliendo il riepilogo alias puoi accedere direttamente alla scheda Alias nella pagina dei dettagli della chiave KMS.

5. Sulla scheda Alias seleziona la casella di controllo accanto agli alias che si desidera eliminare. Scegli Elimina.

Utilizzo dell'API AWS KMS

Per eliminare un alias, usa l'[DeleteAlias](#) operazione. Questa operazione elimina un alias alla volta. Il nome alias fa distinzione tra maiuscole e minuscole e deve essere preceduto dal prefisso `alias/`.

Ad esempio, il comando seguente elimina l'alias `test-key`. Questo comando non restituisce alcun output.

```
$ aws kms delete-alias --alias-name alias/test-key
```

Per verificare che l'alias venga eliminato, utilizzare l'[ListAliases](#) operazione. Il comando seguente utilizza il `--query` parametro in AWS CLI per ottenere solo l'`test-key` alias. Le parentesi vuote nella risposta indicano che la risposta di `ListAliases` non include un alias `test-key`. Per eliminare le parentesi, utilizza il parametro `--output text` e il valore.

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/test-key`]'
[]
```

Usa gli alias per controllare l'accesso alle chiavi KMS

È possibile controllare l'accesso alle chiavi KMS in base agli alias associati alla chiave KMS. Per farlo, usa i tasti [kms: RequestAlias](#) e [kms: condition. ResourceAliases](#). Questa funzionalità fa parte del AWS KMS supporto per il controllo degli [accessi basati sugli attributi \(ABAC\)](#).

La chiave di condizione `kms:RequestAlias` consente o nega l'accesso a una chiave KMS in base all'alias in una richiesta. La chiave di condizione `kms:ResourceAliases` consente o nega l'accesso a una chiave KMS in base agli alias associati alla chiave KMS.

Queste funzionalità non consentono di identificare una chiave KMS utilizzando un alias nell'elemento `resource` di un'istruzione di policy. Quando un alias è il valore di un elemento `resource`, la policy si applica alla risorsa alias e non a qualsiasi chiave KMS che potrebbe essere associata.

Note

È possibile che occorranza fino a cinque minuti affinché le modifiche a tag e alias diventano effettive per l'autorizzazione delle chiavi KMS. Le modifiche recenti potrebbero essere visibili nelle operazioni API prima che influiscano sull'autorizzazione.

Quando si utilizzano alias per controllare l'accesso alle chiavi KMS, considerare quanto segue:

- Utilizzare gli alias per rafforzare la best practice dell'[accesso con privilegio minimo](#). Assegna ai principali IAM solo le autorizzazioni necessarie solo per quelle chiavi KMS che devono utilizzare o gestire. Ad esempio, utilizzare gli alias per identificare le chiavi KMS utilizzate per un progetto. Quindi concedere al team di progetto l'autorizzazione a utilizzare solo le chiavi KMS con gli alias del progetto.
- Fai attenzione a concedere ai principali le autorizzazioni `kms:CreateAlias`, `kms:UpdateAlias`, oppure `kms>DeleteAlias` che consentono di aggiungere, modificare ed eliminare alias. Quando si utilizzano gli alias per controllare l'accesso alle chiavi KMS, la modifica di un alias può concedere alle entità l'autorizzazione di utilizzare chiavi KMS che altrimenti non disponevano dell'autorizzazione per utilizzare. Può inoltre negare l'accesso alle chiavi KMS richieste da altre entità principali per svolgere il proprio lavoro.
- Controlla i tuoi responsabili Account AWS che attualmente dispongono dell'autorizzazione a gestire gli alias e modifica le autorizzazioni, se necessario. Gli amministratori delle chiavi che non dispongono dell'autorizzazione per modificare le policy delle chiavi o creare concessioni possono controllare l'accesso alle chiavi KMS se dispongono dell'autorizzazione per gestire gli alias.

Ad esempio, la console [policy delle chiavi predefinita per amministratori delle chiavi](#) include le autorizzazioni `kms:CreateAlias`, `kms>DeleteAlias`, e `kms:UpdateAlias`. Le policy IAM potrebbero concedere autorizzazioni alias per tutte le chiavi KMS nel Account AWS. Ad esempio, la politica [AWSKeyManagementServicePowerUser](#) gestita consente ai responsabili di creare, eliminare ed elencare gli alias per tutte le chiavi KMS ma non di aggiornarle.

- Prima di impostare una politica che dipenda da un alias, esamina gli alias sulle chiavi KMS del tuo Account AWS. Assicurati che la policy sia valida solo per gli alias che intendi includere. Usa [CloudTrail i registri e gli CloudWatch allarmi](#) per avvisarti delle modifiche agli alias che potrebbero

influire sull'accesso alle tue chiavi KMS. Inoltre, la [ListAliases](#)risposta include la data di creazione e la data dell'ultimo aggiornamento per ogni alias.

- Le condizioni della policy degli alias utilizzano la corrispondenza dei pattern; non sono legate a una particolare istanza di un alias. Una policy che utilizza chiavi di condizione basate su alias influisce su tutti gli alias nuovi ed esistenti che corrispondono al modello. Se si elimina e si ricrea un alias che corrisponde a una condizione di policy, la condizione si applica al nuovo alias, esattamente come quello precedente.

La chiave di condizione `kms:RequestAlias` si basa sull'alias specificato esplicitamente in una richiesta di operazione. La chiave di condizione `kms:ResourceAliases` dipende dagli alias associati a una chiave KMS, anche se non vengono visualizzati nella richiesta.

kms:RequestAlias

Consentire o negare l'accesso a una chiave KMS in base all'alias che identifichi la chiave KMS in una richiesta. Puoi utilizzare la chiave [kms: RequestAlias](#) condition in una policy chiave o in una [policy](#) IAM. Si applica alle operazioni che utilizzano un alias per identificare una chiave KMS in una richiesta, vale a dire [le operazioni crittografiche](#), e. [DescribeKeyGetPublicKey](#) Non è valido per le operazioni di alias, come o. [CreateAliasDeleteAlias](#)

Nella chiave condizione, specificare un [Nome alias](#) o modello di nome alias. Non puoi specificare un [ARN di alias](#).

Ad esempio, la seguente istruzione della policy delle chiavi consente al principale di utilizzare nella chiave KMS le operazioni specificate. L'autorizzazione diventa effettiva solo quando la richiesta utilizza un alias che include `alpha` per identificare la chiave KMS.

```
{
  "Sid": "Key policy using a request alias condition",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/alpha-developer"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*",
}
```

```
"Condition": {
  "StringLike": {
    "kms:RequestAlias": "alias/*alpha*"
  }
}
```

La seguente richiesta di esempio da un principale autorizzato soddisferebbe la condizione. Tuttavia, una richiesta che ha utilizzato un [ID chiave](#), un [ARN della chiave](#) o un alias diverso non soddisferebbe la condizione, anche se questi valori identificavano la stessa chiave KMS.

```
$ aws kms describe-key --key-id "arn:aws:kms:us-west-2:111122223333:alias/project-alpha"
```

kms:ResourceAliases

Consentire o negare l'accesso a una chiave KMS in base agli alias associati alla chiave KMS, anche se l'alias non viene utilizzato in una richiesta. La chiave [kms: ResourceAliases](#) condition consente di specificare un alias o un pattern di alias, ad esempio, in modo da poterlo utilizzare in una policy IAM per controllare l'accesso a diverse chiavi KMS nella stessa `alias/test*` regione. È valido per qualsiasi AWS KMS operazione che utilizza una chiave KMS.

Ad esempio, la seguente politica IAM consente ai principali di richiamare in due le operazioni specificate sulle chiavi KMS. Account AWS Tuttavia, l'autorizzazione si applica solo alle chiavi KMS associate agli alias che iniziano con `restricted`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AliasBasedIAMPolicy",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": [
```



```
    "arn:aws:kms:*:111122223333:key/*",
    "arn:aws:kms:*:444455556666:key/*"
  ],
  "Condition": {
    "ForAnyValue:StringLike": {
      "kms:ResourceAliases": "alias/restricted*"
    }
  }
}
```

La condizione `kms:ResourceAliases` è una condizione della risorsa, non la richiesta. Pertanto, una richiesta che non specifica l'alias può ancora soddisfare la condizione.

La seguente richiesta di esempio, che specifica un alias corrispondente, soddisfa la condizione.

```
$ aws kms enable-key-rotation --key-id "alias/restricted-project"
```

Tuttavia, la seguente richiesta di esempio soddisfa anche la condizione, a condizione che la chiave KMS specificata abbia un alias che inizia con `restricted`, anche se quell'alias non viene utilizzato nella richiesta.

```
$ aws kms enable-key-rotation --key-id "1234abcd-12ab-34cd-56ef-1234567890ab"
```

Scopri come utilizzare gli alias nelle tue applicazioni

Puoi utilizzare un alias per rappresentare una chiave KMS nel codice dell'applicazione. Il `KeyId` parametro nelle [operazioni AWS KMS crittografiche](#) e [GetPublicKey](#) accetta un nome alias o un alias ARN. [DescribeKey](#)

Ad esempio, il comando `GenerateDataKey` seguente utilizza un nome alias (`alias/finance`) per identificare una chiave KMS. Il nome alias è il valore del parametro `KeyId`.

```
$ aws kms generate-data-key --key-id alias/finance --key-spec AES_256
```

Se la chiave KMS si trova in un altro Account AWS, è necessario utilizzare una chiave ARN o un alias ARN in queste operazioni. Quando si utilizza un alias ARN, tenere presente che l'alias per una chiave

KMS è definito nell'account proprietario della chiave KMS e potrebbe differire in ogni regione. Per informazioni sulla ricerca dell'ARN alias, consulta [Trova il nome dell'alias e l'alias ARN per una chiave KMS](#).

Ad esempio, il seguente comando `GenerateDataKey` utilizza una chiave KMS che non è presente nell'account del chiamante. L'alias `ExampleAlias` è associato alla chiave KMS nell'account e nella regione specificati.

```
$ aws kms generate-data-key --key-id arn:aws:kms:us-west-2:444455556666:alias/ExampleAlias --key-spec AES_256
```

Uno degli usi più efficaci degli alias è nelle applicazioni eseguite in più Regioni AWS. Ad esempio, puoi avere un'applicazione globale che utilizza una [chiave KMS asimmetrica](#) RSA per la firma e la verifica.

- Nella regione Stati Uniti occidentali (Oregon) (us-west-2) vuoi utilizzare `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`.
- In Europa (Francoforte) (eu-central-1), vuoi utilizzare `arn:aws:kms:eu-central-1:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321`
- Nella regione Asia Pacifico (Singapore) (ap-southeast-1), vuoi utilizzare `arn:aws:kms:ap-southeast-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d`.

Puoi creare una versione diversa dell'applicazione in ogni regione o utilizzare un dizionario o un'istruzione switch per selezionare la chiave KMS corretta per ogni regione. Tuttavia è molto più semplice creare un alias con lo stesso nome alias in ogni regione. Tieni presente che il nome alias rispetta la distinzione tra maiuscole e minuscole.

```
aws --region us-west-2 kms create-alias \  
  --alias-name alias/new-app \  
  --key-id arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab  
  
aws --region eu-central-1 kms create-alias \  
  --alias-name alias/new-app \  
  --key-id arn:aws:kms:eu-central-1:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321  
  
aws --region ap-southeast-1 kms create-alias \  
  --alias-name alias/new-app \  
  --key-id arn:aws:kms:ap-southeast-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d
```

```
--key-id arn:aws:kms:ap-  
southeast-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d
```

Quindi, utilizza l'alias nel codice. Quando il codice viene eseguito in ogni regione, l'alias farà riferimento alla relativa chiave KMS associata in quella regione. Ad esempio, questo codice chiama l'operazione [Sign](#) con un nome alias.

```
aws kms sign --key-id alias/new-app \  
  --message $message \  
  --message-type RAW \  
  --signing-algorithm RSASSA_PSS_SHA_384
```

Tuttavia, esiste il rischio che l'alias possa essere eliminato o aggiornato per essere associato a un'altra chiave KMS. In tal caso, i tentativi dell'applicazione di verificare le firme utilizzando il nome alias avranno esito negativo e potrebbe essere necessario ricreare o aggiornare l'alias.

Per ridurre questo rischio, presta attenzione a concedere ai principali l'autorizzazione a gestire gli alias utilizzati nell'applicazione. Per informazioni dettagliate, vedi [Controllo dell'accesso agli alias](#).

Esistono diverse altre soluzioni per le applicazioni che crittografano i dati in più regioni AWS, tra cui [AWS Encryption SDK](#).

Trova gli alias nei log AWS CloudTrail

È possibile utilizzare un alias per rappresentare un elemento AWS KMS key in un' AWS KMS operazione API. In tal caso, l'alias e la chiave ARN della chiave KMS vengono registrati nella voce di registro AWS CloudTrail dell'evento. L'alias viene visualizzato nel campo `requestParameters`. L'ARN di chiave viene visualizzato nel campo `resources`. Questo è vero anche quando un AWS servizio ne utilizza una Chiave gestita da AWS nel tuo account.

Ad esempio, la [GenerateDataKey](#) richiesta seguente utilizza l'`project-keyalias` per rappresentare una chiave KMS.

```
$ aws kms generate-data-key --key-id alias/project-key --key-spec AES_256
```

Quando questa richiesta viene registrata nel CloudTrail registro, la voce di registro include sia l'alias che la chiave ARN della chiave KMS effettiva utilizzata.

```
{
```

```

    "eventVersion": "1.05",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "ABCDE",
      "arn": "arn:aws:iam::111122223333:role/ProjectDev",
      "accountId": "111122223333",
      "accessKeyId": "FFHIJ",
      "userName": "example-dev"
    },
    "eventTime": "2020-06-29T23:36:41Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "205.205.123.000",
    "userAgent": "aws-cli/1.18.89 Python/3.6.10
Linux/4.9.217-0.1.ac.205.84.332.metal1.x86_64 boto3/1.17.12",
    "requestParameters": {
      "keyId": "alias/project-key",
      "keySpec": "AES_256"
    },
    "responseElements": null,
    "requestID": "d93f57f5-d4c5-4bab-8139-5a1f7824a363",
    "eventID": "d63001e2-dbc6-4aae-90cb-e5370aca7125",
    "readOnly": true,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }

```

Per informazioni dettagliate sulle AWS KMS operazioni di registrazione nei CloudTrail log, vedere.

[Registrazione delle chiamate AWS KMS API con AWS CloudTrail](#)

Tag in AWS KMS

Un tag è un'etichetta di metadati opzionale che puoi assegnare (o AWS assegnare) a una risorsa. AWS Ogni tag è costituito da una chiave di tag e da un valore di tag, entrambe le stringhe fanno distinzione tra maiuscole e minuscole. Il valore di tag può essere una stringa vuota (null). Ogni tag su una risorsa deve avere una chiave di tag diversa, ma puoi aggiungere lo stesso tag a più risorse. AWS Ogni risorsa può avere fino a 50 tag creati dall'utente.

Non includere informazioni riservate o sensibili nella chiave o nel valore del tag. I tag sono accessibili a molti Servizi AWS, inclusa la fatturazione.

[In AWS KMS, puoi aggiungere tag a una chiave gestita dal cliente quando crei la chiave KMS e taggare o rimuovere tag alle chiavi KMS esistenti a meno che non siano in attesa di eliminazione.](#)

Non puoi taggare alias, archivi di chiavi personalizzati o chiavi Chiavi gestite da AWS KMS in Chiavi di proprietà di AWS altri. Account AWS I tag sono opzionali, ma possono essere molto utili.

Ad esempio, puoi aggiungere un tag "Project"="Alpha" a tutte le chiavi KMS e ai bucket di Amazon S3 utilizzati per il progetto Alpha.

```
TagKey    = "Project"
TagValue  = "Alpha"
```

Per informazioni generali sui tag, inclusi il formato e la sintassi, consulta [Tagging resources AWS](#) in. Riferimenti generali di Amazon Web Services

I tag consentono di:

- Identifica e organizza le tue risorse. AWS Molti AWS servizi supportano l'etichettatura, quindi puoi assegnare lo stesso tag a risorse di servizi diversi per indicare che le risorse sono correlate. Ad esempio, puoi assegnare lo stesso tag a una chiave KMS e a un volume o segreto Amazon Elastic Block Store (Amazon EBS). Gestione dei segreti AWS Puoi inoltre utilizzare i tag per identificare le chiavi KMS per l'automazione.
- Tieni traccia dei costi. AWS Quando aggiungi tag alle tue AWS risorse, AWS genera un rapporto sull'allocazione dei costi con utilizzo e costi aggregati per tag. È possibile utilizzare questa funzionalità per tenere traccia AWS KMS dei costi di un progetto, un'applicazione o un centro di costo.

Per ulteriori informazioni sull'utilizzo dei tag per l'allocazione dei costi, consulta [Uso dei tag per l'allocazione dei costi](#) nella Guida per l'utente di AWS Billing . Per informazioni sulle regole che si

applicano alle chiavi dei tag e ai valori dei tag, consulta [Limitazioni per i tag definiti dall'utente](#) nella Guida per l'utente di AWS Billing .

- Controlla l'accesso alle tue AWS risorse. Consentire e negare l'accesso alle chiavi KMS in base ai relativi tag fa parte del AWS KMS supporto per il controllo degli [accessi basato sugli attributi \(ABAC\)](#). Per informazioni sul controllo dell'accesso in AWS KMS keys base ai relativi tag, consulta [Usa i tag per controllare l'accesso alle chiavi KMS](#) Per informazioni più generali sull'uso dei tag per controllare l'accesso alle AWS risorse, consulta [Controlling Access to AWS Resources Using Resource Tags](#) nella IAM User Guide.

AWS KMS scrive una voce nel AWS CloudTrail registro quando si utilizzano le [ListResourceTags](#) operazioni [TagResource](#) [UntagResource](#), o.

Argomenti

- [Controllo degli accessi ai tag](#)
- [Aggiungi tag a una chiave KMS](#)
- [Modifica i tag associati a una chiave KMS](#)
- [Rimuovi i tag associati a una chiave KMS](#)
- [Visualizza i tag associati a una chiave KMS](#)
- [Usa i tag per controllare l'accesso alle chiavi KMS](#)

Controllo degli accessi ai tag

Per aggiungere, visualizzare ed eliminare i tag, nella AWS KMS console o utilizzando l'API, i responsabili necessitano delle autorizzazioni di etichettatura. Puoi fornire queste autorizzazioni nelle [policy delle chiavi](#). Puoi anche fornirli nelle policy IAM (incluse le [Policy di endpoint VPC](#)), ma solo se [la policy delle chiavi lo consente](#). La policy [AWSKeyManagementServicePowerUser](#) gestita consente ai responsabili di etichettare, rimuovere i tag ed elencare i tag su tutte le chiavi KMS a cui l'account può accedere.

Puoi anche limitare queste autorizzazioni utilizzando le chiavi di condizione AWS globali per i tag. In AWS KMS, queste condizioni possono controllare l'accesso alle operazioni di etichettatura, come e [TagResource](#). [UntagResource](#)

Note

Presta attenzione quando concedi ai principali l'autorizzazione per gestire tag e alias. Modificando un tag o un alias puoi consentire o negare l'autorizzazione alla chiave gestita dal cliente. Per informazioni dettagliate, consulta [ABAC per AWS KMS](#) e [Usa i tag per controllare l'accesso alle chiavi KMS](#).

Per esempi di policy e ulteriori informazioni, consulta [Controllo dell'accesso in base alle chiavi di tag](#) nella Guida per l'utente di IAM.

Le autorizzazioni per creare e gestire i tag funzionano come descritto di seguito.

kms:TagResource

Consente ai principali di aggiungere o modificare tag. Per aggiungere tag durante la creazione di una chiave KMS, il principale deve disporre dell'autorizzazione in una policy IAM che non sia limitato a particolari chiavi KMS.

kms:ListResourceTags

Consente ai principali di visualizzare i tag sulle chiavi KMS.

kms:UntagResource

Consente ai principali di eliminare i tag dalle chiavi KMS.

Autorizzazioni ad assegnare tag nelle policy

Puoi fornire l'autorizzazione ad assegnare tag in una policy delle chiavi o in una policy IAM. Ad esempio, la policy delle chiavi di esempio riportata di seguito fornisce agli utenti selezionati l'autorizzazione di tag nella chiave KMS. Fornisce a tutti gli utenti che possono assumere l'esempio dei ruoli di Amministratore o Sviluppatore il permesso di visualizzare i tag.

JSON

```
{
  "Version": "2012-10-17",
  "Id": "example-key-policy",
  "Statement": [
```

```

{
  "Sid": "EnableIAMUserPermissions",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
  "Action": "kms:*",
  "Resource": "*"
},
{
  "Sid": "AllowAllTaggingPermissions",
  "Effect": "Allow",
  "Principal": {"AWS": [
    "arn:aws:iam::111122223333:user/LeadAdmin",
    "arn:aws:iam::111122223333:user/SupportLead"
  ]},
  "Action": [
    "kms:TagResource",
    "kms:ListResourceTags",
    "kms:UntagResource"
  ],
  "Resource": "*"
},
{
  "Sid": "AllowRolesViewTags",
  "Effect": "Allow",
  "Principal": {"AWS": [
    "arn:aws:iam::111122223333:role/Administrator",
    "arn:aws:iam::111122223333:role/Developer"
  ]},
  "Action": "kms:ListResourceTags",
  "Resource": "*"
}
]
}

```

Per concedere ai principali l'autorizzazione ad assegnare tag su più chiavi KMS, è possibile utilizzare una policy IAM. Affinché questa policy sia efficace, la policy delle chiavi per ogni chiave KMS deve consentire all'account di utilizzare le policy IAM per controllare l'accesso alla chiave KMS.

Ad esempio, la policy IAM seguente consente ai principali di creare chiavi KMS. Consente inoltre di creare e gestire i tag su tutte le chiavi KMS nell'account specificato. Questa combinazione consente ai responsabili di utilizzare il parametro [Tags](#) dell'[CreateKey](#) operazione per aggiungere tag a una chiave KMS durante la creazione.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyCreateKeys",
      "Effect": "Allow",
      "Action": "kms:CreateKey",
      "Resource": "*"
    },
    {
      "Sid": "IAMPolicyTags",
      "Effect": "Allow",
      "Action": [
        "kms:TagResource",
        "kms:UntagResource",
        "kms:ListResourceTags"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    }
  ]
}
```

Limitazione delle autorizzazioni ad assegnare tag

È possibile limitare le autorizzazioni di assegnazione dei tag utilizzando [Condizioni della policy](#). Le seguenti condizioni della policy possono essere applicate alle autorizzazioni `kms:TagResource` e `kms:UntagResource`. Ad esempio, è possibile utilizzare la condizione `aws:RequestTag/tag-key` per consentire a un principale di aggiungere solo tag specifici o impedire a un principale di aggiungere tag con chiavi tag particolari. In alternativa, è possibile utilizzare la condizione `kms:KeyOrigin` per impedire ai principali di assegnare o rimuovere tag dalle chiavi KMS con [materiale della chiave importato](#).

- [Leggi: RequestTag](#)
- [aws:ResourceTag/tag-key](#) (solo politiche IAM)
- [aws: TagKeys](#)
- [km: CallerAccount](#)

- [km: KeySpec](#)
- [km: KeyUsage](#)
- [km: KeyOrigin](#)
- [km: ViaService](#)

Come best practice nell'utilizzo dei tag per controllare l'accesso alle chiavi KMS, è consigliabile utilizzare la chiave di condizione `aws:RequestTag/tag-key` o `aws:TagKeys` per determinare quali tag (o chiavi tag) sono consentiti.

Ad esempio, la seguente istruzione della policy IAM è simile a quella precedente. Tuttavia, questa policy consente ai principali di creare tag (TagResource) ed eliminare i tag UntagResource solo per i tag con chiave di tag Project.

Poiché TagResource le UntagResource richieste possono includere più tag, è necessario specificare un operatore ForAllValues or ForAnyValue set con la TagKeys condizione [aws:](#). L'operatore ForAnyValue richiede che almeno una delle chiavi di tag nella richiesta corrisponda a una delle chiavi di tag nella policy. L'operatore ForAllValues richiede che tutte le chiavi di tag nella richiesta corrispondano a una delle chiavi di tag nella policy. L'ForAllValues operatore restituisce anche true se non ci sono tag nella richiesta, ma TagResource UntagResource fallisce quando non viene specificato alcun tag. Per dettagli sugli operatori del set, consulta [Utilizzare più chiavi e valori](#) nella Guida per l'utente di IAM.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyCreateKey",
      "Effect": "Allow",
      "Action": "kms:CreateKey",
      "Resource": "*"
    },
    {
      "Sid": "IAMPolicyViewAllTags",
      "Effect": "Allow",
      "Action": "kms:ListResourceTags",
      "Resource": "arn:aws:kms:*:111122223333:key/*"
```

```
    },
    {
      "Sid": "IAMPolicyManageTags",
      "Effect": "Allow",
      "Action": [
        "kms:TagResource",
        "kms:UntagResource"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "ForAllValues:StringEquals": {"aws:TagKeys": "Project"}
      }
    }
  ]
}
```

Aggiungi tag a una chiave KMS

I tag aiutano a identificare e organizzare AWS le tue risorse. Puoi aggiungere tag a una chiave gestita dal cliente quando [crei la chiave KMS](#) o aggiungere tag a chiavi KMS esistenti. Non puoi taggare.

Chiavi gestite da AWS

Le seguenti procedure mostrano come aggiungere tag alle chiavi gestite dal cliente utilizzando la AWS KMS console e l' AWS KMS API. Gli esempi di AWS KMS API utilizzano il [AWS Command Line Interface \(AWS CLI\)](#), ma è possibile utilizzare qualsiasi linguaggio di programmazione supportato.

Argomenti

- [Aggiungere tag durante la creazione di una chiave KMS](#)
- [Aggiungi tag alle chiavi KMS esistenti](#)

Aggiungere tag durante la creazione di una chiave KMS

Puoi aggiungere tag a una chiave KMS mentre crei la chiave utilizzando la AWS KMS console o l'[CreateKey](#) operazione. Per aggiungere tag durante la creazione di una chiave KMS, è necessario disporre `kms:TagResource` dell'autorizzazione in una politica IAM oltre alle autorizzazioni necessarie per creare chiavi KMS. Come minimo, l'autorizzazione deve coprire tutte le chiavi KMS nell'account e nella regione. Per informazioni dettagliate, vedi [Controllo degli accessi ai tag](#).

Utilizzo della console AWS KMS

Per aggiungere tag quando si crea una chiave KMS nella console, è necessario disporre delle autorizzazioni necessarie per visualizzare le chiavi KMS nella console oltre alle autorizzazioni necessarie per taggare e creare chiavi KMS. Come minimo, l'autorizzazione deve coprire tutte le chiavi KMS nell'account e nella regione.

1. [Accedi Console di gestione AWS e apri la console AWS Key Management Service \(\)AWS KMS su /kms. https://console.aws.amazon.com](https://console.aws.amazon.com/kms)
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente. (Non è possibile gestire i tag di una Chiave gestita da AWS)
4. Scegli il tipo di chiave, quindi scegli Next (Successivo).
5. Immetti un alias e una descrizione opzionale.
6. Inserisci una chiave di tag e un valore di tag opzionale. Per aggiungere altri tag, scegli Aggiungi nuovo tag. Per rimuovere un tag, scegli Remove (Rimuovi). Al termine dell'applicazione dei tag alla nuova chiave KMS, scegli Successivo.
7. Completa la creazione della chiave KMS.

Utilizzando l'API AWS KMS

Per specificare i tag durante la creazione di chiavi utilizzando l'[CreateKey](#) operazione, utilizzate il Tags parametro dell'operazione.

Il valore del parametro Tags di CreateKey è una raccolta di coppie di chiave di tag e valore di tag per cui si applica la distinzione tra maiuscole e minuscole. Ogni tag in una chiave KMS deve avere un nome di tag diverso. Il valore di tag può essere una stringa nulla o vuota.

Ad esempio, il AWS CLI comando seguente crea una chiave KMS di crittografia simmetrica con un tag. Project:Alpha Quando si specificano più coppie chiave-valore, utilizzare uno spazio per separare ciascuna coppia.

```
$ aws kms create-key --tags TagKey=Project,TagValue=Alpha
```

Quando questo comando ha esito positivo, restituisce un oggetto `KeyMetadata` con informazioni relative alla nuova chiave KMS. Tuttavia, `KeyMetadata` non include tag. Per ottenere i tag, usa l'operazione. [ListResourceTags](#)

Aggiungi tag alle chiavi KMS esistenti

Puoi aggiungere tag alle chiavi KMS esistenti gestite dai clienti nella AWS KMS console o utilizzando l'[TagResource](#) operazione. Per aggiungere tag, è necessaria l'autorizzazione all'aggiunta di tag sulla chiave KMS. È possibile ottenere questa autorizzazione dalla policy delle chiavi per la chiave KMS o, se la policy delle chiavi lo consente, da una policy IAM che include la chiave KMS.

Utilizzo della console AWS KMS

1. Accedi Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente. (Non è possibile gestire i tag di una Chiave gestita da AWS)
4. È possibile utilizzare il filtro tabella per visualizzare solo le chiavi KMS con tag specifici. Per i dettagli, consulta [Visualizzare i tag utilizzando](#) la console. AWS KMS
5. Seleziona la casella di controllo accanto all'alias di una chiave KMS.
6. Scegliere Key actions (Operazioni sulle chiavi), Add or edit tags (Aggiungi o modifica tag).
7. Nella pagina dei dettagli della chiave KMS scegli la scheda Tag.
 - Per creare il primo tag, scegli Crea tag, digita una chiave di tag (obbligatorio) e il valore di tag (opzionale), quindi scegli Salva.

Se lasci vuoto il valore del tag, il valore effettivo del tag è una stringa nulla o vuota.
 - Per aggiungere un tag, scegli Modifica, scegli Aggiungi tag, digita una chiave di tag e il valore di tag, quindi scegli Salva.
8. Per salvare le modifiche, scegliere Save changes (Salva modifiche).

Utilizzo dell' AWS KMS API

L'[TagResource](#) operazione aggiunge uno o più tag a una chiave KMS. Non è possibile utilizzare questa operazione per aggiungere tag in un'altra Account AWS. È inoltre possibile utilizzare l'

TagResource operazione per modificare i tag esistenti. Per ulteriori informazioni, consulta [the section called “Modifica i tag”](#).

Per aggiungere un tag, specifica una nuova chiave di tag e un valore di tag. Ogni tag in una chiave KMS deve avere un nome di tag diverso. Il valore di tag può essere una stringa nulla o vuota.

Ad esempio, il comando seguente aggiunge i tag **Purpose** e **Department** a una chiave KMS di esempio.

```
$ aws kms tag-resource \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --tags TagKey=Purpose,TagValue=Pretest TagKey=Department,TagValue=Finance
```

Quando questo comando ha esito positivo, non restituisce alcun output. Per visualizzare i tag su una chiave KMS, usa l'[ListResourceTags](#) operazione.

Modifica i tag associati a una chiave KMS

I tag aiutano a identificare e organizzare le AWS risorse. Puoi modificare i tag associati alle chiavi KMS gestite dai clienti nella AWS KMS console o utilizzando l'[TagResource](#) operazione. Non è possibile modificare i tag di un Chiave gestita da AWS.

Le seguenti procedure mostrano come modificare i tag associati a una chiave KMS. Gli esempi di AWS KMS API utilizzano il [AWS Command Line Interface \(AWS CLI\)](#), ma è possibile utilizzare qualsiasi linguaggio di programmazione supportato.

Utilizzo della AWS KMS console

1. Accedi Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente. (Non è possibile modificare i tag di un) Chiave gestita da AWS
4. È possibile utilizzare il filtro tabella per visualizzare solo le chiavi KMS con tag specifici. Per i dettagli, consulta [Visualizzare i tag utilizzando la AWS KMS console](#).
5. Seleziona la casella di controllo accanto all'alias di una chiave KMS.

6. Scegliere Key actions (Operazioni sulle chiavi), Add or edit tags (Aggiungi o modifica tag).
7. Nella pagina dei dettagli della chiave KMS scegli la scheda Tag.
 - Per modificare il nome o il valore di un tag, scegliere Edit (Modifica), apportare le modifiche, quindi scegliere Save (Salva).
8. Per salvare le modifiche, scegliere Save changes (Salva modifiche).

Utilizzo dell' AWS KMS API

L'[TagResource](#) operazione aggiunge uno o più tag a una chiave gestita dal cliente;. Tuttavia, puoi anche utilizzare [TagResource](#) per modificare il valore del tag di un tag esistente. Non puoi utilizzare questa operazione per aggiungere o modificare tag in un Account AWS diverso.

Per modificare un tag, specifica una chiave di tag esistente e un nuovo valore di tag. Ogni tag in una chiave KMS deve avere un nome di tag diverso. Il valore di tag può essere una stringa nulla o vuota.

Ad esempio, questo comando modifica il valore del tag Purpose da Pretest a Test.

```
$ aws kms tag-resource \  
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
    --tags TagKey=Purpose,TagValue=Test
```

Rimuovi i tag associati a una chiave KMS

I tag aiutano a identificare e organizzare le AWS risorse. Puoi rimuovere i tag associati alle chiavi KMS gestite dai clienti nella AWS KMS console o utilizzando l'[UntagResource](#) operazione. Non è possibile modificare o rimuovere i tag di un Chiave gestita da AWS.

Le seguenti procedure mostrano come rimuovere i tag da una chiave KMS. Gli esempi di AWS KMS API utilizzano [AWS Command Line Interface \(AWS CLI\)](#), ma è possibile utilizzare qualsiasi linguaggio di programmazione supportato.

Utilizzo della AWS KMS console

1. Accedi Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.

3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente. (Non è possibile gestire i tag di una Chiave gestita da AWS)
4. È possibile utilizzare il filtro tabella per visualizzare solo le chiavi KMS con tag specifici. Per i dettagli, consulta [Visualizzare i tag utilizzando](#) la console. AWS KMS
5. Seleziona la casella di controllo accanto all'alias di una chiave KMS.
6. Scegliere Key actions (Operazioni sulle chiavi), Add or edit tags (Aggiungi o modifica tag).
7. Nella pagina dei dettagli della chiave KMS scegli la scheda Tag.
 - Per eliminare un tag, scegliere Edit (Modifica). Nell riga del tag, scegliere Remove (Rimuovi), quindi Save (Salva).
8. Per salvare le modifiche, scegliere Save changes (Salva modifiche).

Utilizzo dell' AWS KMS API

L'[UntagResource](#) operazione elimina i tag da una chiave KMS. Per identificare i tag da eliminare, specifica le chiavi dei tag. Non puoi utilizzare questa operazione per eliminare tag dalle chiavi KMS in un Account AWS diverso.

Quando l'operazione `UntagResource` ha esito positivo non restituisce alcun output. Inoltre, se la chiave di tag specificata non viene trovata nella chiave KMS, non viene generata un'eccezione né restituita una risposta. Per confermare che l'operazione ha funzionato, usa l'[ListResourceTags](#) operazione.

Ad esempio, questo comando elimina il tag **Purpose** e il relativo valore dalla chiave KMS specificata.

```
$ aws kms untag-resource --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --tag-keys Purpose
```

Visualizza i tag associati a una chiave KMS

I tag aiutano a identificare e organizzare AWS le tue risorse. Puoi visualizzare i tag associati alle chiavi KMS gestite dai clienti nella AWS KMS console o utilizzando l'[ListResourceTags](#) operazione.

Le seguenti procedure mostrano come trovare i tag associati a una chiave KMS specifica. Gli esempi di AWS KMS API utilizzano il [AWS Command Line Interface \(AWS CLI\)](#), ma è possibile utilizzare qualsiasi linguaggio di programmazione supportato.

Utilizzo della AWS KMS console

1. Accedi Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente. (Non è possibile gestire i tag di una Chiave gestita da AWS)
4. È possibile utilizzare il filtro tabella per visualizzare solo le chiavi KMS con tag specifici.

Per visualizzare solo le chiavi KMS con un determinato tag, scegli la casella filtro, scegli la chiave di tag, quindi scegli tra i valori effettivi dei tag. È anche possibile digitare o tutto il valore del tag o solo parte di esso.

Nella tabella risultante vengono visualizzate tutte le chiavi KMS con il tag scelto. Tuttavia, il tag non viene visualizzato. Per visualizzare il tag, scegli l'alias o l'ID chiave della chiave KMS e nella pagina dei dettagli scegli la scheda Tag. Le schede appaiono nella sezione Configurazione generale.

Per questo filtro sono necessari sia la chiave di tag che il valore del tag. Non troverà le chiavi KMS digitando solo la chiave di tag o solo il suo valore. Per filtrare i tag in base alla chiave o al valore del tag, utilizza l'[ListResourceTags](#) operazione per contrassegnare le chiavi KMS, quindi utilizza le funzionalità di filtro del linguaggio di programmazione in uso.

5. Seleziona la casella di controllo accanto all'alias di una chiave KMS.
6. Scegliere Key actions (Operazioni sulle chiavi), Add or edit tags (Aggiungi o modifica tag).
7. Nella pagina dei dettagli della chiave KMS scegli la scheda Tag.

Utilizzo dell'API AWS KMS

L'[ListResourceTags](#) operazione ottiene i tag per una chiave KMS. Il parametro KeyId è obbligatorio. Non puoi utilizzare questa operazione per visualizzare i tag nelle chiavi KMS in un Account AWS diverso.

Ad esempio, il comando seguente restituisce i tag per una chiave KMS di esempio.

```
$ aws kms list-resource-tags --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

```
"Truncated": false,
"Tags": [
  {
    "TagKey": "Project",
    "TagValue": "Alpha"
  },
  {
    "TagKey": "Purpose",
    "TagValue": "Test"
  },
  {
    "TagKey": "Department",
    "TagValue": "Finance"
  }
]
```

Usa i tag per controllare l'accesso alle chiavi KMS

Puoi controllare l'accesso a AWS KMS keys in base ai tag sulla chiave KMS. Ad esempio, è possibile scrivere una policy IAM che consenta ai principali di abilitare e disabilitare solo le chiavi KMS con un tag specifico. In alternativa, è possibile utilizzare una policy IAM per impedire ai principali di utilizzare le chiavi KMS nelle operazioni di crittografia a meno che la chiave KMS non disponga di un tag specifico.

Questa funzionalità fa parte del AWS KMS supporto per il [controllo degli accessi basato sugli attributi \(ABAC\)](#). Per informazioni sull'utilizzo dei tag per controllare l'accesso alle AWS risorse, consulta [A cosa serve ABAC? AWS](#) e [Controllo dell'accesso alle AWS risorse utilizzando i tag delle risorse](#) nella Guida per l'utente IAM. Per informazioni sulla risoluzione dei problemi di accesso relativi a ABAC, vedi [Risoluzione dei problemi ABAC per AWS KMS](#).

Note

Potrebbero essere necessari fino a cinque minuti per le modifiche di tag e alias per influenzare l'autorizzazione delle chiavi KMS. Le modifiche recenti potrebbero essere visibili nelle operazioni API prima che influiscano sull'autorizzazione.

AWS KMS supporta la chiave di [contesto global condition aws:ResourceTag/tag-key](#), che consente di controllare l'accesso alle chiavi KMS in base ai tag sulla chiave KMS. Poiché più chiavi KMS possono

avere lo stesso tag, questa caratteristica consente di applicare l'autorizzazione a un set selezionato di chiavi KMS. È inoltre possibile modificare facilmente le chiavi KMS nel set modificandone i tag.

In AWS KMS, la chiave `aws:ResourceTag/tag-key` condition è supportata solo nelle politiche IAM. Non è supportata nelle politiche chiave, che si applicano solo a una chiave KMS, o nelle operazioni che non utilizzano una chiave KMS particolare, come le operazioni [ListKeys](#) o [ListAliases](#).

Il controllo dell'accesso con i tag offre un modo semplice, scalabile e flessibile per gestire le autorizzazioni. Tuttavia, se non è stata progettata e gestita correttamente, può consentire o negare l'accesso alle chiavi KMS inavvertitamente. Se utilizzi tag per controllare l'accesso, prendi in considerazione le seguenti procedure.

- Utilizza i tag per rafforzare le best practice di [Accesso meno privilegiato](#). Assegna ai principali IAM solo le autorizzazioni necessarie solo per quelle chiavi KMS che devono utilizzare o gestire. Ad esempio, utilizzare i tag per etichettare le chiavi KMS utilizzate per un progetto. Quindi concedere al team del progetto l'autorizzazione a utilizzare solo le chiavi KMS con il tag progetto.
- Fai attenzione a dare ai principali le autorizzazioni `kms:TagResource` e `kms:UntagResource` che consentono di aggiungere, modificare ed eliminare tag. Quando si utilizzano i tag per controllare l'accesso alle chiavi KMS, la modifica di un tag può concedere ai principali l'autorizzazione di utilizzare chiavi KMS che altrimenti non disponevano dell'autorizzazione. Può inoltre negare l'accesso alle chiavi KMS richieste da altre entità per svolgere il proprio lavoro. Gli amministratori delle chiavi che non dispongono dell'autorizzazione per modificare le policy delle chiavi o creare concessioni possono controllare l'accesso alle chiavi KMS se dispongono dell'autorizzazione per gestire i tag.

Quando possibile, utilizzare una condizione della policy, ad esempio `aws:RequestTag/tag-key` o `aws:TagKeys` per [limitare le autorizzazioni di un principale ad assegnare tag](#) a particolari tag o modelli di tag su specifiche chiavi KMS.

- Controlla i tuoi principi Account AWS che attualmente dispongono delle autorizzazioni di etichettatura e rimozione dei tag e modificali, se necessario. Ad esempio, la console [policy delle chiavi predefinita](#) include autorizzazioni `kms:TagResource` e `kms:UntagResource` su quella chiave KMS. Le policy IAM possono consentire autorizzazioni ad assegnare e rimuovere tag per tutte le chiavi KMS. Ad esempio, la policy [AWSKeyManagementServicePowerUser](#) gestita consente ai responsabili di etichettare, rimuovere tag ed elencare i tag su tutte le chiavi KMS.
- Prima di impostare una politica che dipenda da un tag, controlla i tag sulle chiavi KMS del tuo Account AWS. Assicurati che la tua policy si applichi solo ai tag che intendi includere. Usa

[CloudTrail i registri e gli CloudWatch allarmi](#) per avvisarti delle modifiche ai tag che potrebbero influire sull'accesso alle tue chiavi KMS.

- Le condizioni delle policy basate su tag utilizzano la corrispondenza dei modelli; non sono legate a una particolare istanza di un tag. Una policy che utilizza chiavi di condizione basate su tag influisce su tutti i tag nuovi ed esistenti che corrispondono al modello. Se si elimina e si ricrea un tag che corrisponde a una condizione della policy, la condizione si applica al nuovo tag, proprio come quello precedente.

Ad esempio, considerare il seguente esempio di policy IAM. Consente ai responsabili di richiamare [GenerateDataKeyWithoutPlaintext](#) e [decryptare](#) le operazioni solo sulle chiavi KMS del tuo account che si trovano nella regione Asia Pacifico (Singapore) e dispongono di un tag. "Project"="Alpha" È possibile collegare questa policy ai ruoli nel progetto Alpha di esempio.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyWithResourceTag",
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:ap-southeast-1:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "Alpha"
        }
      }
    }
  ]
}
```

La seguente policy IAM di esempio consente al principale di utilizzare qualsiasi chiave KMS nell'account per le operazioni di crittografia. Ma vieta ai principali di utilizzare queste operazioni crittografiche sulle chiavi KMS con un tag "Type"="Reserved" o senza tag "Type".

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMAllowCryptographicOperations",
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    },
    {
      "Sid": "IAMDenyOnTag",
      "Effect": "Deny",
      "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Type": "Reserved"
        }
      }
    },
    {
      "Sid": "IAMDenyNoTag",
      "Effect": "Deny",
      "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
```

```
    "Null": {  
      "aws:ResourceTag/Type": "true"  
    }  
  }  
]  
}
```

Negozi chiave

Un archivio di chiavi è un luogo sicuro per l'archiviazione e l'utilizzo di chiavi crittografiche. L'archivio chiavi predefinito in supporta AWS KMS anche metodi per la generazione e la gestione delle chiavi che memorizza. Per impostazione predefinita, il materiale delle chiavi crittografiche utilizzato per la AWS KMS keys creazione AWS KMS viene generato e protetto da moduli di sicurezza hardware (HSMs) che sono il programma di convalida dei moduli [crittografici FIPS 140-3](#). Il materiale chiave per le tue chiavi KMS non esce mai non crittografato. HSMs

AWS KMS supporta diversi tipi di archivi di chiavi per proteggere il materiale chiave utilizzato AWS KMS per creare e gestire le chiavi di crittografia. Tutte le opzioni di archiviazione delle chiavi fornite da AWS KMS sono continuamente convalidate secondo FIPS 140-3 al livello di sicurezza 3 e sono progettate per impedire a chiunque, compresi AWS gli operatori, di accedere alle chiavi in chiaro o di utilizzarle senza l'autorizzazione dell'utente.

AWS KMS archivio chiavi standard

Per impostazione predefinita, una chiave KMS viene creata utilizzando lo standard AWS KMS HSM. Questo tipo di HSM può essere considerato come una flotta multi-tenant HSMs che consente l'archiviazione di chiavi più scalabile, più economica e più semplice da gestire dal punto di vista dell'utente. Se state creando una chiave KMS da utilizzare all'interno di una o più chiavi in Servizi AWS modo che il servizio possa crittografare i dati per vostro conto, creerete una chiave simmetrica. Se utilizzi una chiave KMS per la progettazione della tua applicazione, puoi scegliere di creare una chiave di crittografia simmetrica, una chiave asimmetrica o una chiave HMAC.

Nell'opzione di archiviazione delle chiavi standard, AWS KMS crea la chiave, quindi la crittografa utilizzando chiavi gestite internamente dal servizio. Più copie delle versioni crittografate delle chiavi vengono quindi archiviate in sistemi progettati per durare a lungo. La generazione e la protezione del materiale chiave nel tipo di archivio chiavi standard consente di sfruttare appieno la scalabilità, la disponibilità e la durata AWS KMS con il minor onere operativo e il costo degli archivi di AWS chiavi.

AWS KMS archivio chiavi standard con materiale chiave importato

Invece di richiedere AWS KMS di generare e archiviare le uniche copie di una determinata chiave, puoi scegliere di importare il materiale chiave in cui generare la tua chiave di crittografia simmetrica a 256 bit AWS KMS, la chiave RSA o a curva ellittica (ECC) o la chiave HMAC (Hash-Based Message

Authentication Code) e applicarla a un identificatore di chiave KMS (KeyID). Questa operazione viene talvolta denominata Bring your own key (BYOK). Il materiale chiave importato dal sistema di gestione delle chiavi locale deve essere protetto utilizzando una chiave pubblica emessa da AWS KMS, un algoritmo di wrapping crittografico supportato e un token di importazione basato sul tempo fornito da AWS KMS. Questo processo verifica che la chiave crittografata importata possa essere decrittografata da un AWS KMS HSM solo dopo aver lasciato l'ambiente.

Il materiale chiave importato può essere utile se avete requisiti specifici relativi al sistema che genera le chiavi o se desiderate una copia della chiave non utilizzata come backup. AWS Tieni presente che sei responsabile della disponibilità e della durabilità complessive di un materiale chiave importato. Sebbene AWS KMS disponga di una copia della chiave importata e rimanga sempre disponibile finché ne avrai bisogno, le chiavi importate offrono un'API speciale per l'eliminazione: `DeleteImportedKeyMaterial`. Questa API eliminerà immediatamente tutte le copie del materiale chiave importato in AWS KMS suo possesso, senza alcuna possibilità AWS di recupero della chiave. Inoltre, puoi impostare una data di scadenza per una chiave importata, dopo la quale la chiave sarà inutilizzabile. Per rendere nuovamente utile la chiave AWS KMS, sarà necessario reimportare il materiale chiave e assegnarlo allo stesso KeyID. Questa azione di eliminazione delle chiavi importate è diversa dalle chiavi standard che vengono AWS KMS generate e archiviate per conto dell'utente. Nel caso standard, il processo di eliminazione delle chiavi prevede un periodo di attesa obbligatorio durante il quale viene inizialmente bloccato l'utilizzo di una chiave programmata per l'eliminazione. Questa azione consente di visualizzare gli errori di accesso negato nei registri di qualsiasi applicazione o AWS servizio che potrebbe aver bisogno di quella chiave per accedere ai dati. Se visualizzi tali richieste di accesso, puoi scegliere di annullare l'eliminazione pianificata e riattivare la chiave. Dopo un periodo di attesa configurabile (tra 7 e 30 giorni), solo allora KMS eliminerà effettivamente il materiale chiave, il KeyID e tutti i metadati associati alla chiave. Per ulteriori informazioni sulla disponibilità e la durabilità, consulta la sezione [Protezione del materiale chiave importato](#) nella Guida per gli sviluppatori.AWS KMS

Esistono alcune limitazioni aggiuntive relative al materiale chiave importato di cui tenere conto. Poiché AWS KMS non è possibile generare nuovo materiale chiave, non è possibile configurare la rotazione automatica delle chiavi importate. Dovrai creare una nuova chiave KMS con un nuovo KeyID, quindi importare nuovo materiale chiave per ottenere una rotazione efficace. Inoltre, i testi cifrati creati AWS KMS con una chiave simmetrica importata non possono essere facilmente decrittografati utilizzando la copia locale della chiave esterna a. AWS Questo perché il formato di crittografia autenticato utilizzato da AWS KMS aggiunge metadati aggiuntivi al testo cifrato per garantire, durante l'operazione di decrittografia, che il testo cifrato sia stato creato dalla chiave KMS prevista nell'ambito di una precedente operazione di crittografia. La maggior parte dei sistemi

crittografici esterni non è in grado di analizzare questi metadati per accedere al testo cifrato non elaborato e utilizzare la propria copia di una chiave simmetrica. I testi cifrati creati con chiavi asimmetriche importate (ad esempio RSA o ECC) possono essere utilizzati al di fuori della parte corrispondente (pubblica o privata) della AWS KMS chiave perché non vi sono metadati aggiuntivi aggiunti al testo cifrato. AWS KMS

AWS KMS archivi di chiavi personalizzati

Tuttavia, se è necessario un controllo ancora maggiore di HSMs, è possibile creare un archivio chiavi personalizzato.

Un archivio chiavi personalizzato è un archivio chiavi interno AWS KMS supportato da un gestore di chiavi esterno AWS KMS, di cui l'utente è proprietario e responsabile. Gli archivi di chiavi personalizzati combinano la comoda e completa interfaccia di gestione delle chiavi AWS KMS con la capacità di possedere e controllare il materiale chiave e le operazioni crittografiche. Quando utilizzi una chiave KMS in un archivio delle chiavi personalizzate, le operazioni di crittografia vengono eseguite dal gestore delle chiavi tramite le chiavi crittografiche. Di conseguenza, l'utente si assume maggiori responsabilità per la disponibilità e la durabilità delle chiavi crittografiche e per il funzionamento di HSMs

Possederne uno HSMs può essere utile per soddisfare determinati requisiti normativi che non consentono ancora ai servizi web multi-tenant, come l'archivio di chiavi KMS standard, di conservare le chiavi crittografiche. Gli archivi di chiavi personalizzati non sono più sicuri degli archivi di chiavi KMS che utilizzano AWS-managed HSMs, ma hanno implicazioni di gestione e costi diverse (e maggiori). Di conseguenza, l'utente si assume maggiori responsabilità per la disponibilità e la durabilità delle chiavi crittografiche e per il funzionamento di HSMs. Indipendentemente dal fatto che si utilizzi l'archivio chiavi standard con AWS KMS HSMs o un archivio chiavi personalizzato, il servizio è progettato in modo che nessuno, compresi AWS i dipendenti, possa recuperare le chiavi in testo normale o utilizzarle senza la tua autorizzazione. AWS KMS supporta due tipi di archivi chiavi personalizzati, archivi AWS CloudHSM chiavi e archivi chiavi esterni.

Caratteristiche non supportate

AWS KMS non supporta le seguenti funzionalità negli archivi di chiavi personalizzati.

- [Chiavi KMS asimmetriche](#)
- [Chiavi KMS HMAC](#)
- [Chiavi KMS con materiale della chiave importato](#)

- [Rotazione automatica delle chiavi](#)
- [Chiavi multi-regione](#)

AWS CloudHSM archivio chiavi

Puoi creare una chiave KMS in un [AWS CloudHSM](#) key store, dove le chiavi utente root vengono generate, archiviate e utilizzate in un AWS CloudHSM cluster di tua proprietà e gestione. Le richieste AWS KMS di utilizzo di una chiave per alcune operazioni di crittografia vengono inoltrate al AWS CloudHSM cluster per eseguire l'operazione. Sebbene un AWS CloudHSM cluster sia ospitato da AWS, si tratta di una soluzione single-tenant gestita e gestita direttamente dall'utente. Sei proprietario di gran parte della disponibilità e delle prestazioni delle chiavi KMS in un cluster. AWS CloudHSM Per vedere se un archivio chiavi AWS CloudHSM personalizzato è adatto alle tue esigenze, leggi [Gli archivi di chiavi AWS KMS personalizzati sono adatti a te?](#) sul blog sulla AWS sicurezza.

Archivio delle chiavi esterne

È possibile AWS KMS configurare l'utilizzo di un External Key Store (XKS), in cui le chiavi utente root vengono generate, archiviate e utilizzate in un sistema di gestione delle chiavi esterno a. Cloud AWS Le richieste di utilizzo AWS KMS di una chiave per alcune operazioni di crittografia vengono inoltrate al sistema ospitato esternamente per eseguire l'operazione. In particolare, le richieste vengono inoltrate a un proxy XKS nella rete, che quindi inoltra la richiesta al sistema crittografico utilizzato. Il proxy XKS è una specifica open source con cui chiunque può integrarsi. Molti fornitori commerciali di gestione delle chiavi supportano la specifica XKS Proxy. Poiché un archivio di chiavi esterno è ospitato dall'utente o da terze parti, l'utente possiede tutta la disponibilità, la durata e le prestazioni delle chiavi del sistema. Per vedere se un External Key Store è adatto alle tue esigenze, leggi [Announcing AWS KMS External Key Store \(XKS\)](#) sul blog AWS News.

AWS CloudHSM negozi chiave

Un AWS CloudHSM key store è un [archivio di chiavi personalizzato](#) supportato da un [AWS CloudHSM cluster](#). Quando ne crei uno AWS KMS key in un archivio di chiavi personalizzato, AWS KMS genera e archivia materiale chiave non estraibile per la chiave KMS in un AWS CloudHSM cluster di tua proprietà e gestione. Quando utilizzi una chiave KMS in un archivio di chiavi personalizzato, le [operazioni crittografiche](#) vengono eseguite nel cluster. HSMs Questa funzionalità combina la praticità e l'ampia integrazione di AWS KMS con il controllo aggiuntivo di un AWS CloudHSM cluster nel tuo. Account AWS

AWS KMS fornisce supporto completo per console e API per la creazione, l'utilizzo e la gestione degli archivi di chiavi personalizzati. È possibile utilizzare le chiavi KMS nell'archivio delle chiavi personalizzate nello stesso modo in cui si utilizza qualsiasi chiave KMS. Ad esempio, puoi utilizzare le chiavi KMS per generare chiavi di dati ed effettuare la crittografia dei dati. Puoi anche utilizzare le chiavi KMS nel tuo archivio chiavi personalizzato con AWS servizi che supportano le chiavi gestite dai clienti.

È necessario uno store di chiavi personalizzato?

Per la maggior parte degli utenti, l'archivio di AWS KMS chiavi predefinito, protetto da [moduli crittografici convalidati FIPS 140-3](#), soddisfa i requisiti di sicurezza. Non è richiesto un livello supplementare di responsabilità per la manutenzione o la dipendenza da un ulteriore servizio.

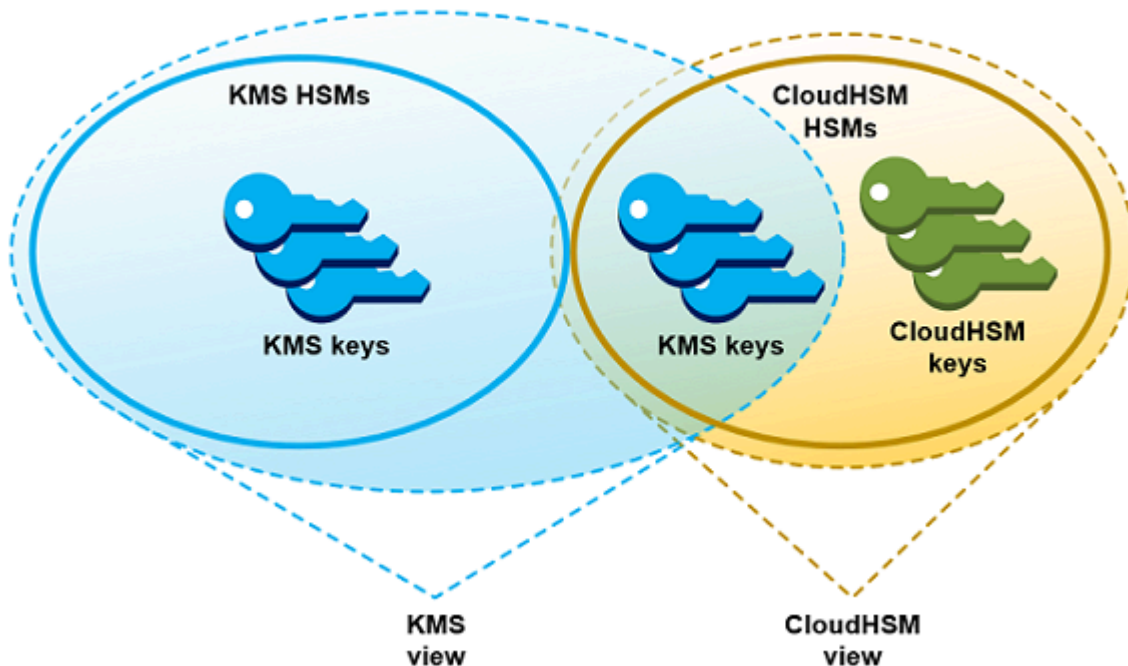
Tuttavia, è possibile prendere in considerazione la creazione di uno store di chiavi personalizzato qualora l'organizzazione possieda i seguenti requisiti:

- Hai chiavi che devono essere protette esplicitamente in un modulo HSM a tenant singolo o in un modulo HSM su cui hai il controllo diretto.
- È necessaria la possibilità di rimuovere immediatamente il materiale chiave da AWS KMS
- Devi essere in grado di controllare tutti gli usi delle tue chiavi indipendentemente da AWS KMS o AWS CloudTrail.

Come funzionano gli store di chiavi personalizzati?

Ogni archivio di chiavi personalizzato è associato a un AWS CloudHSM cluster del tuo Account AWS. Quando connetti l'archivio di chiavi personalizzato al relativo cluster, AWS KMS crea l'infrastruttura di rete per supportare la connessione. Quindi accede al AWS CloudHSM client chiave del cluster utilizzando le credenziali di un [utente crittografico dedicato](#) nel cluster.

Crei e gestisci i tuoi archivi di chiavi personalizzati AWS KMS e crei e gestisci i tuoi cluster HSM in AWS CloudHSM. Quando crei AWS KMS keys in un archivio di chiavi AWS KMS personalizzato, visualizzi e gestisci le chiavi KMS in AWS KMS. Tuttavia, in AWS CloudHSM è anche possibile visualizzare e gestire il proprio materiale chiave, esattamente come avviene per le altre chiavi nel cluster.



Puoi [creare chiavi KMS con crittografia simmetrica con materiale chiave generato dal tuo archivio di chiavi](#) AWS KMS personalizzato. Utilizza quindi le stesse tecniche per visualizzare e gestire le chiavi KMS nell'archivio chiavi personalizzato utilizzate per le chiavi KMS nell'archivio chiavi. AWS KMS Puoi controllare l'accesso con e le policy IAM e delle chiavi, creare tag e alias, abilitare e disabilitare chiavi KMS e pianificare l'eliminazione della chiave. Puoi utilizzare le chiavi KMS per [operazioni crittografiche](#) e utilizzarle con AWS servizi che si integrano con. AWS KMS

Inoltre, hai il pieno controllo del AWS CloudHSM cluster, inclusa la creazione, l'eliminazione HSMs e la gestione dei backup. Puoi utilizzare il AWS CloudHSM client e le librerie software supportate per visualizzare, controllare e gestire il materiale chiave per le tue chiavi KMS. Sebbene l'archivio chiavi personalizzato sia disconnesso, AWS KMS non possono accedervi e gli utenti non possono utilizzare le chiavi KMS nell'archivio chiavi personalizzato per operazioni crittografiche. Questo ulteriore livello di controllo rende gli store di chiavi personalizzati una soluzione potente per le aziende che lo richiedono.

Da dove iniziare?

Per creare e gestire un archivio di AWS CloudHSM chiavi, si utilizzano le funzionalità di e. AWS KMS AWS CloudHSM

1. Inizia in AWS CloudHSM. [Creare un cluster AWS CloudHSM attivo](#) oppure selezionare un cluster esistente. Il cluster deve avere almeno due zone di disponibilità attive HSMs in zone di disponibilità

- diverse. Creare poi un [account utente di crittografia dedicato \(CU, crypto user\)](#) in quel cluster per AWS KMS.
2. In AWS KMS, [crea un archivio di chiavi personalizzato](#) associato al AWS CloudHSM cluster selezionato. AWS KMS fornisce un'interfaccia di gestione completa che consente di creare, visualizzare, modificare ed eliminare gli archivi di chiavi personalizzati.
 3. Quando sei pronto per utilizzare il tuo archivio di chiavi personalizzato, [collegalo al AWS CloudHSM cluster associato](#). AWS KMS crea l'infrastruttura di rete necessaria per supportare la connessione. Effettua poi l'accesso al cluster tramite le credenziali dell'account crypto user (CU) dedicato, in modo da poter generare e gestire il materiale chiave nel cluster.
 4. Ora puoi [creare chiavi KMS di crittografia simmetrica nel tuo archivio delle chiavi personalizzate](#). Basta specificare l'archivio delle chiavi personalizzate in fase di creazione della chiave KMS.

Qualora durante questa procedura non si riesca a procedere, cercare assistenza nell'argomento [Risoluzione di problemi relativi a store delle chiavi personalizzate](#). Se non si trova risposta, utilizzare il collegamento di feedback nella parte inferiore di ogni pagina della guida o pubblicare una domanda nel [Forum di discussione AWS Key Management Service](#).

Quote

AWS KMS consente fino a [10 archivi di chiavi personalizzati](#) in ciascuna Account AWS regione, inclusi archivi [AWS CloudHSM chiavi e archivi chiavi esterni](#), indipendentemente dallo stato della connessione. Inoltre, sono previste quote di AWS KMS richiesta per l'[uso delle chiavi KMS in un AWS CloudHSM archivio di chiavi](#).

Prezzi

[Per informazioni sul costo degli archivi di chiavi AWS KMS personalizzati e delle chiavi gestite dai clienti in un archivio di chiavi personalizzato, consulta AWS Key Management Service i prezzi](#). Per informazioni sul costo dei AWS CloudHSM cluster e HSMs, consulta la sezione [AWS CloudHSM Prezzi](#).

Regioni

AWS KMS supporta i AWS CloudHSM key store in tutti i paesi in Regioni AWS cui AWS KMS è supportato, ad eccezione di Asia Pacifico (Melbourne), Cina (Pechino), Cina (Ningxia) ed Europa (Spagna).

Caratteristiche non supportate

AWS KMS non supporta le seguenti funzionalità negli archivi di chiavi personalizzati.

- [Chiavi KMS asimmetriche](#)
- [Chiavi KMS HMAC](#)
- [Chiavi KMS con materiale della chiave importato](#)
- [Rotazione automatica delle chiavi](#)
- [Chiavi multi-regione](#)

AWS CloudHSM concetti di archivio chiave

In questo argomento vengono descritti alcuni termini e concetti utilizzati negli archivi AWS CloudHSM chiave.

AWS CloudHSM archivio di chiavi

Un AWS CloudHSM key store è un archivio di [chiavi personalizzato](#) associato a un AWS CloudHSM cluster di cui sei proprietario e che gestisci. AWS CloudHSM i cluster sono supportati da moduli di sicurezza hardware (HSMs) certificati [FIPS 140-2 o FIPS 140-3 Livello 3](#).

Quando crei una chiave KMS nel tuo archivio chiavi, AWS KMS genera una AWS CloudHSM chiave simmetrica Advanced Encryption Standard (AES) a 256 bit, persistente e non esportabile nel cluster associato. AWS CloudHSM Questo materiale chiave non esce mai non crittografato. HSMs Quando si utilizza una chiave KMS in un archivio di AWS CloudHSM chiavi, le operazioni crittografiche vengono eseguite HSMs nel cluster.

AWS CloudHSM gli archivi di chiavi combinano la comoda e completa interfaccia di gestione delle chiavi AWS KMS con i controlli aggiuntivi forniti da un AWS CloudHSM cluster del tuo. Account AWS Questa funzionalità integrata consente di creare, gestire e utilizzare le chiavi KMS AWS KMS mantenendo il pieno controllo del materiale in HSMs cui sono archiviate le chiavi, inclusa la gestione dei cluster e dei HSMs backup. Puoi utilizzare la AWS KMS console e APIs gestire l'archivio delle chiavi e le AWS CloudHSM relative chiavi KMS. È inoltre possibile utilizzare la AWS CloudHSM console APIs, il software client e le librerie software associate per gestire il cluster associato.

È possibile [visualizzare e gestire](#) l'archivio delle AWS CloudHSM chiavi, [modificarne le proprietà](#) e [connetterlo](#) e [disconnetterlo](#) dal AWS CloudHSM cluster associato. Se devi [eliminare un archivio AWS CloudHSM chiavi](#), devi prima eliminare le chiavi KMS nell'archivio AWS CloudHSM chiavi pianificandone l'eliminazione e aspettando la scadenza del periodo di prova. L'eliminazione del

AWS CloudHSM key store rimuove la risorsa dal cluster AWS KMS, ma non ha alcun effetto. AWS CloudHSM

AWS CloudHSM ammasso

Ogni archivio di AWS CloudHSM chiavi è associato a un AWS CloudHSM cluster. Quando ne crei uno AWS KMS key nel tuo AWS CloudHSM key store, AWS KMS crea il relativo materiale chiave nel cluster associato. Quando utilizzi una chiave KMS nell'archivio delle chiavi di AWS CloudHSM, l'operazione di crittografia viene eseguita nel cluster associato.

Ogni AWS CloudHSM cluster può essere associato a un solo archivio di AWS CloudHSM chiavi. Il cluster scelto non può essere associato a un altro archivio AWS CloudHSM chiavi o condividere una cronologia di backup con un cluster associato a un altro archivio AWS CloudHSM chiavi. Il cluster deve essere inizializzato e attivo e deve trovarsi nella stessa Account AWS regione dell'archivio delle AWS CloudHSM chiavi. È possibile creare un nuovo cluster o utilizzarne uno esistente. AWS KMS non necessita dell'uso esclusivo del cluster. Per creare chiavi KMS nell'archivio delle AWS CloudHSM chiavi, il cluster associato deve contenerne almeno due attive HSMs. Per tutte le altre operazioni è richiesto un solo HSM.

Il AWS CloudHSM cluster viene specificato quando si crea l'archivio delle AWS CloudHSM chiavi e non è possibile modificarlo. Puoi tuttavia sostituire qualsiasi cluster che condivide una cronologia dei backup con il cluster originale. Ciò ti consente di eliminare il cluster, se necessario, e sostituirlo con un cluster creato a partire da uno dei relativi backup. L'utente mantiene il pieno controllo del AWS CloudHSM cluster associato in modo da poter gestire utenti e chiavi, creare ed eliminare HSMs, utilizzare e gestire i backup.

Quando sei pronto per utilizzare l'archivio AWS CloudHSM delle chiavi, lo connetti al AWS CloudHSM cluster associato. Puoi connettere e disconnettere lo store delle chiavi personalizzate in qualsiasi momento. Quando un archivio delle chiavi personalizzate è connesso, puoi creare chiavi KMS e utilizzare quelle che contiene. Quando è disconnesso, puoi visualizzare e gestire l'archivio delle AWS CloudHSM chiavi e le relative chiavi KMS. Tuttavia, non è possibile creare nuove chiavi KMS o utilizzare le chiavi KMS nell'archivio AWS CloudHSM chiavi per operazioni crittografiche.

Crypto user (CU) **kmsuser**

Per creare e gestire il materiale chiave nel AWS CloudHSM cluster associato per tuo conto, AWS KMS utilizza un [utente AWS CloudHSM crittografico](#) dedicato (CU) nel cluster denominato. **kmsuser**. Il **kmsuser** CU è un account CU standard che viene automaticamente sincronizzato con tutti gli utenti del cluster e salvato HSMs nei backup del cluster.

Prima di creare il tuo AWS CloudHSM key store, [crei un account kmsuser CU](#) nel AWS CloudHSM cluster utilizzando il comando [user create](#) nella CLI di CloudhSM. Quindi, quando [crei il AWS CloudHSM key store](#), fornisci la password dell'`kmsuser` account a. AWS KMS. Quando si [connette l'archivio di chiavi personalizzato](#), AWS KMS accede al cluster come `kmsuser CU` e ne modifica la password. AWS KMS crittografa la `kmsuser` password prima di archivarla in modo sicuro. Quando la password viene ruotata, la nuova password viene crittografata e archiviata in modo analogo.

AWS KMS rimane connesso `kmsuser` finché l'archivio delle AWS CloudHSM chiavi è connesso. Non devi utilizzare questo account utente di crittografia per altri scopi. Mantieni tuttavia il controllo finale dell'account utente di crittografia `kmsuser`. In qualsiasi momento, puoi [trovare le chiavi](#) che `kmsuser` possiede. Se necessario, è possibile [disconnettere l'archivio delle chiavi personalizzate](#), modificare la password di `kmsuser`, [accedere al cluster come kmsuser](#) e visualizzare e gestire le chiavi di cui l'`kmsuser` è proprietario.

Per istruzioni sulla creazione dell'account utente di crittografia `kmsuser`, consulta [Creazione del crypto user \(CU\)`kmsuser`](#).

Chiavi KMS in un archivio di AWS CloudHSM chiavi

Puoi utilizzare l' AWS KMS API o per crearne una AWS KMS keys in un archivio di AWS CloudHSM chiavi. A questo proposito, utilizzi la stessa tecnica che utilizzeresti per qualsiasi chiave KMS. L'unica differenza è che è necessario identificare l'archivio delle AWS CloudHSM chiavi e specificare che l'origine del materiale chiave è il AWS CloudHSM cluster.

Quando si [crea una chiave KMS in un key store](#), AWS KMS crea la [AWS CloudHSM chiave](#) KMS in AWS KMS e genera un materiale chiave simmetrico Advanced Encryption Standard (AES) a 256 bit, persistente e non esportabile nel cluster associato. Quando si utilizza la AWS KMS chiave in un'operazione di crittografia, l'operazione viene eseguita nel cluster utilizzando la chiave AES basata sul cluster. AWS CloudHSM. Sebbene AWS CloudHSM supportino chiavi simmetriche e asimmetriche di diversi tipi, gli archivi di chiavi supportano solo chiavi di crittografia simmetriche AES AWS CloudHSM .

È possibile visualizzare le chiavi KMS in un archivio AWS CloudHSM chiavi della console e utilizzare le opzioni della AWS KMS console per visualizzare l'ID dell'archivio chiavi personalizzato. È inoltre possibile utilizzare l'[DescribeKey](#) operazione per trovare l'ID dell'archivio AWS CloudHSM chiavi e l'ID AWS CloudHSM del cluster.

Le chiavi KMS in un AWS CloudHSM key store funzionano esattamente come tutte le chiavi KMS in esso contenute. AWS KMS. Gli utenti autorizzati devono disporre delle stesse autorizzazioni per

utilizzare e gestire le chiavi KMS. Utilizzi le stesse procedure della console e le stesse operazioni API per visualizzare e gestire le chiavi KMS in un AWS CloudHSM archivio di chiavi. Queste includono l'abilitazione e la disabilitazione di chiavi KMS, la creazione e l'utilizzo di tag e alias e l'impostazione e la modifica di policy IAM e di policy chiave. È possibile utilizzare le chiavi KMS in un AWS CloudHSM archivio di chiavi per operazioni crittografiche e utilizzarle con [AWS servizi integrati](#) che supportano l'uso di chiavi gestite dal cliente. Tuttavia, non è possibile abilitare la [rotazione automatica](#) delle chiavi o [importare materiale chiave in una chiave](#) KMS in un archivio di chiavi. AWS CloudHSM

Lo stesso processo viene utilizzato anche per [pianificare l'eliminazione](#) di una chiave KMS in un archivio di chiavi. AWS CloudHSM Dopo la scadenza del periodo di attesa, AWS KMS elimina la chiave KMS da KMS. Quindi fa del suo meglio per eliminare il materiale chiave per la chiave KMS dal cluster associato. AWS CloudHSM È tuttavia possibile che sia necessario [eliminare manualmente il materiale della chiave orfano](#) dal cluster e dai relativi backup.

Controlla l'accesso al tuo AWS CloudHSM key store

Utilizzi le policy IAM per controllare l'accesso al tuo AWS CloudHSM key store e al tuo AWS CloudHSM cluster. Puoi utilizzare le policy chiave, le policy IAM e le concessioni per controllare l'accesso a esse AWS KMS keys nel tuo AWS CloudHSM key store. Ti consigliamo di concedere a utenti, gruppi e ruoli soltanto le autorizzazioni necessarie per le attività che sono supposti eseguire.

Per supportare i tuoi AWS CloudHSM key store, hai AWS KMS bisogno dell'autorizzazione per ottenere informazioni sui tuoi AWS CloudHSM cluster. È inoltre necessaria l'autorizzazione per creare l'infrastruttura di rete che collega l'archivio delle AWS CloudHSM chiavi al relativo AWS CloudHSM cluster. Per ottenere queste autorizzazioni, AWS KMS crea il ruolo `AWSServiceRoleForKeyManagementServiceCustomKeyStores` collegato al servizio nel tuo Account AWS. Per ulteriori informazioni, consulta [Autorizzazione AWS KMS alla gestione AWS CloudHSM e alle risorse Amazon EC2](#).

Durante la progettazione del tuo archivio di AWS CloudHSM chiavi, assicurati che i responsabili che lo utilizzano e lo gestiscono dispongano solo delle autorizzazioni necessarie. L'elenco seguente descrive le autorizzazioni minime richieste per i gestori e gli AWS CloudHSM utenti dell'archivio chiavi.

- I responsabili che creano e gestiscono l'archivio delle AWS CloudHSM chiavi richiedono la seguente autorizzazione per utilizzare le operazioni dell'API dell'archivio AWS CloudHSM chiavi.
 - `cloudhsm:DescribeClusters`
 - `kms>CreateCustomKeyStore`

- kms:ConnectCustomKeyStore
 - kms>DeleteCustomKeyStore
 - kms:DescribeCustomKeyStores
 - kms:DisconnectCustomKeyStore
 - kms:UpdateCustomKeyStore
 - iam:CreateServiceLinkedRole
- I responsabili che creano e gestiscono il AWS CloudHSM cluster associato all'archivio delle AWS CloudHSM chiavi necessitano dell'autorizzazione per creare e inizializzare un cluster. AWS CloudHSM Ciò include l'autorizzazione a creare o utilizzare un Amazon Virtual Private Cloud (VPC), creare sottoreti e creare un'istanza Amazon. EC2 Potrebbero inoltre aver bisogno di creare HSMs, eliminare e gestire i backup. Per un elenco delle autorizzazioni necessarie, consulta [Identity and access management per AWS CloudHSM](#) nella Guida per l'utente di AWS CloudHSM .
 - I responsabili che creano e gestiscono AWS KMS keys nel tuo archivio delle AWS CloudHSM chiavi richiedono [le stesse autorizzazioni](#) di coloro che creano e gestiscono qualsiasi chiave KMS. AWS KMS La [politica delle chiavi predefinita](#) per una chiave KMS in un archivio AWS CloudHSM chiavi è identica alla politica delle chiavi predefinita per le chiavi KMS in. AWS KMS Il [controllo degli accessi basato sugli attributi](#) (ABAC), che utilizza tag e alias per controllare l'accesso alle chiavi KMS, è efficace anche sulle chiavi KMS negli archivi di chiavi. AWS CloudHSM
 - [I responsabili che utilizzano le chiavi KMS nell'archivio delle chiavi per le operazioni crittografiche necessitano dell'autorizzazione per eseguire l'operazione di crittografia con la AWS CloudHSM chiave KMS, ad esempio KMS:Decrypt.](#) Puoi fornire queste autorizzazioni in una policy delle chiavi, una policy IAM. Tuttavia, non hanno bisogno di autorizzazioni aggiuntive per utilizzare una chiave KMS in un archivio di chiavi. AWS CloudHSM

Crea un archivio di AWS CloudHSM chiavi

Puoi creare uno o più archivi di AWS CloudHSM chiavi nel tuo account. Ogni archivio di AWS CloudHSM chiavi è associato a un AWS CloudHSM cluster nella stessa Account AWS regione. Prima di creare l'archivio delle chiavi di AWS CloudHSM , devi [assemblare i prerequisiti](#). Quindi, prima di poter utilizzare l'archivio AWS CloudHSM delle chiavi, è necessario [collegarlo](#) al relativo AWS CloudHSM cluster.

Note

KMS non può comunicare IPv6 con gli archivi di AWS CloudHSM chiavi.

Se si tenta di creare un archivio AWS CloudHSM chiavi con tutti gli stessi valori di proprietà di un archivio AWS CloudHSM chiavi disconnesso esistente, AWS KMS non crea un nuovo archivio AWS CloudHSM chiavi e non genera un'eccezione né visualizza un errore. AWS KMS Riconosce invece il duplicato come probabile conseguenza di un nuovo tentativo e restituisce l'ID dell'archivio di chiavi esistente. AWS CloudHSM

Non è necessario collegare immediatamente l'archivio delle AWS CloudHSM chiavi. Puoi lasciarlo disconnesso fino a che non sei pronto a utilizzarlo. Tuttavia, per verificare che sia configurato correttamente, puoi [connetterlo](#), [visualizzarne lo stato di connessione](#) e quindi [disconnetterlo](#).

Argomenti

- [Assemblare i prerequisiti](#)
- [Crea un nuovo archivio di AWS CloudHSM chiavi](#)

Assemblare i prerequisiti

Ogni archivio di AWS CloudHSM chiavi è supportato da un AWS CloudHSM cluster. Per creare un AWS CloudHSM key store, è necessario specificare un AWS CloudHSM cluster attivo che non sia già associato a un altro key store. È inoltre necessario creare un utente crittografico (CU) dedicato nei cluster HSMs che AWS KMS possa utilizzare per creare e gestire le chiavi per conto dell'utente.

Prima di creare un archivio di AWS CloudHSM chiavi, procedi come segue:

Seleziona un AWS CloudHSM cluster

Ogni archivio di AWS CloudHSM chiavi è [associato esattamente a un AWS CloudHSM cluster](#). Quando crei AWS KMS keys nel tuo AWS CloudHSM key store, AWS KMS crea i metadati della chiave KMS, come un ID e un Amazon Resource Name (ARN) in. AWS KMS Quindi crea il materiale chiave nel cluster HSMs associato. È possibile [creare un nuovo AWS CloudHSM](#) cluster o utilizzarne uno esistente. AWS KMS non richiede l'accesso esclusivo al cluster.

Il AWS CloudHSM cluster selezionato è associato in modo permanente all'archivio delle AWS CloudHSM chiavi. Dopo aver creato l'archivio delle AWS CloudHSM chiavi, è possibile [modificare l'ID](#) del cluster associato, ma il cluster specificato deve condividere una cronologia di backup con il cluster originale. Per utilizzare un cluster non correlato, è necessario creare un nuovo archivio di AWS CloudHSM chiavi.

Il AWS CloudHSM cluster selezionato deve avere le seguenti caratteristiche:

- Il cluster deve essere attivo.

È necessario creare il cluster, inizializzarlo, installare il software AWS CloudHSM client per la piattaforma e quindi attivare il cluster. Per istruzioni dettagliate, consulta la sezione [Nozioni di base su AWS CloudHSM](#) della Guida per l'utente di AWS CloudHSM .

- Il TLS reciproco (mTLS) non è abilitato.

KMS non supporta MTL per i cluster. Questa impostazione non deve essere abilitata.

- Il cluster deve trovarsi nello stesso account e nella stessa regione dell'archivio delle AWS CloudHSM chiavi. Non è possibile associare un archivio di AWS CloudHSM chiavi in una regione a un cluster in un'altra regione. Per creare un'infrastruttura chiave in più regioni, è necessario creare archivi e cluster di AWS CloudHSM chiavi in ciascuna regione.
- Il cluster non può essere associato a un altro archivio chiavi personalizzate nello stesso account e nella stessa regione. Ogni archivio di AWS CloudHSM chiavi nell'account e nella regione deve essere associato a un AWS CloudHSM cluster diverso. Non puoi specificare un cluster che è già associato a uno store delle chiavi personalizzate o un cluster che condivide una cronologia dei backup con un cluster associato. I cluster che condividono una cronologia dei backup hanno lo stesso certificato di cluster. Per visualizzare il certificato del cluster di un cluster, utilizza la AWS CloudHSM console o l'[DescribeClusters](#) operazione.

Se [effettui il backup in un cluster AWS CloudHSM in una regione differente](#), tale cluster viene considerato diverso e puoi associare il backup a un archivio chiavi personalizzate nella relativa regione. Tuttavia, le chiavi KMS nei due archivi di chiavi personalizzati non sono interoperabili, anche se hanno la stessa chiave di supporto. AWS KMS associa i metadati al testo cifrato in modo che possa essere decrittografato solo dalla chiave KMS che lo ha crittografato.

- Il cluster deve essere configurato con [sottoreti private](#) in almeno due zone di disponibilità nella regione. Poiché non AWS CloudHSM è supportato in tutte le zone di disponibilità, si consiglia di creare sottoreti private in tutte le zone di disponibilità della regione. Non puoi riconfigurare le sottoreti di un cluster esistente, ma puoi [creare un cluster a partire da un backup](#) con varie sottoreti nella configurazione del cluster.

 Important

Dopo aver creato l'archivio AWS CloudHSM delle chiavi, non eliminare nessuna delle sottoreti private configurate per il relativo cluster. AWS CloudHSM Se AWS KMS non riesci a trovare tutte le sottoreti nella configurazione del cluster, i tentativi di

[connessione all'archivio chiavi personalizzato hanno esito negativo e viene](#) generato un SUBNET_NOT_FOUND errore di connessione. Per informazioni dettagliate, consultare [Come correggere un errore di connessione](#).

- Il [gruppo di sicurezza per il cluster](#) (cloudhsm-cluster-*<cluster-id>*-sg) deve includere regole in entrata e regole in uscita che consentano il traffico TCP sulle IPv4 porte 2223-2225. La Source (Origine) nelle regole in entrata e la Destination (Destinazione) nelle regole in uscita devono corrispondere all'ID del gruppo di sicurezza. Tali regole sono configurate per impostazione predefinita quando si crea il cluster. Non eliminarle o modificarle.
- Il cluster deve contenere almeno due zone di disponibilità attive in zone di disponibilità diverse. HSMs Per verificare il numero di HSMs, utilizzare la AWS CloudHSM console o l'[DescribeClusters](#) operazione. Se necessario, puoi [aggiungere un HSM](#).

Ricerca del certificato trust anchor

Quando crei un archivio di chiavi personalizzato, devi caricare il certificato trust anchor per il AWS CloudHSM cluster su AWS KMS. AWS KMS necessita del certificato trust anchor per connettere l'archivio di AWS CloudHSM chiavi al cluster associato AWS CloudHSM .

Ogni AWS CloudHSM cluster attivo ha un certificato trust anchor. Quando [inizializzi il cluster](#), genera questo certificato, salvalo nel file `customerCA.crt` e copialo negli host che si connettono al cluster.

Crea l'utente **kmsuser** crittografico per AWS KMS

Per amministrare il tuo archivio di AWS CloudHSM chiavi, AWS KMS accedi all'account [utente kmsuser crittografico](#) (CU) nel cluster selezionato. Prima di creare il tuo AWS CloudHSM key store, devi creare il CU. kmsuser Quindi, quando crei il tuo archivio di AWS CloudHSM chiavi, fornisci la password kmsuser per AWS KMS. Ogni volta che colleghi l'archivio di AWS CloudHSM chiavi al AWS CloudHSM cluster associato, AWS KMS accede come password kmsuser e ruota la password kmsuser

Important

Non specificare l'opzione 2FA quando crei l'utente di crittografia kmsuser. In caso affermativo, AWS KMS non è possibile effettuare il login e l'archivio AWS CloudHSM delle chiavi non può essere connesso a questo AWS CloudHSM cluster. Una volta specificata, l'opzione 2FA non può essere annullata. Dovrai invece eliminare l'utente di crittografia e ricrearlo.

 Note

Le seguenti procedure utilizzano lo strumento da riga di comando AWS CloudHSM Client SDK 5, [CloudHSM CLI](#). La CLI di CloudhSM sostituisce con. key-handle key-reference

Il 1° gennaio 2025, AWS CloudHSM terminerà il supporto per gli strumenti da riga di comando di Client SDK 3, la CloudHSM Management Utility (CMU) e la Key Management Utility (KMU). Per ulteriori informazioni sulle differenze tra gli strumenti da riga di comando di Client SDK 3 e lo strumento da riga di comando Client SDK 5, consulta [Migrare da Client SDK 3 CMU e KMU a Client SDK 5 CloudHSM CLI nella Guida per l'utente.AWS CloudHSM](#)

1. Segui le procedure introduttive descritte nell'argomento [Guida introduttiva all'interfaccia a riga di comando \(CLI\) di CloudHSM](#) della Guida per l'utente.AWS CloudHSM
2. Utilizzate il comando [user create per creare](#) una CU denominata. kmsuser

La password deve contenere da 7 a 32 caratteri alfanumerici, rispettare la distinzione tra maiuscole e minuscole e non includere caratteri speciali.

Il comando di esempio seguente crea una kmsuser CU.

```
aws-cloudhsm > user create --username kmsuser --role crypto-user
Enter password:
Confirm password:
{
  "error_code": 0,
  "data": {
    "username": "kmsuser",
    "role": "crypto-user"
  }
}
```

Crea un nuovo archivio di AWS CloudHSM chiavi

Dopo [aver assemblato i prerequisiti](#), è possibile creare un nuovo archivio di AWS CloudHSM chiavi nella AWS KMS console o utilizzando l'[CreateCustomKeyStore](#) operazione.

Utilizzo della console AWS KMS

Quando si crea un archivio di AWS CloudHSM chiavi in Console di gestione AWS, è possibile aggiungere e creare i [prerequisiti](#) come parte del flusso di lavoro. Tuttavia, il processo risulta più rapido se li hai assemblati in precedenza.

1. Accedi Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) su <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Archivi di chiavi personalizzate, Archivi di chiavi AWS CloudHSM .
4. Scegli Crea un archivio di chiavi.
5. Immettere un nome descrittivo per lo store delle chiavi personalizzate. Il nome deve essere univoco per tutti gli archivi delle chiavi personalizzate presenti nell'account.

Important

Non includere informazioni riservate o sensibili in questo campo. Questo campo può essere visualizzato in testo semplice nei log e in altri output. CloudTrail

6. Seleziona [un AWS CloudHSM cluster per l'archivio](#) delle chiavi. AWS CloudHSM Oppure, per creare un nuovo AWS CloudHSM cluster, scegli il link Crea un AWS CloudHSM cluster.

Il menu mostra AWS CloudHSM i cluster del tuo account e della tua regione che non sono già associati a un AWS CloudHSM key store. Il cluster deve [soddisfare i requisiti](#) per l'associazione con uno store delle chiavi personalizzate.

7. Scegli il file, quindi carica il certificato trust anchor per il AWS CloudHSM cluster che hai scelto. Si tratta del file `customerCA.crt` creato all'[inizializzazione del cluster](#).
8. Immettere la password del [crypto user \(CU\) kmsuser](#) creato nel cluster selezionato.
9. Scegli Create (Crea).

Quando la procedura ha esito positivo, il nuovo archivio AWS CloudHSM chiavi viene visualizzato nell'elenco degli archivi AWS CloudHSM chiave dell'account e della regione. Se ha esito negativo, viene visualizzato un messaggio di errore che descrive il problema e fornisce istruzioni su come

risolverlo. Per ulteriori informazioni, consulta [Risoluzione di problemi relativi a store delle chiavi personalizzate](#).

Se si tenta di creare un archivio AWS CloudHSM chiavi con tutti gli stessi valori di proprietà di un archivio AWS CloudHSM chiavi disconnesso esistente, AWS KMS non crea un nuovo archivio AWS CloudHSM chiavi e non genera un'eccezione né visualizza un errore. AWS KMS Riconosce invece il duplicato come probabile conseguenza di un nuovo tentativo e restituisce l'ID dell'archivio di chiavi esistente. AWS CloudHSM

Avanti: I nuovi archivi di AWS CloudHSM chiavi non vengono collegati automaticamente. Prima di poter creare AWS KMS keys nel AWS CloudHSM key store, è necessario [connettere l'archivio chiavi personalizzato](#) al AWS CloudHSM cluster associato.

Utilizzo dell' AWS KMS API

È possibile utilizzare l'[CreateCustomKeyStore](#) operazione per creare un nuovo archivio di AWS CloudHSM chiavi associato a un AWS CloudHSM cluster nell'account e nella regione. Questi esempi utilizzano l' AWS Command Line Interface (AWS CLI), ma puoi anche utilizzare qualsiasi linguaggio di programmazione supportato.

L'operazione `CreateCustomKeyStore` richiede i valori di parametro seguenti.

- `CustomKeyStoreName` — Un nome descrittivo per l'archivio di chiavi personalizzato che è unico nell'account.

Important

Non includere informazioni riservate o sensibili in questo campo. Questo campo può essere visualizzato in testo semplice nei CloudTrail registri e in altri output.

- `CloudHsmClusterId` — L'ID del cluster di un AWS CloudHSM cluster che [soddisfa i requisiti per un archivio di chiavi](#). AWS CloudHSM
- `KeyStorePassword` — La password dell'account `kmsuser` CU nel cluster specificato.
- `TrustAnchorCertificate` — Il contenuto del `customerCA.crt` file creato durante l'[inizializzazione del cluster](#).

L'esempio seguente utilizza un ID cluster fittizio. Prima di eseguire il comando, sostituiscilo con un ID cluster valido


```
$ aws kms create-custom-key-store
  --custom-key-store-name ExampleCloudHSMKeyStore \
  --cloud-hsm-cluster-id cluster-1a23b4cdefg \
  --key-store-password kmsPswd \
  --trust-anchor-certificate <certificate-goes-here>
```

Se si utilizza il AWS CLI, è possibile specificare il file del certificato trust anchor, anziché il relativo contenuto. Nell'esempio seguente, il file `customerCA.crt` si trova nella directory principale:

```
$ aws kms create-custom-key-store
  --custom-key-store-name ExampleCloudHSMKeyStore \
  --cloud-hsm-cluster-id cluster-1a23b4cdefg \
  --key-store-password kmsPswd \
  --trust-anchor-certificate file://customerCA.crt
```

Se l'operazione riesce, `CreateCustomKeyStore` restituisce l'ID store chiavi personalizzate, come illustrato nel seguente esempio di risposta.

```
{
  "CustomKeyStoreId": cks-1234567890abcdef0
}
```

Se l'operazione ha esito negativo, correggi l'errore indicato dall'eccezione e riprova. Per ulteriori informazioni, consulta [Risoluzione di problemi relativi a store delle chiavi personalizzate](#).

Se si tenta di creare un archivio AWS CloudHSM chiavi con tutti gli stessi valori di proprietà di un archivio AWS CloudHSM chiavi disconnesso esistente, AWS KMS non crea un nuovo archivio AWS CloudHSM chiavi e non genera un'eccezione né visualizza un errore. AWS KMS Riconosce invece il duplicato come probabile conseguenza di un nuovo tentativo e restituisce l'ID dell'archivio di chiavi esistente. AWS CloudHSM

Avanti: per utilizzare l'archivio delle AWS CloudHSM chiavi, [collegalo al relativo cluster](#). AWS CloudHSM

Visualizza un archivio di AWS CloudHSM chiavi

È possibile visualizzare gli archivi delle AWS CloudHSM chiavi in ogni account e regione utilizzando la AWS KMS console o l'[DescribeCustomKeyStores](#) operazione.

Utilizzo della AWS KMS console

Quando si visualizzano gli archivi delle AWS CloudHSM chiavi in Console di gestione AWS, è possibile visualizzare quanto segue:

- Il nome e l'ID dell'archivio delle chiavi personalizzate
- L'ID del AWS CloudHSM cluster associato
- Il numero di HSMs nel cluster
- Lo stato di connessione corrente

Il valore dello stato della connessione (Status) impostato su Disconnected indica che l'archivio chiavi personalizzato è nuovo e non è mai stato connesso oppure è stato [disconnesso intenzionalmente dal](#) relativo cluster. AWS CloudHSM Tuttavia, se i tentativi di utilizzare una chiave KMS in un archivio di chiavi personalizzate connesso falliscono, ciò potrebbe indicare un problema con l'archivio chiavi personalizzato o il relativo cluster. AWS CloudHSM Per assistenza, consulta [Come correggere una chiave KMS non funzionante](#).

Per visualizzare gli archivi di AWS CloudHSM chiavi in un determinato account e in una determinata regione, utilizzare la procedura seguente.

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Archivi di chiavi personalizzate, Archivi di chiavi AWS CloudHSM .

Per personalizzare la visualizzazione, fai clic sull'icona che raffigura un ingranaggio visualizzata sotto il pulsante Create key store (Crea store delle chiavi).

Utilizzando l'API AWS KMS

Per visualizzare i tuoi archivi di AWS CloudHSM chiavi, usa l'[DescribeCustomKeyStores](#) operazione. Per impostazione predefinita, questa operazione restituisce tutti gli store delle chiavi personalizzate presenti nell'account e nella regione. Puoi tuttavia utilizzare il parametro CustomKeyName o CustomKeyId (ma non entrambi) per limitare l'output a un determinato store delle chiavi personalizzate. Per gli archivi di AWS CloudHSM chiavi, l'output è costituito dall'ID e dal

nome dell'archivio chiavi personalizzati, dal tipo di archivio chiavi personalizzato, dall'ID del AWS CloudHSM cluster associato e dallo stato della connessione. Se lo stato della connessione indica un errore, l'output include un codice di errore che descrive il motivo del problema.

Gli esempi in questa sezione utilizzano [AWS Command Line Interface \(AWS CLI\)](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

Ad esempio, il comando seguente restituisce tutti gli store delle chiavi personalizzate presenti nell'account e nella regione. Per scorrere gli store delle chiavi personalizzate nell'output puoi utilizzare i parametri `Limit` e `Marker`.

```
$ aws kms describe-custom-key-stores
```

Il comando di esempio seguente utilizza il parametro `CustomKeyStoreName` per ottenere solo lo store delle chiavi personalizzate con il nome descrittivo `ExampleCloudHSMKeyStore`. Puoi utilizzare il parametro `CustomKeyStoreName` o `CustomKeyStoreId` (ma non entrambi) in ogni comando.

L'output di esempio seguente rappresenta un archivio di AWS CloudHSM chiavi connesso al relativo AWS CloudHSM cluster.

Note

Il `CustomKeyStoreType` campo è stato aggiunto alla `DescribeCustomKeyStores` risposta per distinguere gli archivi di AWS CloudHSM chiavi dagli archivi di chiavi esterni.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleCloudHSMKeyStore
{
  "CustomKeyStores": [
    {
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionState": "CONNECTED",
      "CreationDate": "1.499288695918E9",
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleCloudHSMKeyStore",
      "CustomKeyStoreType": "AWS_CLOUDHSM",
      "TrustAnchorCertificate": "<certificate appears here>"
    }
  ]
}
```

```
]
}
```

Un `ConnectionState` di `Disconnected` indica che un archivio chiavi personalizzato non è mai stato connesso o che è stato intenzionalmente [disconnesso dal relativo AWS CloudHSM cluster](#). Tuttavia, se i tentativi di utilizzare una chiave KMS in un archivio di AWS CloudHSM chiavi connesso falliscono, ciò potrebbe indicare un problema con l'archivio AWS CloudHSM chiavi o il relativo cluster. AWS CloudHSM Per assistenza, consulta [Come correggere una chiave KMS non funzionante](#).

Se il campo `ConnectionState` dello store delle chiavi personalizzate è `FAILED`, la risposta `DescribeCustomKeyStores` include un elemento `ConnectionErrorCode` che descrive il motivo dell'errore.

Ad esempio, nell'output seguente, il valore `INVALID_CREDENTIALS` indica che la connessione dello store delle chiavi di connessione non è riuscita in quanto la [password kmsuser non è valida](#). Per informazioni su questo e altri errori di connessione, consulta [Risoluzione di problemi relativi a store delle chiavi personalizzate](#).

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
  "CustomKeyStores": [
    {
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionErrorCode": "INVALID_CREDENTIALS",
      "ConnectionState": "FAILED",
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleCloudHSMKeyStore",
      "CustomKeyStoreType": "AWS_CLOUDHSM",
      "CreationDate": "1.499288695918E9",
      "TrustAnchorCertificate": "<certificate appears here>"
    }
  ]
}
```

Ulteriori informazioni:

- [Visualizza gli archivi di chiavi esterni](#)
- [Identifica le chiavi KMS negli AWS CloudHSM archivi delle chiavi](#)
- [Registrazione delle chiamate AWS KMS API con AWS CloudTrail](#)

Modifica le impostazioni del AWS CloudHSM key store

È possibile modificare le impostazioni di un AWS CloudHSM key store esistente. L'archivio chiavi personalizzato deve essere disconnesso dal relativo AWS CloudHSM cluster.

Per modificare le impostazioni AWS CloudHSM del key store:

1. [Disconnettere lo store delle chiavi personalizzate](#) dal relativo cluster AWS CloudHSM .

Mentre l'archivio chiavi personalizzato è disconnesso, non è possibile creare AWS KMS keys (chiavi KMS) nell'archivio chiavi personalizzato e non è possibile utilizzare le chiavi KMS in esso contenute per operazioni crittografiche.

2. Modifica una o più impostazioni dell'archivio chiavi. AWS CloudHSM

Puoi modificare le impostazioni seguenti in uno store delle chiavi personalizzate:

Il nome descrittivo dello store delle chiavi personalizzate

Immetti un nuovo nome descrittivo. Il nuovo nome deve essere univoco tra tutti gli archivi di chiavi personalizzati presenti nel tuo Account AWS.

Important

Non includere informazioni riservate o sensibili in questo campo. Questo campo può essere visualizzato in testo semplice nei CloudTrail log e in altri output.

L'ID del cluster associato. AWS CloudHSM

Modifica questo valore per sostituire un AWS CloudHSM cluster correlato con quello originale. È possibile utilizzare questa funzionalità per riparare un archivio di chiavi personalizzato se il relativo AWS CloudHSM cluster viene danneggiato o eliminato.

Specificate un AWS CloudHSM cluster che condivida una cronologia di backup con il cluster originale e [soddisfi i requisiti per l'associazione](#) a un archivio di chiavi personalizzato, di cui due attivi HSMs in diverse zone di disponibilità. I cluster che condividono una cronologia dei backup hanno lo stesso certificato di cluster. Per visualizzare il certificato del cluster di un cluster, utilizzare l'[DescribeClusters](#) operazione. Non puoi utilizzare la funzionalità di modifica per associare lo store delle chiavi personalizzate a un cluster AWS CloudHSM non correlato.

La password corrente del [crypto user \(CU\) kmsuser](#)

Indica AWS KMS la password corrente della kmsuser CU nel AWS CloudHSM cluster. Questa azione non modifica la password della kmsuser CU nel AWS CloudHSM cluster.

Se modificate la password della kmsuser CU nel AWS CloudHSM cluster, utilizzate questa funzione per indicare AWS KMS la nuova kmsuser password. In caso contrario, AWS KMS non può accedere al cluster e tutti i tentativi di connessione dello store delle chiavi personalizzate al cluster hanno esito negativo.

3. [Riconnettere lo store delle chiavi personalizzate](#) al relativo cluster AWS CloudHSM .

Modifica le impostazioni del tuo key store

È possibile modificare le impostazioni del AWS CloudHSM key store nella AWS KMS console o utilizzando l'[UpdateCustomKeyStore](#) operazione.

Utilizzo della AWS KMS console

Quando si modifica un archivio di AWS CloudHSM chiavi, è possibile modificare uno qualsiasi dei valori configurabili.

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) su <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Archivi di chiavi personalizzate, Archivi di chiavi AWS CloudHSM .
4. Scegliete la riga del AWS CloudHSM key store che desiderate modificare.

Se il valore nella colonna Stato connessione non è Disconnesso, devi scollegare l'archivio di chiavi personalizzate per poterlo modificare. Dal menu Key store actions (Operazioni per l'archivio delle chiavi), scegli Disconnect (Disconnetti).

Quando un archivio AWS CloudHSM chiavi è disconnesso, puoi gestire l'archivio AWS CloudHSM chiavi e le relative chiavi KMS, ma non puoi creare o utilizzare le chiavi KMS nell'AWS CloudHSM archivio chiavi.

5. Dal menu Key store actions (Operazioni per l'archivio delle chiavi), scegli Edit (Modifica).

6. Effettuare una o più delle operazioni seguenti.

- Digitare un nuovo nome descrittivo per lo store delle chiavi personalizzate.
- Digita l'ID del cluster correlato. AWS CloudHSM
- Digita la password corrente dell'utente kmsuser crittografico nel AWS CloudHSM cluster associato.

7. Scegli Save (Salva).

Se la procedura ha esito positivo, un messaggio descrive le impostazioni modificate. Se ha esito negativo, viene visualizzato un messaggio di errore che descrive il problema e fornisce istruzioni su come risolverlo. Per ulteriori informazioni, consulta [Risoluzione di problemi relativi a store delle chiavi personalizzate](#).

8. [Riconnettere lo store delle chiavi personalizzate](#).

Per utilizzare l'archivio AWS CloudHSM chiavi, è necessario ricollegarlo dopo la modifica. Puoi lasciare disconnesso l'archivio delle chiavi di AWS CloudHSM , [Tuttavia, mentre è disconnesso, non è possibile creare chiavi KMS nell'archivio AWS CloudHSM chiavi o utilizzare le chiavi KMS nell'archivio AWS CloudHSM chiavi nelle operazioni crittografiche](#).

Utilizzando l'API AWS KMS

Per modificare le proprietà di un archivio di AWS CloudHSM chiavi, utilizzare l'[UpdateCustomKeyStore](#) operazione. Puoi modificare più proprietà di un store delle chiavi personalizzate nello stesso comando. Se l'operazione ha esito positivo, AWS KMS restituisce una risposta HTTP 200 e un oggetto JSON senza proprietà. Per verificare che le modifiche siano effettive, utilizzate l'[DescribeCustomKeyStores](#) operazione.

Gli esempi in questa sezione utilizzano [AWS Command Line Interface \(AWS CLI\)](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

Inizia utilizzando [DisconnectCustomKeyStore](#) per [disconnettere l'archivio di chiavi personalizzato](#) dal relativo AWS CloudHSM cluster. Sostituisci l'ID archivio chiavi personalizzate di esempio, cks-1234567890abcdef0, con un ID effettivo.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Il primo esempio utilizza [UpdateCustomKeyStore](#) per modificare il nome descrittivo del AWS CloudHSM key store inDevelopmentKeys. Il comando utilizza il CustomKeyId parametro

per identificare l'archivio AWS CloudHSM chiavi e CustomKeyStoreName specificare il nuovo nome per l'archivio chiavi personalizzato.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --new-custom-key-store-name DeveLopmentKeys
```

L'esempio seguente modifica il cluster associato a un archivio AWS CloudHSM chiavi in un altro backup dello stesso cluster. Il comando utilizza il CustomKeyId parametro per identificare l'archivio delle AWS CloudHSM chiavi e il CloudHsmClusterId parametro per specificare il nuovo ID del cluster.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --cloud-hsm-cluster-id cluster-1a23b4cdefg
```

L'esempio seguente indica AWS KMS che la kmsuser password corrente èExamplePassword. Il comando utilizza il CustomKeyId parametro per identificare l'archivio delle AWS CloudHSM chiavi e il KeyStorePassword parametro per specificare la password corrente.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --key-store-password ExamplePassword
```

Il comando finale riconnette l'archivio AWS CloudHSM chiavi al relativo AWS CloudHSM cluster. Puoi lasciare l'archivio delle chiavi personalizzate disconnesso, ma devi connetterlo per poter creare nuove chiavi KMS o utilizzare le chiavi KMS esistenti per [operazioni di crittografia](#). Sostituisci l'ID store chiavi personalizzate di esempio con un ID effettivo.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Connect a AWS CloudHSM key store

I nuovi archivi di AWS CloudHSM chiavi non sono collegati. Prima di poterlo creare e utilizzare AWS KMS keys nell'archivio delle AWS CloudHSM chiavi, è necessario collegarlo al AWS CloudHSM cluster associato. È possibile connettere e disconnettere l'archivio AWS CloudHSM delle chiavi in qualsiasi momento e [visualizzarne lo stato della connessione](#).

Non è necessario collegare il proprio AWS CloudHSM key store. È possibile lasciare un archivio AWS CloudHSM chiavi in uno stato disconnesso a tempo indeterminato e collegarlo solo quando è necessario utilizzarlo. Puoi tuttavia testare la connessione periodicamente per verificare che le impostazioni sono corrette e che non vi sono problemi di connessione dello store.

 Note

AWS CloudHSM gli archivi chiavi hanno uno stato di DISCONNECTED connessione solo quando l'archivio chiavi non è mai stato connesso o lo si disconnette esplicitamente. Se lo stato di connessione del AWS CloudHSM keystore è lo stesso CONNECTED ma hai problemi a utilizzarlo, assicurati che il AWS CloudHSM cluster associato sia attivo e ne contenga almeno uno attivo. HSMs Per informazioni sugli errori di connessione, consulta [the section called “Risoluzione di problemi relativi a store delle chiavi personalizzate”](#).

Quando ti connetti a un AWS CloudHSM key store, AWS KMS trova il AWS CloudHSM cluster associato, si connette ad esso, accede al AWS CloudHSM client come [utente kmsuser crittografico](#) (CU), quindi ruota la password. kmsuser AWS KMS rimane connesso al AWS CloudHSM client finché l'archivio delle AWS CloudHSM chiavi è connesso.

Per stabilire la connessione, AWS KMS crea un [gruppo di sicurezza](#) denominato kms-*<custom key store ID>* nel cloud privato virtuale (VPC) del cluster. Il gruppo di sicurezza ha un'unica regola che consente il traffico in entrata dal gruppo di sicurezza del cluster. AWS KMS crea anche un'[interfaccia di rete elastica](#) (ENI) in ogni zona di disponibilità della sottorete privata per il cluster. AWS KMS aggiunge ENIs al gruppo di kms-*<cluster ID>* sicurezza e al gruppo di sicurezza per il cluster. La descrizione di ogni ENI è KMS managed ENI for cluster *<cluster-ID>*.

Il completamento del processo di connessione può richiedere fino a 20 minuti.

Prima di connettere il AWS CloudHSM key store, verificate che soddisfi i requisiti.

- Il AWS CloudHSM cluster associato deve contenere almeno un HSM attivo. Per trovare il numero di HSMs nel cluster, visualizza il cluster nella AWS CloudHSM console o usa l'[DescribeClusters](#) operazione. Se necessario, puoi [aggiungere un HSM](#).
- Il cluster deve disporre di un account [utente kmsuser crittografico](#) (CU), ma tale CU non può essere registrato nel cluster quando si connette l'archivio delle AWS CloudHSM chiavi. Per informazioni su come effettuare la disconnessione, consulta [Come scollegarsi e riconnettersi](#).
- Lo stato di connessione del AWS CloudHSM key store non può essere o. DISCONNECTING FAILED Per visualizzare lo stato della connessione, utilizzare la AWS KMS console o la [DescribeCustomKeyStores](#) risposta. Se lo stato della connessione è FAILED, disconnetti l'archivio delle chiavi personalizzate, correggi il problema e riconnettilo.

Per informazioni sugli errori di connessione, consulta [Come correggere un errore di connessione](#).

Quando il tuo archivio AWS CloudHSM chiavi è connesso, puoi [creare chiavi KMS al suo interno e utilizzare le chiavi](#) KMS esistenti nelle operazioni [crittografiche](#).

Connect e riconnettiti al tuo AWS CloudHSM key store

È possibile connettere o ricollegare l'archivio AWS CloudHSM chiavi nella AWS KMS console o utilizzando l'[ConnectCustomKeyStore](#) operazione.

Utilizzo della console AWS KMS

Per connettere un AWS CloudHSM key store in Console di gestione AWS, iniziate selezionando il AWS CloudHSM key store dalla pagina Custom key store. Il completamento del processo di connessione può richiedere fino a 20 minuti.

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) su <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Archivi di chiavi personalizzate, Archivi di chiavi AWS CloudHSM .
4. Scegli la riga dell'archivio di AWS CloudHSM chiavi che desideri connettere.

Se lo stato di connessione dell'archivio AWS CloudHSM chiavi è Fallito, è necessario [disconnettere l'archivio chiavi personalizzato](#) prima di connetterlo.

5. Dal menu Key store actions (Operazioni per l'archivio delle chiavi), scegli Connect (Connetti).

AWS KMS avvia il processo di connessione dell'archivio chiavi personalizzato. Trova il cluster AWS CloudHSM associato, crea l'infrastruttura di rete necessaria, si connette alla stessa, accede al cluster AWS CloudHSM come utente di crittografia `kmsuser` ed esegue la rotazione della password `kmsuser`. Al termine dell'operazione, lo stato della connessione diventa Connesso.

Se l'operazione ha esito negativo, viene visualizzato un messaggio di errore che descrive il motivo del problema. Prima di riprovare a connetterti, [visualizza lo stato della connessione](#) del tuo AWS CloudHSM key store. Se è Non riuscito, devi [scollegare l'archivio di chiavi personalizzate](#) prima di ricollegarlo. Per assistenza, consulta [Risoluzione di problemi relativi a store delle chiavi personalizzate](#).

Successivo: [the section called “Crea una chiave KMS in un AWS CloudHSM archivio di chiavi”](#).

Utilizzo dell' AWS KMS API

Per connettere un archivio di AWS CloudHSM chiavi disconnesso, utilizzare l'[ConnectCustomKeyStore](#) operazione. Il AWS CloudHSM cluster associato deve contenere almeno un HSM attivo e lo stato della connessione non può esserlo. FAILED

Il completamento del processo di connessione può richiedere fino a 20 minuti. Se l'operazione non genera rapidamente un errore, l'operazione restituisce una risposta HTTP 200 e un oggetto JSON senza proprietà. Questa risposta iniziale non indica tuttavia che la connessione è riuscita. Per determinare lo stato di connessione dell'archivio chiavi personalizzato, consulta la [DescribeCustomKeyStores](#) risposta.

Gli esempi in questa sezione utilizzano [AWS Command Line Interface \(AWS CLI\)](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

Per identificare l'archivio AWS CloudHSM chiavi, utilizza il relativo ID dell'archivio chiavi personalizzato. È possibile trovare l'ID nella pagina Custom key stores della console o utilizzando l'[DescribeCustomKeyStores](#) operazione senza parametri. Prima di eseguire questo esempio, sostituisci l'ID di esempio con uno valido.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Per verificare che l'archivio AWS CloudHSM chiavi sia connesso, usa l'[DescribeCustomKeyStores](#) operazione. Per impostazione predefinita, questa operazione restituisce tutti gli store delle chiavi personalizzate presenti nel tuo account e nella regione. Puoi tuttavia utilizzare il parametro CustomKeyStoreName o CustomKeyStoreId (ma non entrambi) per limitare la risposta a determinati store delle chiavi personalizzate. Se il valore di ConnectionState è CONNECTED, indica che lo store delle chiavi personalizzate è connesso al relativo cluster AWS CloudHSM .

Note

Il CustomKeyStoreType campo è stato aggiunto alla DescribeCustomKeyStores risposta per distinguere gli archivi di AWS CloudHSM chiavi dagli archivi di chiavi esterni.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
  "CustomKeyStores": [
```

```

    "CustomKeyStoreId": "cks-1234567890abcdef0",
    "CustomKeyStoreName": "ExampleCloudHSMKeyStore",
    "CloudHsmClusterId": "cluster-1a23b4cdefg",
    "CustomKeyStoreType": "AWS_CLOUDHSM",
    "TrustAnchorCertificate": "<certificate string appears here>",
    "CreationDate": "1.499288695918E9",
    "ConnectionState": "CONNECTED"
  ],
}
```

Se il valore di `ConnectionState` è `Failed` (Non riuscito), l'elemento `ConnectionErrorCode` indica il motivo dell'errore. In questo caso, non è AWS KMS stato possibile trovare un AWS CloudHSM cluster nel tuo account con l'ID del cluster `cluster-1a23b4cdefg`. Se hai eliminato il cluster, puoi [ripristinarlo a partire da un backup](#) del cluster originale e quindi [modificare l'ID cluster](#) per lo store delle chiavi personalizzate. Per informazioni sulla risposta a un codice di errore di connessione, consulta [Come correggere un errore di connessione](#).

```

$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
  "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "CustomKeyStoreType": "AWS_CLOUDHSM",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "FAILED",
      "ConnectionErrorCode": "CLUSTER_NOT_FOUND"
    }
  ],
}
```

Disconnetti un archivio AWS CloudHSM chiavi

Quando si disconnette un archivio di AWS CloudHSM chiavi, si AWS KMS disconnette dal AWS CloudHSM client, si disconnette dal AWS CloudHSM cluster associato e rimuove l'infrastruttura di rete creata per supportare la connessione.

Mentre un archivio AWS CloudHSM chiavi è disconnesso, puoi gestire l'archivio chiavi e le AWS CloudHSM relative chiavi KMS, ma non puoi creare o utilizzare le chiavi KMS nell'archivio chiavi. AWS CloudHSM Lo stato di connessione dell'archivio delle chiavi è `DISCONNECTED` e lo [stato della chiave](#) delle chiavi KMS nell'archivio delle chiavi personalizzate è `Unavailable`, a meno che

non siano PendingDeletion. Puoi ricollegare l'archivio delle AWS CloudHSM chiavi in qualsiasi momento.

Note

AWS CloudHSM gli archivi chiavi hanno uno stato di DISCONNECTED connessione solo quando l'archivio chiavi non è mai stato connesso o lo si disconnette esplicitamente. Se lo stato di connessione del AWS CloudHSM keystore è lo stesso CONNECTED ma hai problemi a utilizzarlo, assicurati che il AWS CloudHSM cluster associato sia attivo e ne contenga almeno uno attivo. HSMs Per informazioni sugli errori di connessione, consulta [the section called “Risoluzione di problemi relativi a store delle chiavi personalizzate”](#).

Quando disconnetti un archivio delle chiavi personalizzate, le chiavi KMS nell'archivio diventano immediatamente inutilizzabili (in base alla coerenza finale). Tuttavia, le risorse crittografate con [chiavi di dati](#) protette dalla chiave KMS non sono interessate fino a quando la chiave KMS non viene nuovamente utilizzata, ad esempio per decrittografare la chiave dati. Questo problema riguarda i Servizi AWS, molti dei quali proteggono le risorse tramite le chiavi dati. Per informazioni dettagliate, consultare [In che modo le chiavi KMS inutilizzabili influiscono sulle chiavi dati](#).

Note

Quando un archivio delle chiavi personalizzate è disconnesso, tutti i tentativi di creare chiavi KMS nell'archivio delle chiavi personalizzate o di utilizzare le chiavi KMS in operazioni di crittografia avrà esito negativo. Questa azione può impedire agli utenti di archiviare e accedere ai dati sensibili.

Per valutare meglio l'effetto della disconnessione dell'archivio delle chiavi personalizzate, [identifica le chiavi KMS](#) nell'archivio delle chiavi personalizzate e [determinane l'utilizzo precedente](#).

È possibile disconnettere un archivio di AWS CloudHSM chiavi per motivi come i seguenti:

- Per effettuare una rotazione della password **kmsuser**. AWS KMS modifica la password **kmsuser** ogni volta che si connette al cluster AWS CloudHSM . Per forzare una rotazione della password, esegui la disconnessione e quindi una nuova connessione.
- Per controllare il materiale chiave per le chiavi KMS nel AWS CloudHSM cluster. Quando si disconnette l'archivio di chiavi personalizzato, si AWS KMS disconnette dall'account [utente](#)

[kmsuser crittografico](#) nel client. AWS CloudHSM Ciò ti consente di accedere al cluster come utente di crittografia `kmsuser` e di verificare e gestire il materiale chiave per la chiave KMS.

- Per disabilitare immediatamente tutte le chiavi KMS nell'archivio delle chiavi di AWS CloudHSM . È possibile [disabilitare e riattivare le chiavi KMS](#) in un AWS CloudHSM key store utilizzando l'Console di gestione AWS operazione o. [DisableKey](#) Queste operazioni vengono completate rapidamente, ma agiscono su una sola chiave KMS alla volta. La disconnessione dell'archivio AWS CloudHSM chiavi modifica immediatamente lo stato della chiave di tutte le chiavi KMS nell'archivio AWS CloudHSM chiavi in `Unavailable`, il che impedisce che vengano utilizzate in qualsiasi operazione crittografica.
- Per riparare un tentativo di connessione non riuscito. Se un tentativo di connessione a un AWS CloudHSM key store fallisce (lo stato di connessione dell'archivio chiavi personalizzato è `FAILED`), è necessario disconnettere l'archivio AWS CloudHSM chiavi prima di riprovare a connetterlo.

Disconnetti il tuo key store AWS CloudHSM

È possibile disconnettere l'archivio delle AWS CloudHSM chiavi nella AWS KMS console o utilizzando l'[DisconnectCustomKeyStore](#) operazione.

Disconnettiti utilizzando la console AWS KMS

Per disconnettere un archivio di AWS CloudHSM chiavi connesso nella AWS KMS console, inizia selezionando l'archivio AWS CloudHSM chiavi dalla pagina Custom Key Stores.

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) su <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Archivi di chiavi personalizzate, Archivi di chiavi AWS CloudHSM .
4. Scegli la riga relativa all'archivio delle chiavi esterne che vuoi disconnettere.
5. Dal menu Key store actions (Operazioni per l'archivio delle chiavi), scegli Disconnect (Disconnetti).

Al termine dell'operazione, lo stato della connessione da Disconnessione in corso diventa Disconnesso. Se l'operazione ha esito negativo, viene visualizzato un messaggio di errore che

descrive il problema e fornisce istruzioni su come risolverlo. Per ulteriori informazioni, consulta [Risoluzione di problemi relativi a store delle chiavi personalizzate](#).

Disconnettiti utilizzando l'API AWS KMS

Per disconnettere un archivio di AWS CloudHSM chiavi connesso, utilizzare l'[DisconnectCustomKeyStore](#) operazione. Se l'operazione ha esito positivo, AWS KMS restituisce una risposta HTTP 200 e un oggetto JSON senza proprietà.

Gli esempi in questa sezione utilizzano [AWS Command Line Interface \(AWS CLI\)](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

Questo esempio disconnette un archivio di AWS CloudHSM chiavi. Prima di eseguire questo esempio, sostituisci l'ID di esempio con uno valido.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Per verificare che l'archivio AWS CloudHSM chiavi sia disconnesso, utilizzare l'[DescribeCustomKeyStores](#) operazione. Per impostazione predefinita, questa operazione restituisce tutti gli store delle chiavi personalizzate presenti nel tuo account e nella regione. Puoi tuttavia utilizzare il parametro CustomKeyStoreName o CustomKeyStoreId (ma non entrambi) per limitare la risposta a determinati store delle chiavi personalizzate. Il ConnectionState valore di DISCONNECTED indica che questo AWS CloudHSM key store di esempio non è connesso al relativo AWS CloudHSM cluster.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
  "CustomKeyStores": [
    {
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionState": "DISCONNECTED",
      "CreationDate": "1.499288695918E9",
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "CustomKeyStoreType": "AWS_CLOUDHSM",
      "TrustAnchorCertificate": "<certificate string appears here>"
    }
  ],
}
```


Eliminare un archivio AWS CloudHSM chiavi

Quando elimini un archivio di AWS CloudHSM chiavi, AWS KMS elimina tutti i metadati relativi all'archivio AWS CloudHSM chiavi da KMS, incluse le informazioni sulla sua associazione a un cluster. AWS CloudHSM Questa operazione non influisce sul AWS CloudHSM cluster, sui suoi o sui suoi HSMs utenti. È possibile creare un nuovo archivio di AWS CloudHSM chiavi associato allo stesso AWS CloudHSM cluster, ma non è possibile annullare l'operazione di eliminazione.

È possibile eliminare solo un archivio di AWS CloudHSM chiavi disconnesso dal relativo AWS CloudHSM cluster e che non ne contiene. AWS KMS keys Prima di eliminare uno store delle chiavi personalizzate, esegui le operazioni descritte di seguito.

- Verifica che non avrai la necessità di utilizzare alcuna chiave KMS presente nell'archivio delle chiavi per qualsiasi [operazione di crittografia](#). Quindi [pianifica l'eliminazione](#) di tutte le chiavi KMS dall'archivio delle chiavi. Per informazioni su come trovare le chiavi KMS in un AWS CloudHSM archivio di chiavi, consulta. [Trova le chiavi KMS in un AWS CloudHSM archivio di chiavi](#)
- Verifica che tutte le chiavi KMS siano state eliminate. Per visualizzare le chiavi KMS in un AWS CloudHSM archivio di chiavi, consulta. [the section called “Identifica le chiavi KMS negli AWS CloudHSM archivi delle chiavi”](#)
- [Disconnetti l'archivio delle AWS CloudHSM chiavi](#) dal relativo AWS CloudHSM cluster.

Invece di eliminare l'archivio delle AWS CloudHSM chiavi, valuta la possibilità di [disconnetterlo](#) dal cluster associato. AWS CloudHSM Quando un AWS CloudHSM key store è disconnesso, puoi gestirlo e il AWS CloudHSM relativo. AWS KMS keys Tuttavia, non è possibile creare o utilizzare le chiavi KMS nell'archivio delle AWS CloudHSM chiavi. Puoi ricollegare l'archivio delle AWS CloudHSM chiavi in qualsiasi momento.

Elimina il tuo archivio di AWS CloudHSM chiavi

È possibile eliminare l'archivio delle AWS CloudHSM chiavi nella AWS KMS console o utilizzando l'[DeleteCustomKeyStore](#) operazione.

Utilizzo della AWS KMS console

Per eliminare un AWS CloudHSM key store in Console di gestione AWS, iniziate selezionando il AWS CloudHSM key store dalla pagina Custom key store.

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) su <https://console.aws.amazon.com/kms>.

2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel riquadro di navigazione, scegli Archivi di chiavi personalizzate, Archivi di chiavi AWS CloudHSM .
4. Trova la riga che rappresenta l'archivio di AWS CloudHSM chiavi che desideri eliminare. Se lo stato di connessione dell'archivio AWS CloudHSM chiavi non è Disconnesso, è necessario [disconnettere l'archivio AWS CloudHSM chiavi](#) prima di eliminarlo.
5. Dal menu Key store actions (Operazioni per l'archivio delle chiavi), scegli Delete (Elimina).

Al termine dell'operazione, viene visualizzato un messaggio di operazione riuscita e l'archivio AWS CloudHSM chiavi non viene più visualizzato nell'elenco degli archivi di chiavi. Se l'operazione ha esito negativo, viene visualizzato un messaggio di errore che descrive il problema e fornisce istruzioni su come risolverlo. Per ulteriori informazioni, consulta [Risoluzione di problemi relativi a store delle chiavi personalizzate](#).

Utilizzo dell'API AWS KMS

Per eliminare un archivio di AWS CloudHSM chiavi, utilizzare l'[DeleteCustomKeyStore](#) operazione. Se l'operazione ha esito positivo, AWS KMS restituisce una risposta HTTP 200 e un oggetto JSON senza proprietà.

Per iniziare, verifica che l'archivio delle AWS CloudHSM chiavi non ne contenga AWS KMS keys. Non puoi eliminare un archivio delle chiavi personalizzate che contiene chiavi KMS. Il primo comando di esempio utilizza [ListKeyse](#) [DescribeKey](#) cerca AWS KMS keys nell'archivio AWS CloudHSM chiavi con l'ID dell'archivio chiavi *cks-1234567890abcdef0* personalizzato di esempio. In questo caso, il comando non restituisce alcuna chiave KMS. In caso affermativo, utilizza l'[ScheduleKeyDeletion](#) operazione per pianificare l'eliminazione di ciascuna chiave KMS.

Bash

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;  
do aws kms describe-key --key-id $key |  
grep '"CustomKeyIdStoreId": "cks-1234567890abcdef0"' --context 100; done
```

PowerShell

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyIdStoreId -eq  
'cks-1234567890abcdef0'
```

Quindi, disconnetti l'archivio delle AWS CloudHSM chiavi. Questo comando di esempio utilizza l'[DisconnectCustomKeyStore](#) operazione per disconnettere un archivio di AWS CloudHSM chiavi dal relativo AWS CloudHSM cluster. Prima di eseguire questo comando, sostituisci l'ID store chiavi personalizzate di esempio con uno valido.

Bash

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

PowerShell

```
PS C:\> Disconnect-KMSCustomKeyStore -CustomKeyStoreId cks-1234567890abcdef0
```

Dopo la disconnessione dell'archivio chiavi personalizzato, è possibile utilizzare l'[DeleteCustomKeyStore](#) operazione per eliminarlo.

Bash

```
$ aws kms delete-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

PowerShell

```
PS C:\> Remove-KMSCustomKeyStore -CustomKeyStoreId cks-1234567890abcdef0
```

Risoluzione di problemi relativi a store delle chiavi personalizzate

AWS CloudHSM gli archivi di chiavi sono progettati per essere disponibili e resistenti. Tuttavia, ci sono alcune condizioni di errore che potresti dover correggere per mantenere operativo l'archivio AWS CloudHSM delle chiavi.

Argomenti

- [Come correggere chiavi KMS non disponibili](#)
- [Come correggere una chiave KMS non funzionante](#)
- [Come correggere un errore di connessione](#)
- [Come rispondere a un errore di un'operazione di crittografia](#)

- [Come correggere credenziali kmsuser non valide](#)
- [Come eliminare materiale della chiave orfano](#)
- [Come recuperare il materiale chiave eliminato per una chiave KMS](#)
- [Come accedere come kmsuser](#)

Come correggere chiavi KMS non disponibili

[Lo stato della chiave](#) di AWS KMS keys un AWS CloudHSM key store è in genere `Enabled`. Come tutte le chiavi KMS, lo stato delle chiavi cambia quando si disabilitano le chiavi KMS in un archivio AWS CloudHSM chiavi o se ne pianifica l'eliminazione. Tuttavia, a differenza delle altre chiavi KMS, le chiavi KMS in un archivio delle chiavi personalizzate possono anche avere lo [stato di chiave](#) `Unavailable`.

Lo stato di chiave `Unavailable` indica che la chiave KMS si trova in un archivio delle chiavi personalizzate che è stato intenzionalmente [disconnesso](#) e che gli eventuali tentativi di riconnetterlo non sono riusciti. Quando una chiave KMS non è disponibile, puoi visualizzare e gestire la chiave KMS, ma non utilizzarla per [operazioni di crittografia](#).

Per trovare lo stato di chiave di una chiave KMS, nella pagina Chiavi gestite cliente, visualizza il campo Stato della chiave KMS. Oppure, usa l'[DescribeKey](#) operazione e visualizza l'`KeyState` elemento nella risposta. Per informazioni dettagliate, consultare [Identifica e visualizza le chiavi](#).

Le chiavi KMS in uno store delle chiavi personalizzate disconnesso avranno lo stato di chiave `Unavailable` o `PendingDeletion`. Le chiavi KMS per le quali è stata pianificata l'eliminazione da un archivio delle chiavi personalizzate hanno lo stato della chiave `Pending Deletion`, anche quando tale archivio è disconnesso. Ciò ti consente di annullare l'eliminazione pianificata delle chiavi senza riconnettere lo store delle chiavi personalizzate.

Per correggere una chiave KMS non disponibile, [riconnetti l'archivio delle chiavi personalizzate](#). Dopo la riconnessione, per le chiavi KMS nello store delle chiavi personalizzate viene ripristinato lo stato di chiave precedente, ovvero `Enabled` o `Disabled`. Lo stato delle chiavi KMS in attesa di eliminazione rimane `PendingDeletion`. Tuttavia, se il problema persiste, l'[abilitazione e la disabilitazione di una chiave KMS](#) non disponibile non ne modifica lo stato. L'abilitazione o la disabilitazione ha effetto solo quando la chiave diventa disponibile.

Per informazioni sulle connessioni non riuscite, consulta [Come correggere un errore di connessione](#).

Come correggere una chiave KMS non funzionante

I problemi con la creazione e l'utilizzo delle chiavi KMS negli AWS CloudHSM archivi delle chiavi possono essere causati da un problema con l'archivio delle AWS CloudHSM chiavi, il AWS CloudHSM cluster associato, la chiave KMS o il relativo materiale chiave.

Quando un archivio AWS CloudHSM chiavi viene disconnesso dal relativo AWS CloudHSM cluster, lo stato delle chiavi KMS nell'archivio chiavi personalizzato è `Unavailable`. Tutte le richieste di creazione di chiavi KMS in un archivio di chiavi disconnesso restituiscono un'AWS CloudHSM eccezione `CustomKeyStoreInvalidStateException`. Tutte le richieste di crittografare, decrittografare, ricrittografare o generare chiavi di dati restituiscono un'eccezione `KMSInvalidStateException`. Per risolvere il problema, [ricollega l'archivio delle AWS CloudHSM chiavi](#).

Tuttavia, i tentativi di utilizzare una chiave KMS in un archivio di AWS CloudHSM chiavi per [operazioni crittografiche](#) potrebbero fallire anche se lo stato della chiave è `Enabled` e lo stato di connessione dell'archivio AWS CloudHSM chiavi è uguale a `Connected`. Ciò può essere dovuto a una qualsiasi delle condizioni seguenti.

- Il materiale della chiave per la chiave KMS potrebbe essere stato eliminato dal cluster AWS CloudHSM associato. Per indagare, [trova l'ID chiave](#) del materiale chiave per una chiave KMS e, se necessario, prova a [recuperare il](#) materiale chiave.
- Tutti HSMs sono stati eliminati dal AWS CloudHSM cluster associato all'archivio delle AWS CloudHSM chiavi. Per utilizzare una chiave KMS in un archivio di AWS CloudHSM chiavi in un'operazione crittografica, il relativo AWS CloudHSM cluster deve contenere almeno un HSM attivo. Per verificare il numero e lo stato di HSMs un AWS CloudHSM cluster, [usa la AWS CloudHSM console o l'operazione. `DescribeClusters`](#). Per aggiungere un HSM al cluster, usa la AWS CloudHSM console o l'[CreateHsm](#) operazione.
- Il AWS CloudHSM cluster associato al AWS CloudHSM key store è stato eliminato. Per risolvere il problema, [crea un cluster da un backup](#) che è correlato al cluster originale, ad esempio un backup del cluster originale o un backup utilizzato per creare il cluster originale. Quindi, [modifica l'ID cluster](#) nelle impostazioni relative allo store delle chiavi personalizzate. Per istruzioni, consultare [Come recuperare il materiale chiave eliminato per una chiave KMS](#).
- Il AWS CloudHSM cluster associato all'archivio di chiavi personalizzato non aveva sessioni PKCS #11 disponibili. Ciò si verifica in genere durante i periodi di traffico di espansione elevato, ovvero quando sono necessarie sessioni aggiuntive per gestire il traffico. Per rispondere a un'eccezione

KMSInternalException con un messaggio di errore relativo alle sessioni PKCS #11, torna indietro e riprova a eseguire la richiesta.

Come correggere un errore di connessione

Se si tenta di [connettere un AWS CloudHSM key store](#) al relativo AWS CloudHSM cluster, ma l'operazione non riesce, lo stato di connessione dell'archivio AWS CloudHSM chiavi cambia in FAILED. Per trovare lo stato di connessione di un AWS CloudHSM key store, usa la AWS KMS console o l'[DescribeCustomKeyStores](#) operazione.

In alternativa, alcuni tentativi di connessione si interrompono rapidamente a causa di errori di configurazione del cluster facilmente rilevati. In questo caso, lo stato della connessione è ancora DISCONNECTED. Questi errori restituiranno un messaggio di errore o un'[eccezione](#) che spiega perché il tentativo non è riuscito. Esamina la descrizione dell'eccezione e [i requisiti del cluster](#), risolvi il problema, [aggiorna l'archivio delle AWS CloudHSM chiavi](#), se necessario, e riprova a connetterti.

Quando lo stato della connessione è FAILED impostato, esegui l'[DescribeCustomKeyStores](#) operazione e visualizza l'ConnectionErrorCode elemento nella risposta.

Note

Quando lo stato di connessione di un AWS CloudHSM key store è FAILED, è necessario [disconnetterlo AWS CloudHSM prima di tentare](#) di ricollegarlo. Non è possibile connettere un archivio AWS CloudHSM chiavi con uno FAILED stato di connessione.

- CLUSTER_NOT_FOUND indica che AWS KMS non è possibile trovare un AWS CloudHSM cluster con l'ID cluster specificato. Il problema potrebbe essere dovuto a un ID cluster errato fornito a un'operazione API oppure all'eliminazione e alla mancata sostituzione del cluster. Per correggere questo errore, verifica l'ID del cluster, ad esempio utilizzando la AWS CloudHSM console o l'[DescribeClusters](#) operazione. Se il cluster è stato eliminato, [crea un cluster a partire da un backup recente](#) dell'originale. Quindi, [disconnetti l'archivio AWS CloudHSM chiavi](#), [modifica l'impostazione dell'ID del cluster del AWS CloudHSM key store](#) e [ricollega l'archivio AWS CloudHSM chiavi](#) al cluster.
- INSUFFICIENT_CLOUDHSM_HSMs indica che il AWS CloudHSM cluster associato non ne contiene. HSMs Per eseguire la connessione, il cluster deve avere almeno un HSM. Per trovare il numero di HSMs nel cluster, usa l'[DescribeClusters](#) operazione. Per correggere questo errore, [aggiungi](#)

[almeno un HSM](#) al cluster. Se ne aggiungi più HSMs, è meglio crearli in zone di disponibilità diverse.

- `INSUFFICIENT_FREE_ADDRESSES_IN_SUBNET` indica che non è AWS KMS stato possibile connettere l'archivio di AWS CloudHSM chiavi al relativo AWS CloudHSM cluster perché almeno una [sottorete privata associata al cluster](#) non dispone di indirizzi IP disponibili. Una connessione al AWS CloudHSM key store richiede un indirizzo IP libero in ciascuna delle sottoreti private associate, sebbene ne siano preferibili due.

[Non puoi aggiungere indirizzi IP](#) (blocchi CIDR) a una sottorete esistente. Se possibile, sposta o elimina altre risorse che utilizzano gli indirizzi IP nella sottorete, ad esempio EC2 istanze non utilizzate o interfacce di rete elastiche. [Altrimenti, è possibile creare un cluster da un backup recente del AWS CloudHSM cluster con sottoreti private nuove o esistenti che dispongono di più spazio libero per gli indirizzi](#). Quindi, per associare il nuovo cluster al tuo AWS CloudHSM key store, [disconnetti l'archivio chiavi personalizzato](#), [modifica l'ID del cluster](#) dell'archivio AWS CloudHSM chiavi con l'ID del nuovo cluster e prova a connetterti di nuovo.

 Tip

Per evitare di [reimpostare la kmsuser password](#), utilizza il backup più recente del cluster. AWS CloudHSM

- `INTERNAL_ERROR` indica che non è AWS KMS stato possibile completare la richiesta a causa di un errore interno. Riprova la richiesta. Per `ConnectCustomKeyStore` le richieste, disconnetti l'archivio delle AWS CloudHSM chiavi prima di riprovare a connetterti.
- `INVALID_CREDENTIALS` indica che AWS KMS non può accedere al AWS CloudHSM cluster associato perché non dispone della password dell'`kmsuser` account corretta. Per informazioni su questo errore, consulta [Come correggere credenziali kmsuser non valide](#).
- `NETWORK_ERRORS` indica in genere problemi di rete temporanei. [Disconnetti il AWS CloudHSM key store](#), attendi qualche minuto e riprova a connetterti.
- `SUBNET_NOT_FOUND` indica che almeno una sottorete nella configurazione del AWS CloudHSM cluster è stata eliminata. Se AWS KMS non è possibile trovare tutte le sottoreti nella configurazione del cluster, i tentativi di connettere l'archivio delle AWS CloudHSM chiavi al cluster hanno esito negativo. AWS CloudHSM

Per correggere questo errore, [crea un cluster da un backup recente](#) dello stesso AWS CloudHSM cluster. Questo processo crea una nuova configurazione del cluster con un VPC e sottoreti private. Verificare che il nuovo cluster soddisfi i [requisiti per uno store delle chiavi personalizzate](#)

e prendere nota del nuovo ID cluster. Quindi, per associare il nuovo cluster al tuo archivio AWS CloudHSM chiavi, [disconnetti l'archivio chiavi personalizzato](#), [modifica l'ID del cluster](#) dell'archivio AWS CloudHSM chiavi con l'ID del nuovo cluster e prova a connetterti di nuovo.

 Tip

Per evitare di [reimpostare la kmsuser password](#), utilizza il backup più recente del cluster. AWS CloudHSM

- USER_LOCKED_OUT indica che l'[account crypto user \(CU\) kmsuser](#) è bloccato per il cluster AWS CloudHSM a causa di troppi tentativi con password errate. Per informazioni su questo errore, consulta [Come correggere credenziali kmsuser non valide](#).

Per correggere questo errore, [scollegate l'archivio delle AWS CloudHSM chiavi e utilizzate il comando user change-password](#) nella CLI di CloudHSM per modificare la password dell'account kmsuser. Quindi, [modifica l'impostazione della password kmsuser](#) per lo store delle chiavi personalizzate ed esegui un nuovo tentativo di connessione. Per assistenza, utilizza la procedura descritta nell'argomento [Come correggere credenziali kmsuser non valide](#).

- USER_LOGGED_IN indica che l'account kmsuser CU è connesso al cluster associato. AWS CloudHSM Ciò AWS KMS impedisce di ruotare la password dell'account kmsuser e di accedere al cluster. Per correggere questo errore, registra l'utente di crittografia kmsuser fuori dal cluster. Se hai cambiato la kmsuser password per accedere al cluster, devi anche aggiornare il valore della password dell'archivio chiavi per l'archivio delle AWS CloudHSM chiavi. Per assistenza, consulta [Come scollegarsi e riconnettersi](#).
- USER_NOT_FOUND indica che AWS KMS non è possibile trovare un account kmsuser CU nel AWS CloudHSM cluster associato. Per correggere questo errore, [crea un account kmsuser CU](#) nel cluster, quindi [aggiorna il valore della password del key store](#) per l'archivio AWS CloudHSM chiavi. Per assistenza, consulta [Come correggere credenziali kmsuser non valide](#).

Come rispondere a un errore di un'operazione di crittografia

L'esito di un'operazione di crittografia che utilizza una chiave KMS in un archivio di chiavi personalizzate potrebbe essere negativo con un `KMSInvalidStateException`. L'eccezione `KMSInvalidStateException` può essere accompagnata dai seguenti messaggi di errore.

KMS non può comunicare con il tuo cluster CloudHSM. Potrebbe trattarsi di un problema di rete temporaneo. Se visualizzi ripetutamente questo errore, verifica che la rete ACLs e le regole del gruppo di sicurezza per il VPC del AWS CloudHSM cluster siano corrette.

- Sebbene si tratti di un errore HTTPS 400, potrebbe derivare da problemi di rete transitori. Per rispondere, prova a riformulare la richiesta. Tuttavia, se il problema persiste, esamina la configurazione dei componenti di rete. Questo errore è probabilmente causato dall'errata configurazione di un componente di rete, ad esempio una regola firewall o una regola di gruppo di protezione VPC che blocca il traffico in uscita. Ad esempio, KMS non può comunicare con i AWS CloudHSM cluster tramite IPv6. Per i dettagli sui prerequisiti, vedere. [Crea un archivio di AWS CloudHSM chiavi](#)

KMS non può comunicare con il AWS CloudHSM cluster perché l'utente kmsuser è bloccato. Se vedi questo errore ripetutamente, disconnetti l'archivio delle AWS CloudHSM chiavi e reimposta la password dell'account kmsuser. Aggiorna la password kmsuser per l'archivio di chiavi personalizzate e ritenta la richiesta.

- Questo messaggio di errore indica che l'[account crypto user \(CU\) kmsuser](#) è bloccato per il cluster AWS CloudHSM associato a causa del numero eccessivo di tentativi con password errate. Per informazioni su questo errore, consulta [Come disconnettersi ed eseguire l'accesso](#).

Come correggere credenziali **kmsuser** non valide

Quando [connetti un archivio di AWS CloudHSM chiavi](#), AWS KMS accede al AWS CloudHSM cluster associato come [utente kmsuser crittografico](#) (CU). Rimane connesso finché il AWS CloudHSM key store non viene disconnesso. La risposta [DescribeCustomKeyStores](#) mostra un ConnectionState di FAILED e il valore ConnectionErrorCode di INVALID_CREDENTIALS, come mostrato nell'esempio seguente.

Se si disconnette il AWS CloudHSM key store e si modifica la kmsuser password, AWS KMS non è possibile accedere al AWS CloudHSM cluster con le credenziali dell'account CU. kmsuser. Di conseguenza, tutti i tentativi di connessione al AWS CloudHSM key store falliscono. La risposta DescribeCustomKeyStores mostra un ConnectionState di FAILED e il valore ConnectionErrorCode di INVALID_CREDENTIALS, come mostrato nell'esempio seguente.


```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleKeyStore
{
  "CustomKeyStores": [
    {
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionErrorCode": "INVALID_CREDENTIALS",
      "CustomKeyId": "cks-1234567890abcdef0",
      "CustomKeyName": "ExampleKeyStore",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "FAILED"
    }
  ],
}
```

Inoltre, dopo cinque tentativi di accesso al cluster non riusciti a causa di una password errata, AWS CloudHSM blocca l'account utente. Per accedere al cluster, devi modificare la password dell'account.

Se AWS KMS riceve una risposta di blocco quando tenta di accedere al cluster come kmsuser CU, la richiesta di connessione al AWS CloudHSM key store ha esito negativo.

La [DescribeCustomKeyStores](#) risposta include un ConnectionState of FAILED e un ConnectionErrorCode valore di USER_LOCKED_OUT, come illustrato nell'esempio seguente.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleKeyStore
{
  "CustomKeyStores": [
    {
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionErrorCode": "USER_LOCKED_OUT",
      "CustomKeyId": "cks-1234567890abcdef0",
      "CustomKeyName": "ExampleKeyStore",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "FAILED"
    }
  ],
}
```

Per correggere queste condizioni, utilizza la procedura seguente.

1. [Disconnettere l'archivio delle AWS CloudHSM chiavi](#).
2. Esegui l'[DescribeCustomKeyStores](#) operazione e visualizza il valore dell'ConnectionErrorCode elemento nella risposta.

- Se `ConnectionErrorCode` è `INVALID_CREDENTIALS`, determinare la password corrente per l'account `kmsuser`. Se necessario, utilizzate il comando [user change-password](#) nella CLI di CloudHSM per impostare la password su un valore noto.
 - Se il `ConnectionErrorCode` valore è `USER_LOCKED_OUT`, è necessario utilizzare il comando [user change-password](#) nella CLI di CloudHSM per modificare la password. `kmsuser`
3. [Modificare l'impostazione della password kmsuser](#) di modo che corrisponda alla password `kmsuser` corrente nel cluster. Questa operazione indica a AWS KMS quale password utilizzare per accedere al cluster. Non modifica la password `kmsuser` nel cluster.
 4. [Connettere lo store delle chiavi personalizzate](#).

Come eliminare materiale della chiave orfano

Dopo aver pianificato l'eliminazione di una chiave KMS da un archivio di AWS CloudHSM chiavi, potrebbe essere necessario eliminare manualmente il materiale chiave corrispondente dal cluster associato. AWS CloudHSM

Quando crei una chiave KMS in un archivio AWS CloudHSM chiavi, AWS KMS crea i metadati della chiave KMS AWS KMS e genera il materiale chiave nel cluster associato. AWS CloudHSM Quando pianifichi l'eliminazione di una chiave KMS in un archivio AWS CloudHSM chiavi, dopo il periodo di attesa, AWS KMS elimina i metadati della chiave KMS. Quindi AWS KMS fa del suo meglio per eliminare il materiale chiave corrispondente dal cluster. AWS CloudHSM Il tentativo potrebbe fallire se AWS KMS non è possibile accedere al cluster, ad esempio quando viene disconnesso dall'archivio delle AWS CloudHSM chiavi o la `kmsuser` password viene modificata. AWS KMS non tenta di eliminare il materiale chiave dai backup del cluster.

AWS KMS riporta i risultati del tentativo di eliminare il materiale chiave dal cluster nell'immissione degli `DeleteKey` eventi nei AWS CloudTrail log dell'utente. Appare nell'elemento `backingKeysDeletionStatus` dell'elemento `additionalEventData`, come mostrato nella seguente voce di esempio. La voce include anche l'ARN della chiave KMS, AWS CloudHSM l'ID del cluster e l'ID `backing-key-id` () del materiale chiave.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "accountId": "111122223333",
    "invokedBy": "AWS Internal"
  },

```

```

    "eventTime": "2021-12-10T14:23:51Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": {
      "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "additionalEventData": {
      "customKeyStoreId": "cks-1234567890abcdef0",
      "clusterId": "cluster-1a23b4cdefg",
      "backingKeys": "[{\"backingKeyId\":\"backing-key-id\"}]",
      "backingKeysDeletionStatus": "[{\"backingKeyId\":\"backing-key-id\", \"deletionStatus\":\"FAILURE\"}]"]
    },
    "eventID": "c21f1f47-f52b-4ffe-bff0-6d994403cf40",
    "readOnly": false,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:eu-west-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
    "managementEvent": true,
    "eventCategory": "Management"
  }
}

```

Note

Le seguenti procedure utilizzano lo strumento da riga di comando AWS CloudHSM Client SDK 5, [CloudHSM CLI](#). La CLI di CloudhSM sostituisce con. key-handle key-reference. Il 1° gennaio 2025, AWS CloudHSM terminerà il supporto per gli strumenti da riga di comando di Client SDK 3, la CloudHSM Management Utility (CMU) e la Key Management Utility (KMU). Per ulteriori informazioni sulle differenze tra gli strumenti da riga di comando di

Client SDK 3 e lo strumento da riga di comando di Client SDK 5, consulta [Migrare da Client SDK 3 CMU e KMU a Client SDK 5 CloudHSM CLI nella Guida per l'utente](#).AWS CloudHSM

Le seguenti procedure mostrano come eliminare il materiale chiave orfano dal cluster associato. AWS CloudHSM

1. Disconnetti l'archivio delle AWS CloudHSM chiavi, se non è già disconnesso, quindi [accedi](#), come spiegato in. [Come disconnettersi ed eseguire l'accesso](#)

 Note

Quando un archivio delle chiavi personalizzate è disconnesso, tutti i tentativi di creare chiavi KMS nell'archivio delle chiavi personalizzate o di utilizzare le chiavi KMS in operazioni di crittografia avrà esito negativo. Questa azione può impedire agli utenti di archiviare e accedere ai dati sensibili.

2. Utilizza il comando [key delete](#) nella CLI di CloudhSM per eliminare la chiave HSMs dal cluster.

Tutte le voci di CloudTrail registro per le operazioni crittografiche con una chiave KMS in un archivio di AWS CloudHSM chiavi includono un `additionalEventData` campo con e. `customKeyStoreId` `backingKey` Il valore restituito nel `backingKeyId` campo è l'attributo chiave `id` CloudHSM. Ti consigliamo di filtrare l'operazione di eliminazione delle chiavi `id` per eliminare il materiale chiave orfano che hai identificato nei tuoi log. CloudTrail

AWS CloudHSM riconosce il `backingKeyId` valore come valore esadecimale. Per filtrare in base `id`, è necessario anteporre il suffisso con. `backingKeyId 0x` Ad esempio, se `backingKeyId` nel CloudTrail registro è presente `1a2b3c45678abcdef`, è necessario filtrare per. `0x1a2b3c45678abcdef`

L'esempio seguente elimina una chiave dal HSMs cluster. `backing-key-id` È elencato nella voce di CloudTrail registro. Prima di eseguire questo comando, sostituisci l'esempio `backing-key-id` con uno valido del tuo account.

```
aws-cloudhsm key delete --filter attr.id="0x<backing-key-id>"
{
  "error_code": 0,
  "data": {
    "message": "Key deleted successfully"
```

```
}  
}
```

3. Disconnettersi e ricollegare l'archivio delle AWS CloudHSM chiavi come descritto in [Come scollegarsi e riconnettersi](#).

Come recuperare il materiale chiave eliminato per una chiave KMS

Se il materiale chiave di un AWS KMS key viene eliminato, la chiave KMS è inutilizzabile e tutto il testo cifrato che è stato crittografato con la chiave KMS non può essere decrittografato. Ciò può accadere se il materiale chiave per una chiave KMS in un AWS CloudHSM archivio di chiavi viene eliminato dal cluster associato. AWS CloudHSM Potrebbe tuttavia essere possibile recuperare questo materiale.

Quando crei una AWS KMS key (chiave KMS) in un archivio AWS CloudHSM chiavi, AWS KMS accede al AWS CloudHSM cluster associato e crea il materiale chiave per la chiave KMS. Inoltre, modifica la password con un valore che solo lui conosce e rimane connesso finché l'archivio delle AWS CloudHSM chiavi è connesso. Poiché solo il proprietario della chiave, ovvero la CU che ha creato una chiave, può eliminare la chiave, è improbabile che la chiave venga eliminata accidentalmente. HSMs

Tuttavia, se il materiale chiave di una chiave KMS viene eliminato da un cluster, lo stato della chiave della chiave KMS alla fine cambia HSMs in. UNAVAILABLE Se tenti di utilizzare la chiave KMS per un'operazione di crittografia, questa ha esito negativo con un'eccezione `KMSInvalidStateException`. Cosa più importante, tutti i dati crittografati con la chiave KMS non possono essere decrittati.

In alcuni casi, è possibile recuperare il materiale della chiave [creando un cluster a partire da un backup](#) contenente tale materiale. Questa strategia funziona solo se almeno un backup è stato creato quando la chiave esisteva e prima di essere eliminata.

Utilizza la procedura seguente per recuperare il materiale della chiave.

1. Trovare un backup del cluster che contiene il materiale della chiave. Il backup deve contenere almeno tutti gli utenti e le chiavi necessari per supportare il cluster e i relativi dati crittografati.

Usa l'[DescribeBackups](#) operazione per elencare i backup per un cluster. Utilizzare quindi il timestamp del backup per la selezione di un backup. Per limitare l'output al cluster associato al AWS CloudHSM key store, utilizzate il `Filters` parametro, come illustrato nell'esempio seguente.

```
$ aws cloudhsmv2 describe-backups --filters clusterIds=<cluster ID>
{
  "Backups": [
    {
      "ClusterId": "cluster-1a23b4cdefg",
      "BackupId": "backup-9g87f6edcba",
      "CreateTimestamp": 1536667238.328,
      "BackupState": "READY"
    },
    ...
  ]
}
```

2. [Creare un cluster a partire dal backup selezionato](#). Verificare che il backup contenga la chiave eliminata e altri utenti e chiavi che il cluster richiede.
3. [Disconnetti l'archivio delle AWS CloudHSM chiavi](#) in modo da poterne modificare le proprietà.
4. [Modifica l'ID del cluster](#) del AWS CloudHSM key store. Immettere l'ID cluster del cluster creato a partire dal backup. Poiché il cluster condivide una cronologia dei backup con il cluster originale, il nuovo ID cluster deve essere valido.
5. [Ricollegare l'archivio delle AWS CloudHSM chiavi](#).

Come accedere come **kmsuser**

Per creare e gestire il materiale chiave nel AWS CloudHSM cluster per il tuo AWS CloudHSM key store, AWS KMS utilizza l'account [kmsuserCrypto User \(CU\)](#). [Crea l'account kmsuser CU](#) nel tuo cluster e fornisci la sua password AWS KMS quando crei il tuo archivio di AWS CloudHSM chiavi.

In generale, AWS KMS gestisce l'**kmsuser**account. Tuttavia, per alcune attività, è necessario disconnettere l'archivio delle AWS CloudHSM chiavi, accedere al cluster come **kmsuser CU** e utilizzare l'interfaccia a [riga di comando \(CLI\) di CloudhSM](#).

Note

Quando un archivio delle chiavi personalizzate è disconnesso, tutti i tentativi di creare chiavi KMS nell'archivio delle chiavi personalizzate o di utilizzare le chiavi KMS in operazioni di crittografia avrà esito negativo. Questa azione può impedire agli utenti di archiviare e accedere ai dati sensibili.

Questo argomento spiega come [disconnettere l'archivio AWS CloudHSM chiavi e accedere come `kmsuser`](#), eseguire lo strumento da riga di AWS CloudHSM comando, [disconnettersi e ricollegare](#) l'archivio chiavi. AWS CloudHSM

Argomenti

- [Come disconnettersi ed eseguire l'accesso](#)
- [Come scollegarsi e riconnettersi](#)

Come disconnettersi ed eseguire l'accesso

Utilizza la seguente procedura ogni volta che devi accedere a un cluster associato come utente `kmsuser` crittografico.

Note

Le seguenti procedure utilizzano lo strumento da riga di comando AWS CloudHSM Client SDK 5, [CloudHSM CLI](#). La CLI di CloudhSM sostituisce con. `key-handle` `key-reference`. Il 1° gennaio 2025, AWS CloudHSM terminerà il supporto per gli strumenti da riga di comando di Client SDK 3, la CloudHSM Management Utility (CMU) e la Key Management Utility (KMU). Per ulteriori informazioni sulle differenze tra gli strumenti da riga di comando di Client SDK 3 e lo strumento da riga di comando di Client SDK 5, consulta [Migrare da Client SDK 3 CMU e KMU a Client SDK 5 CloudHSM CLI nella Guida per l'utente AWS CloudHSM](#)

1. Disconnetti l'archivio delle chiavi, se non è già disconnesso. AWS CloudHSM Puoi usare la AWS KMS console o AWS KMS l'API.

Mentre la AWS CloudHSM chiave è connessa, AWS KMS viene effettuato l'accesso come `kmsuser`. Ciò impedisce l'accesso come `kmsuser` o la modifica della password `kmsuser`.

Ad esempio, questo comando utilizza [DisconnectCustomKeyStore](#) per disconnettere un archivio di chiavi di esempio. Sostituisci l'ID del AWS CloudHSM key store di esempio con uno valido.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

2. Usa il comando login per accedere come amministratore. Utilizza le procedure descritte nella sezione [Using CloudhSM CLI](#) della Guida per l'utente AWS CloudHSM

```
aws-cloudhsm > login --username admin --role admin
Enter password:
{
  "error_code": 0,
  "data": {
    "username": "admin",
    "role": "admin"
  }
}
```

3. Utilizza il comando [user change-password](#) nella CLI di CloudhSM per cambiare la password dell'`kmsuseraccount` con una che conosci. (AWS KMS ruota la password quando colleghi il tuo key store.) AWS CloudHSM La password deve contenere da 7 a 32 caratteri alfanumerici, rispettare la distinzione tra maiuscole e minuscole e non includere caratteri speciali.
4. Effettua il login `kmsuser` utilizzando la password che hai impostato. Per istruzioni dettagliate, consulta la sezione [Using CloudhSM](#) CLI della Guida per l'utente.AWS CloudHSM

```
aws-cloudhsm > login --username kmsuser --role crypto-user
Enter password:
{
  "error_code": 0,
  "data": {
    "username": "kmsuser",
    "role": "crypto-user"
  }
}
```

Come scollegarsi e riconnettersi

Usa la seguente procedura ogni volta che devi disconnetterti come utente `kmsuser` crittografico e ricollegare il tuo key store.

Note

Le seguenti procedure utilizzano lo strumento da riga di comando AWS CloudHSM Client SDK 5, [CloudHSM](#) CLI. La CLI di CloudhSM sostituisce con. `key-handle` `key-reference`. Il 1° gennaio 2025, AWS CloudHSM terminerà il supporto per gli strumenti da riga di comando di Client SDK 3, la CloudHSM Management Utility (CMU) e la Key Management Utility (KMU). Per ulteriori informazioni sulle differenze tra gli strumenti da riga di comando di

Client SDK 3 e lo strumento da riga di comando di Client SDK 5, consulta [Migrare da Client SDK 3 CMU e KMU a Client SDK 5 CloudHSM CLI nella Guida per l'utente](#).AWS CloudHSM

1. Esegui l'operazione, quindi utilizza il comando [logout](#) nella CLI di CloudHSM per disconnetterti. Se non ti disconnetti, i tentativi di ricollegare il tuo AWS CloudHSM key store falliranno.

```
aws-cloudhsm  logout
{
  "error_code": 0,
  "data": "Logout successful"
}
```

2. [Modificare la password kmsuser](#) per lo store delle chiavi personalizzate.

Indica AWS KMS la password corrente per kmsuser il cluster. Se si omette questo passaggio, non AWS KMS sarà possibile accedere al cluster e tutti i tentativi di riconnessione dell'archivio chiavi personalizzato falliranno. kmsuser È possibile utilizzare la AWS KMS console o il KeyStorePassword parametro dell'[UpdateCustomKeyStore](#) operazione.

Ad esempio, questo comando indica AWS KMS che la password corrente è tempPassword. Sostituire la password di esempio con una effettiva.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --key-store-password tempPassword
```

3. Ricollegare l'archivio AWS KMS chiavi al relativo AWS CloudHSM cluster. Sostituisci l'ID del AWS CloudHSM key store di esempio con uno valido. Durante il processo di connessione, AWS KMS modifica la kmsuser password con un valore che solo lei conosce.

L'[ConnectCustomKeyStore](#) operazione viene ripristinata rapidamente, ma il processo di connessione può richiedere un periodo di tempo prolungato. La risposta iniziale non indica se il processo di connessione è riuscito.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

4. Utilizzare l'[DescribeCustomKeyStores](#) operazione per verificare che l'archivio delle AWS CloudHSM chiavi sia connesso. Sostituisci l'ID del AWS CloudHSM key store di esempio con uno valido.

In questo esempio, il campo dello stato della connessione mostra che il AWS CloudHSM key store è ora connesso.

```
$ aws kms describe-custom-key-stores --custom-key-store-  
id cks-1234567890abcdef0  
{  
  "CustomKeyStores": [  
    "CustomKeyId": "cks-1234567890abcdef0",  
    "CustomKeyName": "ExampleKeyStore",  
    "CloudHsmClusterId": "cluster-1a23b4cdefg",  
    "TrustAnchorCertificate": "<certificate string appears here>",  
    "CreationDate": "1.499288695918E9",  
    "ConnectionState": "CONNECTED"  
  ],  
}
```

Archivi delle chiavi esterne

Gli archivi di chiavi esterni consentono di proteggere le AWS risorse utilizzando chiavi crittografiche esterne a. AWS Questa funzionalità avanzata è progettata per carichi di lavoro regolamentati che è necessario proteggere con chiavi crittografiche archiviate in un sistema di gestione delle chiavi esterno da te controllato. Gli archivi di chiavi esterni supportano l'[impegno di sovranitàAWS digitale](#) per darti il controllo sovrano sui tuoi dati AWS, inclusa la possibilità di cifrarli con materiale chiave che possiedi e che controlli all'esterno. AWS

Un archivio di chiavi esterno è un archivio di [chiavi personalizzato](#) supportato da un gestore di chiavi esterno che possiedi e gestisci esternamente. AWS Il gestore di chiavi esterno può essere un modulo di sicurezza hardware fisico o virtuale (HSMs) o qualsiasi sistema basato su hardware o software in grado di generare e utilizzare chiavi crittografiche. Le operazioni di crittografia e decrittografia che utilizzano una chiave KMS in un archivio di chiavi esterno vengono eseguite dal gestore delle chiavi esterno utilizzando il materiale relativo alla chiave crittografica, una funzionalità nota come hold your own keys (). HYOKs

AWS KMS non interagisce mai direttamente con il gestore delle chiavi esterno e non può creare, visualizzare, gestire o eliminare le chiavi. AWS KMS Interagisce invece solo con il software proxy di [archiviazione delle chiavi esterno \(proxy XKS\)](#) fornito dall'utente. Il proxy dell'archivio chiavi esterno media tutte le comunicazioni tra AWS KMS e il gestore delle chiavi esterno. Trasmette tutte le richieste AWS KMS al gestore delle chiavi esterno e ritrasmette le risposte dal gestore delle chiavi

esterno a. AWS KMS Il proxy dell'archivio chiavi esterno traduce anche le richieste generiche AWS KMS in un formato specifico del fornitore comprensibile al gestore delle chiavi esterno, consentendoti di utilizzare archivi di chiavi esterni con gestori di chiavi di diversi fornitori.

Puoi utilizzare le chiavi KMS in un archivio delle chiavi esterne per la crittografia lato client, tra cui [AWS Encryption SDK](#). Tuttavia, gli archivi di chiavi esterni sono una risorsa importante per la crittografia lato server, poiché consentono di proteggere più AWS Servizi AWS risorse utilizzando chiavi crittografiche esterne. AWS Servizi AWS che supportano [le chiavi gestite dal cliente](#) per la crittografia simmetrica supportano anche le chiavi KMS in un archivio di chiavi esterno. Per i dettagli sul supporto del servizio, consulta la sezione [Integrazione del servizio AWS](#).

Gli archivi di chiavi esterni ne consentono l'utilizzo AWS KMS per carichi di lavoro regolamentati in cui le chiavi di crittografia devono essere archiviate e utilizzate all'esterno di. AWS Tuttavia, si discostano notevolmente dal modello standard di responsabilità condivisa e richiedono oneri operativi aggiuntivi. Il rischio maggiore in termini di disponibilità e latenza supererà, per la maggior parte dei clienti, i vantaggi di sicurezza percepiti per gli archivi delle chiavi esterne.

Gli archivi delle chiavi esterne ti consentono di controllare la radice di attendibilità. I dati crittografati con le chiavi KMS dell'archivio delle chiavi esterne possono essere decrittografati solo utilizzando il gestore delle chiavi esterne che è sotto tuo controllo. Se si revoca temporaneamente l'accesso al gestore di chiavi esterno, ad esempio disconnettendo l'archivio chiavi esterno o disconnettendo il gestore di chiavi esterno dal proxy dell'archivio chiavi esterno, AWS perde ogni accesso alle chiavi crittografiche finché non le si ripristina. Durante tale intervallo di tempo, il testo criptato con le chiavi KMS non può essere decifrato. Se revochi definitivamente l'accesso al gestore delle chiavi esterne, tutto il testo criptato con una chiave KMS nell'archivio delle chiavi esterne diventa irrecuperabile. Le uniche eccezioni sono i AWS servizi che memorizzano brevemente nella cache le [chiavi dati protette dalle chiavi KMS](#). Queste chiavi dati continuano a funzionare fino alla disattivazione della risorsa o alla scadenza della cache. Per informazioni dettagliate, vedi [In che modo le chiavi KMS inutilizzabili influiscono sulle chiavi dati](#).

Gli archivi di chiavi esterni sbloccano i pochi casi d'uso per carichi di lavoro regolamentati in cui le chiavi di crittografia devono rimanere esclusivamente sotto il tuo controllo e inaccessibili. AWS Ciò rappresenta un cambiamento importante nel modo in cui gestisci l'infrastruttura basata sul cloud e una modifica sostanziale nel modello di responsabilità condivisa. Per la maggior parte dei carichi di lavoro, gli oneri operativi aggiuntivi e i maggiori rischi associati alla disponibilità e alle prestazioni superano i vantaggi di sicurezza percepiti per gli archivi delle chiavi esterne.

È necessario un archivio delle chiavi esterne?

Per la maggior parte degli utenti, l'archivio AWS KMS chiavi predefinito, protetto da [moduli di sicurezza hardware di livello 3 convalidati FIPS 140-3, soddisfa i requisiti di sicurezza](#), controllo e normativi. Gli utenti dell'archivio delle chiavi esterne devono sostenere costi elevati, oneri di manutenzione e risoluzione dei problemi, nonché rischi per latenza, disponibilità e affidabilità.

Quando prendi in considerazione un archivio di chiavi esterno, dedica del tempo alla comprensione delle alternative, tra cui un [archivio AWS CloudHSM chiavi](#) supportato da un AWS CloudHSM cluster di tua proprietà e gestione e le chiavi KMS con [materiale chiave importato](#) che generi internamente HSMS e che puoi eliminare dalle chiavi KMS su richiesta. In particolare, l'importazione del materiale della chiave con un intervallo di scadenza molto breve potrebbe fornire un livello di controllo simile senza comportare rischi in termini di prestazioni o disponibilità.

Un archivio delle chiavi esterne potrebbe essere la soluzione ideale per la tua organizzazione se disponi dei requisiti seguenti:

- Ti viene richiesto di utilizzare chiavi crittografiche nel tuo gestore di chiavi locale o in un gestore di chiavi al di fuori del AWS tuo controllo.
- Devi dimostrare che le chiavi crittografiche vengono conservate esclusivamente sotto il tuo controllo al di fuori del cloud.
- Devi crittografare e decrittografare tramite chiavi crittografiche con autorizzazione indipendente.
- Il materiale chiave deve essere sottoposto a un percorso di audit secondario e indipendente.

Se scegli un archivio delle chiavi esterne, limita il suo utilizzo ai carichi di lavoro che richiedono protezione con chiavi crittografiche al di fuori di AWS.

Modello di responsabilità condivisa

Le chiavi KMS standard utilizzano materiale chiave generato e utilizzato in un ambiente di AWS KMS proprietà e HSMS gestione. Stabilisci le politiche di controllo degli accessi sulle tue chiavi KMS e configurale Servizi AWS che utilizzano le chiavi KMS per proteggere le tue risorse. AWS KMS si assume la responsabilità della sicurezza, della disponibilità, della latenza e della durabilità del materiale chiave contenuto nelle chiavi KMS.

Le chiavi KMS negli archivi delle chiavi esterne si basano sul materiale della chiave e sulle operazioni del gestore delle chiavi esterne. Pertanto, l'equilibrio delle responsabilità si sposta a tuo carico. Sei

responsabile della sicurezza, dell'affidabilità, della durata e delle prestazioni delle chiavi crittografiche nel tuo gestore di chiavi esterno. AWS KMS è responsabile della risposta tempestiva alle richieste e della comunicazione con il proxy di archiviazione delle chiavi esterno, nonché del mantenimento dei nostri standard di sicurezza. [Per garantire che ogni chiave esterna memorizzi testo cifrato almeno quanto il testo AWS KMS cifrato standard, AWS KMS prima crittografa tutto il testo in chiaro con il materiale AWS KMS chiave specifico della chiave KMS, quindi lo invia al gestore delle chiavi esterno per la crittografia con la chiave esterna, una procedura nota come doppia crittografia.](#) Di conseguenza, né AWS KMS né il proprietario del materiale della chiave esterna possono decrittografare da soli il testo criptato con doppia crittografia.

Sei responsabile del mantenimento di un gestore di chiavi esterno che soddisfi i tuoi standard normativi e prestazionali, della fornitura e della manutenzione di un proxy dell'archivio di chiavi esterno che sia conforme alla [specifica API del proxy dell'archivio di chiavi esterno AWS KMS](#) e della garanzia di durabilità e disponibilità del materiale della tua chiave. Devi inoltre creare, configurare e gestire un archivio delle chiavi esterne. Quando si verificano errori causati da componenti gestiti, è necessario essere pronti a identificarli e risolverli in modo che AWS i servizi possano accedere alle risorse senza interruzioni indebite. AWS KMS fornisce [indicazioni sulla risoluzione dei problemi](#) per aiutarvi a determinare la causa dei problemi e le soluzioni più probabili.

Esamina le [CloudWatch metriche e le dimensioni di Amazon](#) registrate per gli AWS KMS archivi di chiavi esterni. AWS KMS consiglia vivamente di creare CloudWatch allarmi per monitorare l'archivio di chiavi esterno in modo da poter rilevare i primi segnali di problemi prestazionali e operativi prima che si verifichino.

Cosa sta cambiando?

Gli archivi delle chiavi esterne supportano solo chiavi KMS di crittografia simmetrica. All'interno AWS KMS, si utilizzano e gestiscono le chiavi KMS in un archivio di chiavi esterno più o meno allo stesso modo in cui si gestiscono le altre [chiavi gestite dai clienti](#), inclusa [l'impostazione delle politiche di controllo degli accessi](#) e il [monitoraggio dell'uso delle chiavi](#). Utilizzi lo stesso APIs con gli stessi parametri per richiedere un'operazione crittografica con una chiave KMS in un archivio di chiavi esterno che utilizzi per qualsiasi chiave KMS. Anche i prezzi sono gli stessi delle chiavi KMS standard. [Per i dettagli, consulta la sezione PrezziChiavi KMS negli archivi di chiavi esterni.AWS Key Management Service](#)

Tuttavia, con gli archivi delle chiavi esterne cambiano i seguenti principi:

- Sei responsabile della disponibilità, della durata e della latenza delle operazioni con le chiavi.

- Sei responsabile di tutti i costi per lo sviluppo, l'acquisto, il funzionamento e la concessione di licenze per il sistema di gestione delle chiavi esterne.
- Puoi implementare [l'autorizzazione indipendente](#) di tutte le richieste inviate AWS KMS al tuo proxy di archiviazione delle chiavi esterno.
- È possibile monitorare, controllare e registrare tutte le operazioni del proxy dell'archivio chiavi esterno e tutte le operazioni del gestore di chiavi esterno relative alle AWS KMS richieste.

Da dove iniziare?

Per creare e gestire un archivio delle chiavi esterne, è necessario [scegliere l'opzione di connettività proxy dell'archivio delle chiavi esterne](#), [assemblare i prerequisiti](#) e infine [creare e configurare l'archivio delle chiavi esterne](#).

Quote

AWS KMS consente fino a [10 archivi di chiavi personalizzati](#) in ciascuna Account AWS regione, inclusi archivi di [AWS CloudHSM chiavi e archivi](#) di [chiavi esterni](#), indipendentemente dallo stato della connessione. Inoltre, sono previste quote di richieste AWS KMS sull'[uso delle chiavi KMS in un archivio delle chiavi esterno](#).

Se scegli la [connettività proxy VPC](#) per il tuo proxy di archiviazione chiavi esterno, potrebbero esserci anche delle quote sui componenti richiesti, ad esempio sottoreti e sistemi di VPCs bilanciamento del carico di rete. Per ulteriori informazioni su queste quote, utilizza la [console Service Quotas](#).

Regioni

Per ridurre al minimo la latenza di rete, crea i componenti dell'archivio delle chiavi esterne nella Regione AWS più vicina al [gestore delle chiavi esterne](#). Se possibile, scegli una regione con un tempo di andata e ritorno (RTT) della rete di 35 millisecondi o meno.

Gli archivi di chiavi esterni sono supportati in tutti i Regioni AWS paesi in cui AWS KMS è supportato ad eccezione della Cina (Pechino) e della Cina (Ningxia).

Caratteristiche non supportate

AWS KMS non supporta le seguenti funzionalità negli archivi di chiavi personalizzati.

- [Chiavi KMS asimmetriche](#)
- [Chiavi KMS HMAC](#)
- [Chiavi KMS con materiale della chiave importato](#)
- [Rotazione automatica delle chiavi](#)
- [Chiavi multi-regione](#)

Ulteriori informazioni:

- [Annuncio degli archivi delle chiavi esterne di AWS KMS](#) nel Blog AWS News.

Concetti fondamentali sull'archivio delle chiavi esterne

Scopri i termini e i concetti di base utilizzati negli archivi di chiavi esterni.

Archivio delle chiavi esterne

Un archivio chiavi esterno è un [archivio di chiavi AWS KMS personalizzato](#) supportato da un gestore di chiavi esterno AWS di tua proprietà e gestione. Ogni chiave KMS in un archivio delle chiavi esterne è associata a una [chiave esterna](#) nel gestore delle chiavi esterne. Quando utilizzi una chiave KMS in un archivio delle chiavi esterne per la crittografia o la decrittografia, l'operazione viene eseguita nel gestore delle chiavi esterne utilizzando la chiave esterna, una funzionalità nota come HYOK (Hold Your Own Keys). Questa funzionalità è progettata per le organizzazioni che devono mantenere le chiavi crittografiche nel proprio gestore delle chiavi esterne.

Gli archivi di chiavi esterni assicurano che le chiavi crittografiche e le operazioni che proteggono le AWS risorse rimangano sotto il controllo del gestore delle chiavi esterno. AWS KMS invia richieste al gestore di chiavi esterno per crittografare e decrittografare i dati, ma AWS KMS non può creare, eliminare o gestire chiavi esterne. Tutte le richieste inviate AWS KMS al gestore di chiavi esterno sono mediate da un componente software [proxy di archiviazione chiavi esterno](#) fornito, posseduto e gestito dall'utente.

AWS i servizi che supportano [le chiavi gestite dai AWS KMS clienti](#) possono utilizzare le chiavi KMS presenti nell'archivio di chiavi esterno per proteggere i dati. Di conseguenza, i tuoi dati sono infine protetti dalle chiavi utilizzando le operazioni di crittografia nel gestore delle chiavi esterne.

Le chiavi KMS in un archivio delle chiavi esterne presentano modelli di attendibilità, [accordi di responsabilità condivisa](#) e aspettative prestazionali fondamentalmente diversi rispetto alle chiavi

KMS standard. Con gli archivi delle chiavi esterne, sei responsabile della sicurezza e dell'integrità del materiale della chiave e delle operazioni di crittografia. La disponibilità e la latenza delle chiavi KMS in un archivio delle chiavi esterne sono influenzate dall'hardware, dal software, dai componenti di rete e dalla distanza tra AWS KMS e il gestore delle chiavi esterne. È inoltre probabile che vengano sostenuti costi aggiuntivi per il gestore delle chiavi esterno e per l'infrastruttura di rete e bilanciamento del carico necessaria per comunicare con il gestore delle chiavi esterno AWS KMS.

Puoi utilizzare l'archivio delle chiavi esterne come parte di una strategia di protezione dei dati più ampia. Per ogni AWS risorsa da proteggere, puoi decidere quali richiedono una chiave KMS in un archivio chiavi esterno e quali possono essere protette da una chiave KMS standard. Ciò ti offre la flessibilità di scegliere le chiavi KMS per classificazioni di dati, applicazioni o progetti specifici.

Gestore delle chiavi esterne

Un gestore delle chiavi esterne è un componente al di fuori di AWS che può generare chiavi simmetriche AES a 256 bit ed eseguire operazioni di crittografia e decrittografia simmetriche. Il gestore delle chiavi esterne per un relativo archivio può essere costituito da un modulo di sicurezza hardware (HSM) fisico, un HSM virtuale o un gestore delle chiavi software con o senza un componente HSM. Può essere posizionato ovunque all'esterno AWS, anche in sede, in un data center locale o remoto o in qualsiasi cloud. L'archivio delle chiavi esterne può essere supportato da un singolo gestore delle chiavi esterne o da più istanze di gestione delle chiavi correlate che condividono chiavi crittografiche, ad esempio un cluster HSM. Gli archivi delle chiavi esterne sono progettati per supportare una varietà di gestori esterni di diversi fornitori. Per informazioni dettagliate sulla connessione al gestore di chiavi esterno, consulta [Scegli un'opzione di connettività proxy per l'archivio chiavi esterno](#).

Chiave esterna

Ogni chiave KMS in un archivio delle chiavi esterne è associata a una chiave crittografica, nota come chiave esterna, nel [gestore delle chiavi esterne](#). Quando si esegue la crittografia o la decrittografia con una chiave KMS nell'archivio delle chiavi esterne, l'operazione di crittografia viene eseguita nel [gestore delle chiavi esterne](#) tramite la chiave esterna.

Warning

La chiave esterna è essenziale per il funzionamento della chiave KMS. Se la chiave esterna viene persa o eliminata, il testo criptato che è stato crittografato con la chiave KMS associata non è recuperabile.

Per gli archivi delle chiavi esterne, una chiave esterna deve essere una chiave AES a 256 bit abilitata per e in grado di eseguire le operazioni di crittografia e decrittografia. Per maggiori dettagli sui requisiti della chiave esterna, consulta [Requisiti per una chiave KMS in un archivio delle chiavi esterne](#).

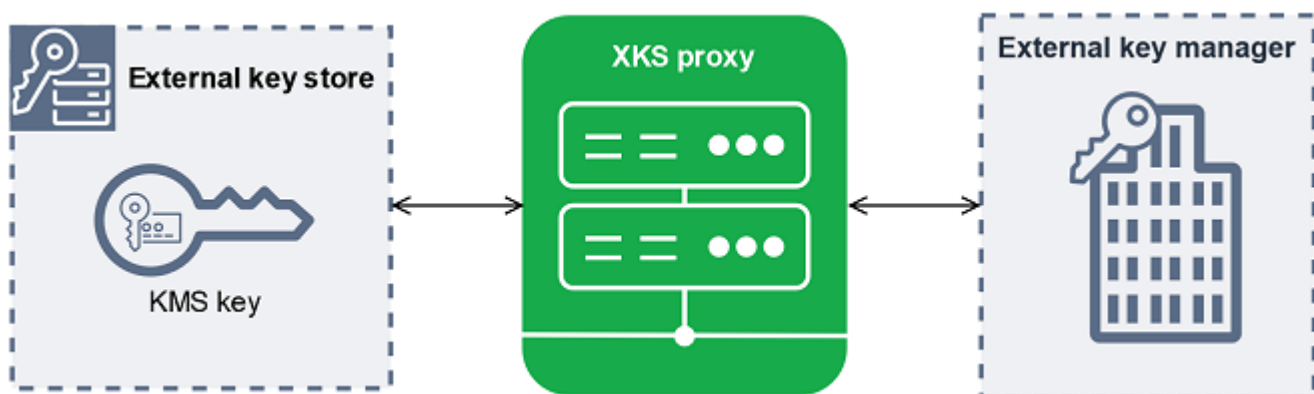
AWS KMS non può creare, eliminare o gestire chiavi esterne. Il materiale della chiave crittografica resta sempre all'interno del gestore delle chiavi esterne. Quando crei una chiave KMS in un archivio delle chiavi esterne, sei tu a fornire l'ID di una chiave esterna (XksKeyId). Non puoi modificare l'ID della chiave esterna associato a una chiave KMS, sebbene il gestore delle chiavi esterne possa ruotare il materiale della chiave associato a tale ID.

Oltre alla chiave esterna, una chiave KMS in un archivio delle chiavi esterne contiene anche il materiale della chiave AWS KMS. I dati protetti dalla chiave KMS vengono crittografati prima AWS KMS utilizzando il materiale AWS KMS chiave, quindi dal gestore delle chiavi esterno utilizzando la chiave esterna. Questo processo di [doppia crittografia](#) garantisce che il testo criptato protetto dalla chiave KMS sia sempre perlomeno altrettanto sicuro che il testo criptato protetto solo da AWS KMS.

Molte chiavi crittografiche presentano diversi tipi di identificatori. Quando crei una chiave KMS in un archivio delle chiavi esterne, fornisci l'ID della chiave esterna che il [proxy dell'archivio delle chiavi esterne](#) utilizza per fare riferimento alla chiave esterna. Se utilizzi l'identificatore sbagliato, il tentativo di creare una chiave KMS nell'archivio delle chiavi esterne ha esito negativo.

Proxy dell'archivio delle chiavi esterne

Il proxy di archiviazione delle chiavi esterno («proxy XKS») è un'applicazione software di proprietà e gestita dal cliente che media tutte le comunicazioni tra e il gestore delle chiavi esterno. AWS KMS Inoltre, traduce le AWS KMS richieste generiche in un formato comprensibile al gestore di chiavi esterno specifico del fornitore. Per un archivio delle chiavi esterne è necessario un relativo proxy. Ogni archivio delle chiavi esterne è associato a un relativo proxy.



AWS KMS non può creare, eliminare o gestire chiavi esterne. Il materiale delle chiavi crittografiche resta sempre all'interno del gestore delle chiavi. Tutte le comunicazioni tra AWS KMS e il gestore delle chiavi esterno sono mediate dal proxy dell'archivio chiavi esterno. AWS KMS invia richieste al proxy dell'archivio chiavi esterno e riceve risposte dal proxy dell'archivio chiavi esterno. Il proxy dell'archivio chiavi esterno è responsabile della trasmissione delle richieste dal AWS KMS gestore delle chiavi esterno e della trasmissione delle risposte dal gestore delle chiavi esterno a AWS KMS.

Tieni presente che, in quanto proprietario e gestore del proxy dell'archivio delle chiavi esterne, sei responsabile della manutenzione e del funzionamento. È possibile sviluppare un proxy di archiviazione chiavi esterno in base alla [specificazione dell'API proxy dell'archivio chiavi esterno](#) open source che AWS KMS pubblica o acquista un'applicazione proxy da un fornitore. Il proxy dell'archivio delle chiavi esterne potrebbe essere incluso nel gestore delle chiavi esterne. Per supportare lo sviluppo del proxy, fornisce AWS KMS anche un esempio di key store proxy esterno ([aws-kms-xks-proxy](#)) e un client di test ([xks-kms-xksproxy-test-client](#)) che verifica che il proxy dell'archivio chiavi esterno sia conforme alle specifiche.

Per l'autenticazione AWS KMS, il proxy utilizza certificati TLS lato server. [Per autenticarsi sul proxy, AWS KMS firma tutte le richieste sul proxy di archiviazione delle chiavi esterno con una credenziale di autenticazione proxy SigV4](#). Facoltativamente, il proxy può abilitare il protocollo TLS reciproco (MTLS) per garantire ulteriormente che accetti solo richieste da AWS KMS.

Il proxy dell'archivio di chiavi esterno deve supportare HTTP/1.1 o versione successiva e TLS 1.2 o versione successiva con almeno una delle seguenti suite di crittografia:

- TLS_AES_256_GCM_ (TLS 1.3SHA384)
- TLS_0_05_ CHACHA2 (POLY13SHA256 TLS 1.3)

 Note

AWS GovCloud (US) Region Non supporta TLS_0_05_ CHACHA2 POLY13 SHA256

- TLS_ECDHE_RSA_WITH_AES_256_GCM_ (TLS 1.2SHA384)
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_ (TLS 1.2SHA384)

Per creare e utilizzare le chiavi KMS nell'archivio delle chiavi esterne, devi innanzitutto [connettere l'archivio delle chiavi esterne](#) al relativo proxy. Puoi anche disconnettere l'archivio delle chiavi esterne dal relativo proxy su richiesta. Quando esegui questa operazione, tutte le chiavi KMS nell'archivio

delle chiavi esterne diventano [non disponibili](#), di conseguenza non possono essere utilizzate in alcuna operazione di crittografia.

Connettività proxy dell'archivio delle chiavi esterne

La connettività proxy dell'archivio chiavi esterno («connettività proxy XKS») descrive il metodo utilizzato per comunicare con il proxy dell'archivio chiavi esterno. AWS KMS

Puoi specificare l'opzione di connettività proxy durante la creazione dell'archivio delle chiavi esterne, rendendola così una proprietà di tale archivio. Puoi modificare l'opzione di connettività proxy aggiornando la proprietà dell'archivio delle chiavi personalizzate, tuttavia devi accertarti che il proxy dell'archivio delle chiavi esterne possa comunque accedere alle stesse chiavi esterne.

AWS KMS supporta le seguenti opzioni di connettività.

- [Connettività pubblica degli endpoint](#): AWS KMS invia le richieste per il proxy dell'archivio di chiavi esterno tramite Internet a un endpoint pubblico controllato dall'utente. Questa opzione è semplice da creare e gestire, ma potrebbe non soddisfare i requisiti di sicurezza per ogni installazione.
- [Connettività del servizio endpoint VPC](#): AWS KMS invia le richieste a un servizio endpoint Amazon Virtual Private Cloud (Amazon VPC) creato e gestito da te. Puoi ospitare il proxy dell'archivio chiavi esterno all'interno di Amazon VPC o ospitare il proxy dell'archivio chiavi esterno all'esterno AWS e utilizzare Amazon VPC solo per la comunicazione. Puoi anche connettere il tuo archivio di chiavi esterno a un servizio endpoint Amazon VPC di proprietà di un altro. Account AWS

Per informazioni dettagliate sulle opzioni di connettività proxy dell'archivio delle chiavi esterne, consulta [Scegli un'opzione di connettività proxy per l'archivio chiavi esterno](#).

Credenziali di autenticazione al proxy dell'archivio delle chiavi esterne

Per autenticarti sul tuo proxy di archivio chiavi esterno, AWS KMS firma tutte le richieste al tuo proxy di archivio chiavi esterno con una credenziale di autenticazione [Signature V4 \(SigV4\)](#). Stabilisci e gestisci la credenziale di autenticazione sul tuo proxy, quindi fornisci questa credenziale quando crei l'archivio esterno. AWS KMS

Note

La credenziale SigV4 AWS KMS utilizzata per firmare le richieste al proxy XKS non è correlata alle credenziali SigV4 associate ai principali presenti nel tuo. AWS Identity and

Access Management Account AWS Non riutilizzare le credenziali SigV4 IAM per il proxy dell'archivio delle chiavi esterne.

Ogni credenziale di autenticazione proxy è costituita da due parti. Devi fornire entrambe le parti durante la creazione di un archivio delle chiavi esterne o l'aggiornamento delle credenziali di autenticazione per l'archivio delle chiavi esterne.

- ID chiave di accesso: identifica la chiave di accesso segreta. Puoi fornire questo ID come un testo non crittografato.
- Chiave di accesso segreta: la parte segreta della credenziale. AWS KMS crittografa la chiave di accesso segreta nella credenziale prima di archivarla.

Puoi [modificare l'impostazione delle credenziali](#) in qualsiasi momento, ad esempio quando inserisci valori errati, quando modifichi le credenziali nel proxy o quando il proxy esegue la rotazione delle credenziali. Per dettagli tecnici sull' AWS KMS autenticazione al proxy dell'archivio chiavi esterno, vedere [Authentication](#) in the AWS KMS External Key Store Proxy Specification.

Per consentirti di ruotare le tue credenziali senza interrompere i Servizi AWS utilizzo delle chiavi KMS nell'archivio di chiavi esterno, consigliamo che il proxy dell'archivio chiavi esterno supporti almeno due credenziali di autenticazione valide per. AWS KMS Ciò garantisce che le credenziali precedenti continuino a funzionare mentre fornisci le nuove credenziali a AWS KMS.

Per aiutarti a tenere traccia dell'età delle tue credenziali di autenticazione proxy, AWS KMS definisce una CloudWatch metrica Amazon, [XksProxyCredentialAge](#). Puoi utilizzare questa metrica per creare un CloudWatch allarme che ti avvisa quando l'età delle tue credenziali raggiunge una soglia da te stabilita.

Per garantire inoltre che il proxy dell'archivio delle chiavi esterne risponda solo a AWS KMS, alcuni proxy supportano l'autenticazione TLS reciproca. Per informazioni dettagliate, vedi [Autenticazione TLS reciproca \(facoltativa\)](#).

Proxy APIs

Per supportare un archivio di chiavi AWS KMS esterno, un [proxy di archiviazione chiavi esterno](#) deve implementare il proxy richiesto APIs come descritto nella [specifica dell'API proxy AWS KMS External Key Store](#). Queste richieste API proxy sono le uniche richieste AWS KMS inviate al proxy. Sebbene non si inviino mai direttamente queste richieste, conoscerle potrebbe aiutarti a risolvere eventuali

problemi che potrebbero verificarsi con l'archivio delle chiavi esterne o il relativo proxy. Ad esempio, AWS KMS include informazioni sulla latenza e le percentuali di successo di queste chiamate API nelle [CloudWatch metriche Amazon](#) per gli archivi di chiavi esterni. Per informazioni dettagliate, vedi [Monitora gli archivi di chiavi esterni](#).

La tabella seguente elenca e descrive ogni proxy. APIs Include inoltre le AWS KMS operazioni che attivano una chiamata all'API proxy e tutte le eccezioni AWS KMS operative relative all'API proxy.

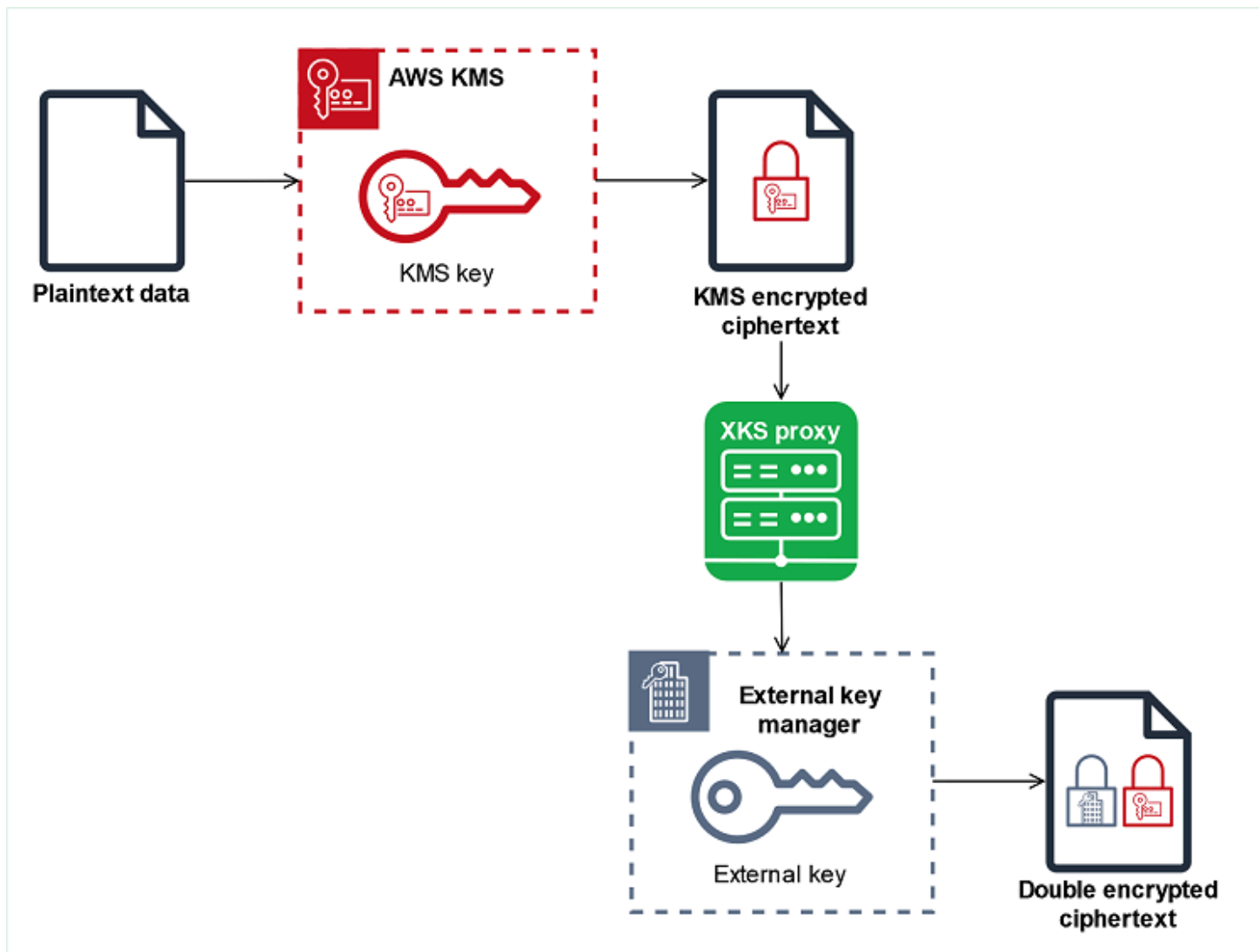
API proxy	Description	Operazioni correlate AWS KMS
Decrittografia	AWS KMS invia il testo cifrato da decrittografare e l'ID della chiave esterna da utilizzare. L'algoritmo di crittografia richiesto è AES_GCM.	Decrittografia , ReEncrypt
Crittografia	AWS KMS invia i dati da crittografare e l'ID della chiave esterna da utilizzare. L'algoritmo di crittografia richiesto è AES_GCM.	Crittografia , GenerateDataKey , GenerateDataKeyWithoutPlainText , ReEncrypt
GetHealth Status	AWS KMS richiede informazioni sullo stato del proxy e sul gestore delle chiavi esterno. Lo stato di ogni gestore delle chiavi esterne può essere uno dei seguenti. • Active: integro, può servire il traffico • Degraded: non integro, ma può servire il traffico • Unavailable : non integro, non può servire il traffico	CreateCustomKeyStore(per la connettività degli endpoint pubblici), ConnectCustomKeyStore(per la connettività del servizio endpoint VPC) Se tutte le istanze del gestore delle chiavi esterne sono Unavailable, i tentativi di creare o connettere e l'archivio delle chiavi hanno esito negativo con l'eccezione XksProxyUriUnreachableException.
GetKeyMetadata	AWS KMS richiede informazioni sulla chiave esterna associata a una chiave KMS nel tuo archivio di chiavi esterno.	CreateKey Se la specifica della chiave non è AES_256, se l'utilizzo della chiave non è [ENCRYPT, DECRYPT] o

API proxy	Description	Operazioni correlate AWS KMS
	La risposta include le specifiche della chiave (AES_256), il suo utilizzo ([ENCRYPT, DECRYPT]) e se la chiave esterna è ENABLED o DISABLED.	lo stato è DISABLED, l'operazione CreateKey ha esito negativo con l'eccezione XksKeyInvalidConfigurationException .

Doppia crittografia

I dati crittografati con una chiave KMS in un archivio delle chiavi esterne vengono crittografati due volte. Innanzitutto, AWS KMS crittografa i dati con materiale AWS KMS chiave specifico per la chiave KMS. Quindi, il testo criptato con AWS KMS viene crittografato dal [gestore delle chiavi esterne](#) utilizzando la [chiave esterna](#). Questo processo è noto come doppia crittografia.

La doppia crittografia garantisce che i dati crittografati da una chiave KMS in un archivio delle chiavi esterne siano almeno altrettanto sicuri del testo criptato crittografato da una chiave KMS standard. Protegge inoltre il testo in chiaro in transito dal proxy dell' AWS KMS archivio chiavi esterno. Con la doppia crittografia, mantieni il pieno controllo dei tuoi testi criptati. Se revochi definitivamente l'accesso AWS alla chiave esterna tramite il proxy esterno, qualsiasi testo criptato rimasto in AWS viene effettivamente eliminato in modo crittografato.



Per abilitare la doppia crittografia, ogni chiave KMS in un archivio delle chiavi esterne dispone di due materiali della chiave crittografica:

- Un materiale AWS KMS chiave esclusivo della chiave KMS. Questo materiale chiave viene generato e utilizzato solo nei moduli di sicurezza hardware certificati AWS KMS [FIPS 140-3 Security Level 3](#) (). HSMs
- Una [chiave esterna](#) nel gestore delle chiavi esterne.

La doppia crittografia ha i seguenti effetti:

- AWS KMS non può decrittografare alcun testo cifrato crittografato da una chiave KMS in un archivio di chiavi esterno senza accedere alle chiavi esterne tramite il proxy dell'archivio chiavi esterno.

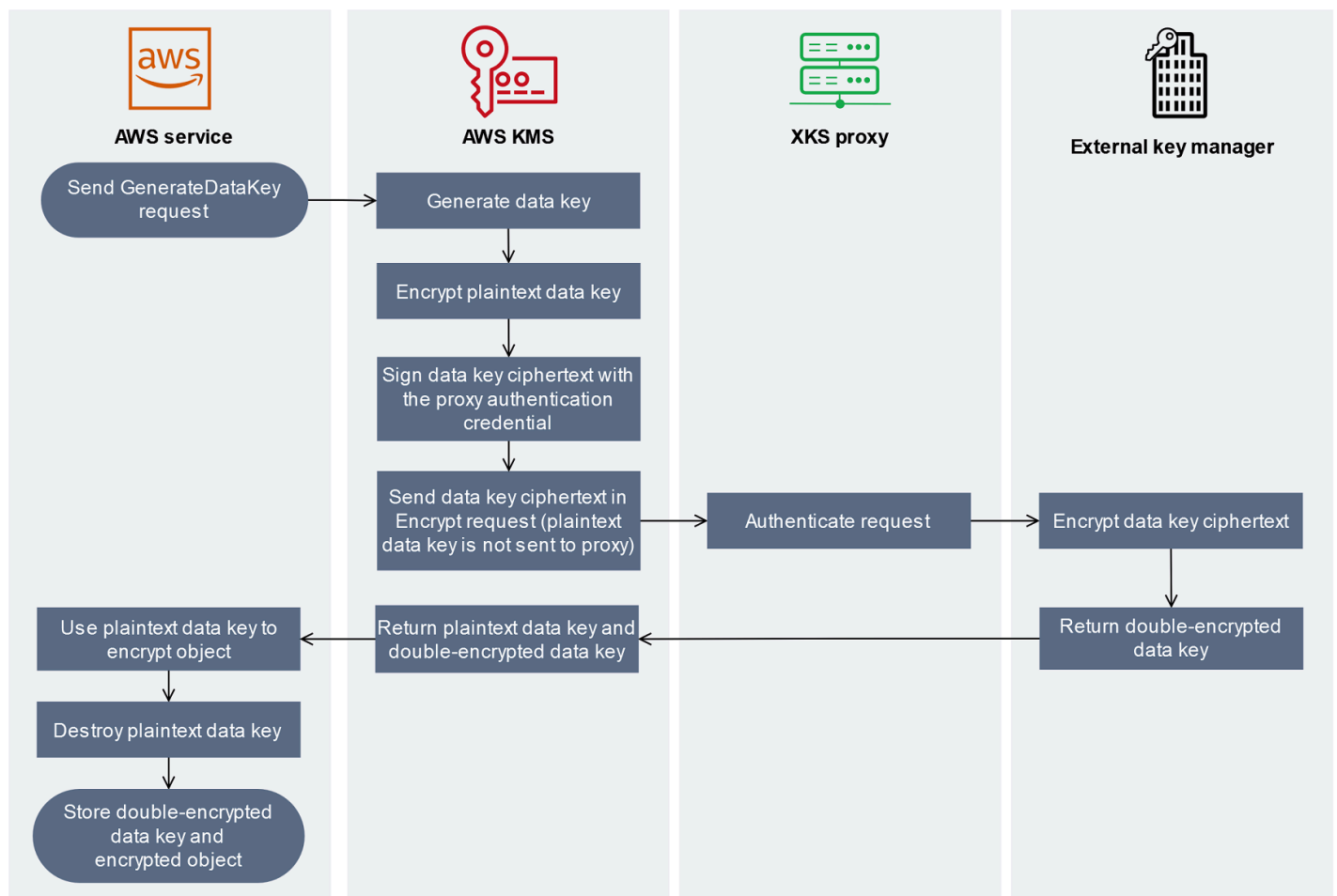
- Non è possibile decrittografare alcun testo cifrato crittografato da una chiave KMS in un archivio di chiavi esterno a, anche se si dispone del relativo materiale di AWS chiave esterno.
- Non puoi ricreare una chiave KMS eliminata da un archivio delle chiavi esterne, anche se disponi del relativo materiale della chiave. Ogni chiave KMS presenta metadati univoci inclusi nel testo criptato simmetrico. Una nuova chiave KMS non sarebbe in grado di decrittografare il testo criptato con la chiave originale, anche se utilizza lo stesso materiale della chiave esterna.

Per un esempio pratico di doppia crittografia, consulta [Funzionamento degli archivi delle chiavi esterne](#).

Funzionamento degli archivi delle chiavi esterne

L'[archivio delle chiavi esterne](#), il [proxy dell'archivio delle chiavi esterne](#) e il [gestore delle chiavi esterne](#) collaborano insieme per proteggere le tue risorse AWS . La procedura seguente descrive il flusso di lavoro di crittografia di un tipico Servizio AWS che esegue la crittografia di ogni oggetto con una chiave dati univoca protetta da una chiave KMS. In questo caso, hai scelto una chiave KMS in un archivio delle chiavi esterne per proteggere l'oggetto. L'esempio mostra come AWS KMS utilizza la [doppia crittografia](#) per proteggere la chiave di dati in transito e garantire che il testo cifrato generato da una chiave KMS in un archivio di chiavi esterno sia sempre almeno altrettanto efficace del testo cifrato crittografato da una chiave KMS simmetrica standard contenente materiale chiave. AWS KMS

I metodi di crittografia utilizzati da ogni effettivo che si integra con variano. Servizio AWS AWS KMS Per maggiori dettagli, consulta l'argomento "Protezione dei dati" nel capitolo Sicurezza della documentazione di Servizio AWS .



1. Aggiungi un nuovo oggetto alla tua Servizio AWS risorsa. Per crittografare l'oggetto, Servizio AWS invia una [GenerateDataKey](#) richiesta all' AWS KMS utilizzo di una chiave KMS nell'archivio di chiavi esterno.
2. AWS KMS genera una [chiave dati simmetrica a 256 bit e si prepara a inviare una copia della chiave](#) di dati in chiaro al gestore delle chiavi esterno tramite il proxy dell'archivio chiavi esterno. AWS KMS avvia il processo di [doppia crittografia](#) crittografando la chiave di dati in chiaro con il [materiale chiave associato alla chiave KMS nell'AWS KMS archivio chiavi esterno](#).
3. AWS KMS invia una richiesta di [crittografia](#) al proxy dell'archivio chiavi esterno associato all'archivio di chiavi esterno. La richiesta include il testo cifrato della chiave dati da crittografare e l'ID della [chiave esterna](#) associata alla chiave KMS. AWS KMS firma la richiesta utilizzando la [credenziale di autenticazione proxy](#) per il proxy di archiviazione delle chiavi esterno.

La copia non crittografata della chiave dati non viene inviata al proxy dell'archivio delle chiavi esterno.

4. Il proxy dell'archivio delle chiavi esterne autentica la richiesta di crittografia e quindi la trasmette al gestore delle chiavi esterne.

Alcuni proxy dell'archivio delle chiavi esterne implementano anche una [policy di autorizzazione](#) facoltativa che consente solo ai principali selezionati di eseguire operazioni in condizioni specifiche.

5. Il gestore delle chiavi esterne esegue la crittografia del testo criptato della chiave dati utilizzando la chiave esterna specificata e restituisce la chiave dati con doppia crittografia al proxy dell'archivio delle chiavi esterne che a sua volta la restituisce a AWS KMS.
6. AWS KMS restituisce la chiave di dati in testo semplice e la copia con doppia crittografia di tale chiave dati a. Servizio AWS
7. Servizio AWS utilizza la chiave dati in testo semplice per crittografare l'oggetto risorsa, distrugge la chiave di dati in testo semplice e archivia la chiave dati crittografata con l'oggetto crittografato.

Alcuni Servizi AWS potrebbero memorizzare nella cache la chiave di dati in testo semplice da utilizzare per più oggetti o da riutilizzare mentre la risorsa è in uso. Per informazioni dettagliate, vedi [In che modo le chiavi KMS inutilizzabili influiscono sulle chiavi dati](#).

[Per decrittografare l'oggetto crittografato, è Servizio AWS necessario inviare nuovamente la chiave di dati crittografata a AWS KMS in una richiesta Decrypt](#). Per decrittografare la chiave dati crittografata, è AWS KMS necessario inviare la chiave dati crittografata al proxy dell'archivio chiavi esterno con l'ID della chiave esterna. Se la richiesta di decrittografia al proxy dell'archivio chiavi esterno non riesce per qualsiasi motivo, non AWS KMS può decrittografare la chiave dati crittografata e non può decrittografare l' Servizio AWS oggetto crittografato.

Controlla l'accesso al tuo archivio di chiavi esterno

Tutte le funzionalità di controllo degli AWS KMS accessi ([politiche chiave](#), [politiche IAM](#) e [concessioni](#)) che utilizzi con le chiavi KMS standard funzionano allo stesso modo per le chiavi KMS in un archivio di chiavi esterno. Puoi utilizzare le policy IAM per controllare l'accesso alle operazioni API che creano e gestiscono archivi delle chiavi esterne. Utilizzi le policy IAM e le policy chiave per controllare l'accesso al tuo archivio AWS KMS keys di chiavi esterno. Puoi anche utilizzare [le policy di controllo del servizio](#) per la tua AWS organizzazione e le [policy degli endpoint VPC](#) per controllare l'accesso alle chiavi KMS nel tuo archivio di chiavi esterno.

Ti consigliamo di concedere a utenti e ruoli soltanto le autorizzazioni necessarie per le attività che sono supposti eseguire.

Argomenti

- [Autorizzazione dei gestori dell'archivio delle chiavi esterne](#)
- [Autorizzazione degli utenti delle chiavi KMS in archivi delle chiavi esterne](#)
- [Autorizzazione AWS KMS alla comunicazione con il proxy dell'archivio di chiavi esterno](#)
- [Autorizzazione proxy dell'archivio delle chiavi esterne \(facoltativo\)](#)
- [Autenticazione TLS reciproca \(facoltativa\)](#)

Autorizzazione dei gestori dell'archivio delle chiavi esterne

I principali che creano e gestiscono un archivio delle chiavi esterne necessitano di autorizzazioni per eseguire le operazioni dell'archivio delle chiavi personalizzate. L'elenco seguente descrive le autorizzazioni minime necessarie per i gestori dell'archivio delle chiavi esterne. Poiché un archivio chiavi personalizzato non è una AWS risorsa, non è possibile fornire l'autorizzazione a un archivio di chiavi esterno per i responsabili di altri. Account AWS

- `kms:CreateCustomKeyStore`
- `kms:DescribeCustomKeyStores`
- `kms:ConnectCustomKeyStore`
- `kms:DisconnectCustomKeyStore`
- `kms:UpdateCustomKeyStore`
- `kms>DeleteCustomKeyStore`

Per creare un archivio di chiavi esterno con la [connettività del servizio endpoint Amazon VPC e il servizio](#) endpoint VPC è di proprietà di un'altra persona Account AWS, avrai anche bisogno della seguente autorizzazione:

- `ec2:DescribeVPCEndpointServices`

I principali che creano un archivio delle chiavi esterne devono disporre dell'autorizzazione per creare e configurare i componenti di tale archivio. I principali possono creare archivi delle chiavi esterne solo nei propri account. Per creare un archivio delle chiavi esterne con [connettività del servizio endpoint VPC](#), i principali devono disporre dell'autorizzazione per creare i seguenti componenti:

- Un Amazon VPC
- Sottoreti pubbliche e private

- Un Network Load Balancer e un gruppo di destinazione
- Un servizio endpoint Amazon VPC

Per maggiori dettagli, consulta le sezioni [Identity and access management per Amazon VPC](#), [Identity and Access Management per endpoint VPC e servizi endpoint VPC](#) e [Autorizzazioni API di Elastic Load Balancing](#).

Autorizzazione degli utenti delle chiavi KMS in archivi delle chiavi esterne

I responsabili che creano e gestiscono AWS KMS keys nel tuo archivio di chiavi esterno richiedono [le stesse autorizzazioni](#) di coloro che creano e gestiscono qualsiasi chiave KMS. AWS KMS La [policy chiave predefinita](#) per le chiavi KMS in un archivio delle chiavi esterne è identica alla policy chiave predefinita per le chiavi KMS in AWS KMS. Il [controllo degli accessi basato su attributi](#) (ABAC), che utilizza tag e alias per controllare l'accesso alle chiavi KMS, sono efficaci anche nelle chiavi KMS negli archivi delle chiavi esterne.

I principali che utilizzano le chiavi KMS nell'archivio delle chiavi personalizzate per [operazioni di crittografia](#) devono disporre dell'autorizzazione per eseguire l'operazione di crittografia con la chiave KMS, ad esempio [kms:Decrypt](#). Puoi fornire queste autorizzazioni in una policy IAM o in una policy delle chiavi. I principali non hanno tuttavia bisogno di autorizzazioni supplementari per utilizzare una chiave KMS in un archivio delle chiavi personalizzate.

Per impostare un'autorizzazione che si applica solo alle chiavi KMS in un archivio delle chiavi esterne, utilizza la condizione della policy [kms:KeyOrigin](#) con un valore di `EXTERNAL_KEY_STORE`. Puoi utilizzare questa condizione per limitare l'autorizzazione [kms:](#) o qualsiasi `CreateKey` autorizzazione specifica per una risorsa chiave KMS. Ad esempio, la policy IAM seguente consente all'identità a cui è associata di chiamare le operazioni specificate in tutte le chiavi KMS, a condizione che le chiavi KMS si trovino in un archivio delle chiavi esterne. Nota che puoi limitare l'autorizzazione alle chiavi KMS in un archivio di chiavi esterno e alle chiavi KMS in un archivio di chiavi esterno Account AWS, ma non in un particolare archivio di chiavi esterno nell'account.

```
{
  "Sid": "AllowKeysInExternalKeyStores",
  "Effect": "Allow",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
```

```
"kms:GenerateDataKey*",
"kms:DescribeKey"
],
"Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
"Condition": {
  "StringEquals": {
    "kms:KeyOrigin": "EXTERNAL_KEY_STORE"
  }
}
}
```

Autorizzazione AWS KMS alla comunicazione con il proxy dell'archivio di chiavi esterno

AWS KMS comunica con il gestore delle chiavi esterno solo tramite il [proxy di archiviazione chiavi esterno](#) fornito dall'utente. AWS KMS si autentica con il proxy firmando le relative richieste utilizzando il [processo Signature Version 4 \(SigV4\)](#) con la credenziale di [autenticazione proxy dell'archivio chiavi esterno](#) specificata dall'utente. Se si utilizza la [connettività endpoint pubblica](#) per il proxy dell'archivio chiavi esterno, AWS KMS non richiede autorizzazioni aggiuntive.

Tuttavia, se utilizzi la [connettività del servizio endpoint VPC](#), devi AWS KMS autorizzare la creazione di un endpoint di interfaccia per il tuo servizio endpoint Amazon VPC. Questa autorizzazione è richiesta indipendentemente dal fatto che il proxy dell'archivio chiavi esterno si trovi nel tuo VPC o che il proxy dell'archivio chiavi esterno si trovi altrove, ma utilizzi il servizio endpoint VPC per comunicare. AWS KMS

AWS KMS Per consentire la creazione di un endpoint di interfaccia, utilizza la console [Amazon VPC](#) o [ModifyVpcEndpointServicePermissions](#) l'operazione. Consenti le autorizzazioni per il seguente principale: `cks.kms.<region>.amazonaws.com`.

Se il tuo servizio endpoint Amazon VPC è di proprietà di una persona Account AWS diversa da quella Account AWS proprietaria dell'archivio di chiavi esterne (XKS), dovrai anche consentire a XKS l'accesso al servizio endpoint VPC. A tale scopo, [consenti l' Account AWS ID XKS come principale](#) per il servizio endpoint Amazon VPC.

Same Account AWS

Se il servizio endpoint VPC è di proprietà dello stesso Account AWS archivio chiavi esterno, è necessario aggiungerlo AWS KMS all'elenco dei principali indirizzi consentiti per il servizio endpoint VPC.

L'esempio seguente utilizza il [modify-vpc-endpoint-service-permissions](#) AWS CLI comando per consentire la connessione AWS KMS al servizio endpoint VPC specificato nella regione Stati Uniti occidentali (Oregon) (us-west-2). Prima di utilizzare questo comando, sostituisci l'ID del servizio Amazon VPC Regione AWS con valori validi per la tua configurazione.

```
modify-vpc-endpoint-service-permissions
--service-id vpce-svc-12abc34567def0987
--add-allowed-principals '["cks.kms.us-west-2.amazonaws.com"]'
```

Per rimuovere questa autorizzazione, usa la [console Amazon VPC](#) o [ModifyVpcEndpointServicePermissions](#) con il `RemoveAllowedPrincipals` parametro.

Cross Account AWS

Quando il servizio endpoint VPC è di proprietà di un altro Account AWS, è necessario aggiungere entrambi AWS KMS e l'archivio chiavi esterno all'elenco dei principali indirizzi consentiti per il servizio endpoint VPC.

L'esempio seguente utilizza il [modify-vpc-endpoint-service-permissions](#) AWS CLI comando per consentire a entrambi AWS KMS e all'archivio di chiavi esterno (XKS) di connettersi al servizio endpoint VPC specificato nella regione Stati Uniti occidentali (Oregon) (us-west-2). Prima di utilizzare questo comando, sostituisci l'ID del servizio Amazon VPC e l'ARN Regione AWS principale IAM con valori validi per la tua configurazione. Il principale IAM deve essere sostituito con un principale nel proprietario di XKS. Account AWS

In questo esempio, `arn:aws:iam::123456789012:role/cks_role` è il principale IAM nell'account proprietario XKS, che verrà utilizzato per creare, aggiornare o connettere XKS al servizio endpoint VPC. Se desideri consentire a tutti i principali dell'account proprietario XKS di accedere al tuo servizio endpoint VPC, puoi specificare `arn:aws:iam::123456789012:root`

```
modify-vpc-endpoint-service-permissions
--service-id vpce-svc-12abc34567def0987
--add-allowed-principals '["cks.kms.us-west-2.amazonaws.com",
"arn:aws:iam::123456789012:role/cks_role"]'
```

Per rimuovere questa autorizzazione, usa la [console Amazon VPC](#) o [ModifyVpcEndpointServicePermissions](#) con il `RemoveAllowedPrincipals` parametro.

Autorizzazione proxy dell'archivio delle chiavi esterne (facoltativo)

Alcuni proxy degli archivi delle chiavi esterne implementano i requisiti di autorizzazione per l'uso delle relative chiavi esterne. Un proxy dell'archivio delle chiavi esterne è consentito, ma non obbligatorio, per progettare e implementare uno schema di autorizzazione che consenta a determinati utenti di richiedere determinate operazioni solo in base ad alcune condizioni. Ad esempio, un proxy potrebbe essere configurato per consentire all'utente A di eseguire la crittografia con una particolare chiave esterna, ma non di effettuare l'operazione inversa.

L'autorizzazione proxy è indipendente dall'[autenticazione proxy basata su SigV4 che AWS KMS richiede tutti i proxy](#) di archiviazione chiavi esterni. È inoltre indipendente dalle policy delle chiavi, dalle policy IAM e dalle concessioni che autorizzano l'accesso alle operazioni che riguardano l'archivio delle chiavi esterne o le relative chiavi KMS.

Per abilitare l'autorizzazione da parte del proxy di archiviazione delle chiavi esterno, AWS KMS include i metadati in ogni [richiesta API proxy](#), tra cui il chiamante, la chiave KMS, l'operazione e (se presente). AWS KMS Servizio AWS I metadati della richiesta per la versione 1 (v1) dell'API proxy dell'archivio delle chiavi esterne sono i seguenti.

```
"requestMetadata": {  
  "awsPrincipalArn": string,  
  "awsSourceVpc": string, // optional  
  "awsSourceVpce": string, // optional  
  "kmsKeyArn": string,  
  "kmsOperation": string,  
  "kmsRequestId": string,  
  "kmsViaService": string // optional  
}
```

Ad esempio, è possibile configurare il proxy per consentire le richieste provenienti da un particolare principale (`awsPrincipalArn`), ma solo quando la richiesta viene effettuata per conto del principale da un particolare Servizio AWS (`kmsViaService`).

Se l'autorizzazione del proxy fallisce, l'AWS KMS operazione correlata fallisce e viene visualizzato un messaggio che spiega l'errore. Per maggiori dettagli, consulta [Problemi relativi all'autorizzazione proxy](#).

Autenticazione TLS reciproca (facoltativa)

Per consentire al proxy dell'archivio chiavi esterno di autenticare le richieste AWS KMS, AWS KMS firma tutte le richieste al proxy dell'archivio chiavi esterno con la [credenziale di autenticazione proxy](#) Signature V4 (SigV4) per l'archivio chiavi esterno.

Per garantire ulteriormente che il proxy dell'archivio chiavi esterno risponda solo alle AWS KMS richieste, alcuni proxy di chiavi esterni supportano la Mutual Transport Layer Security (MTLS), in cui entrambe le parti di una transazione utilizzano i certificati per l'autenticazione reciproca. mTLS aggiunge l'autenticazione lato client, in cui il server proxy dell'archivio chiavi esterno autentica il AWS KMS client, all'autenticazione lato server fornita dal TLS standard. Nel raro caso in cui le credenziali di autenticazione proxy siano compromesse, l'autenticazione TLS reciproca impedisce a terzi di effettuare richieste API al proxy dell'archivio delle chiavi esterne.

Per implementare l'autenticazione TLS reciproca, configura il proxy dell'archivio delle chiavi esterne in modo che accetti solo certificati TLS lato client con le seguenti proprietà:

- Il nome comune dell'oggetto sul certificato TLS deve essere `cks.kms.<Region>.amazonaws.com`, ad esempio `cks.kms.eu-west-3.amazonaws.com`.
- Il certificato deve essere concatenato a un'autorità di certificazione associata ai [servizi di trust di Amazon](#).

Scegli un'opzione di connettività proxy per l'archivio chiavi esterno

Prima di creare l'archivio di chiavi esterno, scegli l'opzione di connettività che determina il modo in cui AWS KMS comunica con i componenti dell'archivio chiavi esterno. L'opzione di connettività scelta determina il resto del processo di pianificazione.

Se state creando un archivio di chiavi esterno, dovete determinare in che modo AWS KMS comunica con il proxy dell'[archivio chiavi esterno](#). Questa scelta determinerà quali componenti sono necessari e come configurarli. AWS KMS supporta le seguenti opzioni di connettività.

- [Connettività dell'endpoint pubblico](#)
- [Connettività del servizio endpoint VPC](#)

Scegli l'opzione che soddisfa gli obiettivi di prestazioni e sicurezza.

Prima di iniziare, [verifica che sia necessario un archivio delle chiavi esterne](#). La maggior parte dei clienti può utilizzare chiavi KMS supportate da materiale AWS KMS chiave.

Considerazioni

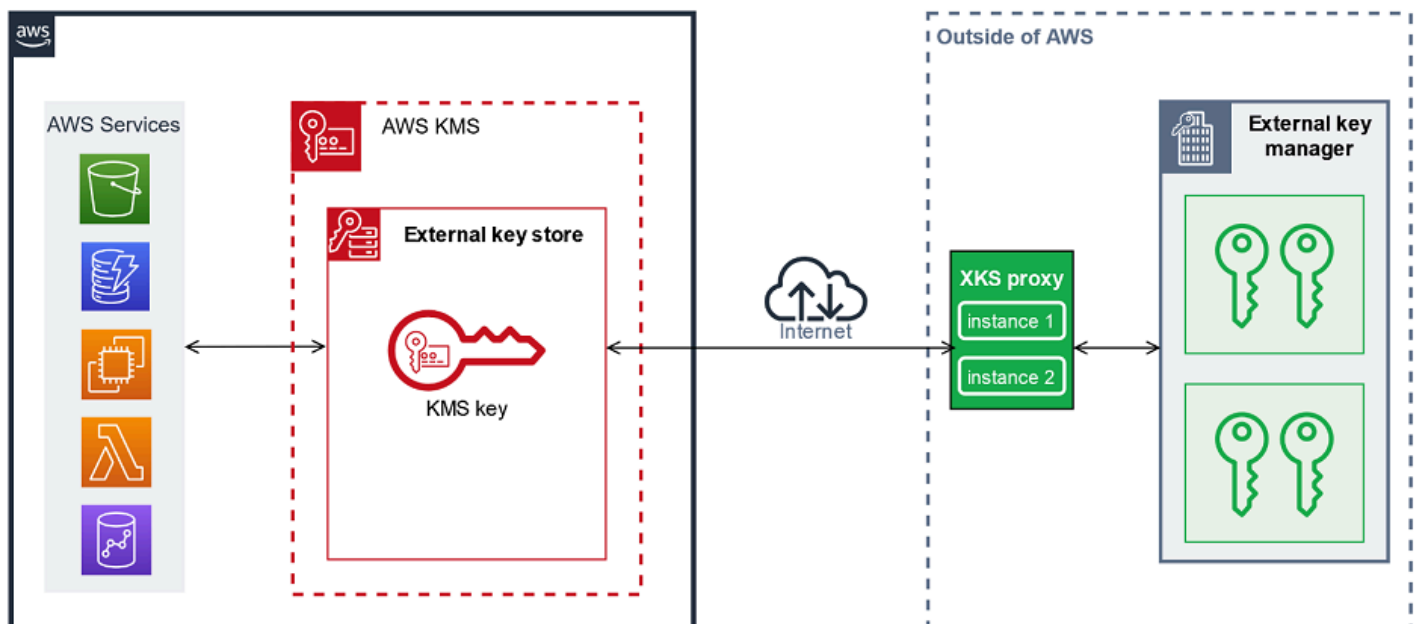
- Se il proxy dell'archivio delle chiavi esterne è integrato nel gestore delle chiavi esterne, la connettività potrebbe essere predeterminata. Per informazioni, consulta la documentazione del gestore delle chiavi esterne o del proxy dell'archivio delle chiavi esterne.
- Puoi [modificare l'opzione di connettività proxy dell'archivio delle chiavi esterne](#) anche su un archivio delle chiavi esterne operativo. Tuttavia, il processo deve essere pianificato ed eseguito con cura per ridurre al minimo le interruzioni, evitare errori e garantire l'accesso continuo alle chiavi crittografiche che crittografano i dati.

Connettività dell'endpoint pubblico

AWS KMS si connette al proxy di archiviazione delle chiavi esterno (proxy XKS) su Internet utilizzando un endpoint pubblico.

Questa opzione di connettività è molto semplice da configurare e gestire e si allinea bene con alcuni modelli di gestione delle chiavi. Tuttavia, potrebbe non soddisfare i requisiti di sicurezza di alcune organizzazioni.

XKS proxy connected by a public endpoint



Requisiti

Se scegli la connettività all'endpoint pubblico, è necessario quanto segue.

- Il proxy dell'archivio delle chiavi esterne deve essere raggiungibile da un endpoint indirizzabile pubblicamente.
 - Puoi utilizzare lo stesso endpoint pubblico per più archivi delle chiavi esterne, a condizione che utilizzino valori diversi per il [percorso URI proxy](#).
 - Non è possibile utilizzare lo stesso endpoint per un archivio di chiavi esterno con connettività endpoint pubblica e qualsiasi archivio di chiavi esterno con connettività dei servizi endpoint VPC nello stesso archivio Regione AWS, anche se gli archivi di chiavi si trovano in archivi diversi.
- Account AWS
- Devi ottenere un certificato TLS emesso da un'autorità di certificazione pubblica supportata per gli archivi delle chiavi esterne. Per un elenco, consulta [Autorità di certificazione attendibili](#).

Il nome comune del soggetto (CN) sul certificato TLS deve corrispondere al nome di dominio nell'[endpoint URI proxy](#) del proxy dell'archivio delle chiavi esterne. Ad esempio, se l'endpoint pubblico è `https://myproxy.xks.example.com` il TLS, il CN sul certificato TLS deve essere `myproxy.xks.example.com` o `*.xks.example.com`.

- Assicurati che tutti i firewall tra AWS KMS e il proxy dell'archivio di chiavi esterno consentano il traffico da e verso la porta 443 del proxy. AWS KMS comunica sulla porta 443 tramite IPv4. Questo valore non è configurabile.

Per informazioni su tutti i requisiti di un archivio delle chiavi esterne, consulta [Assemblare i prerequisiti](#).

Connettività del servizio endpoint VPC

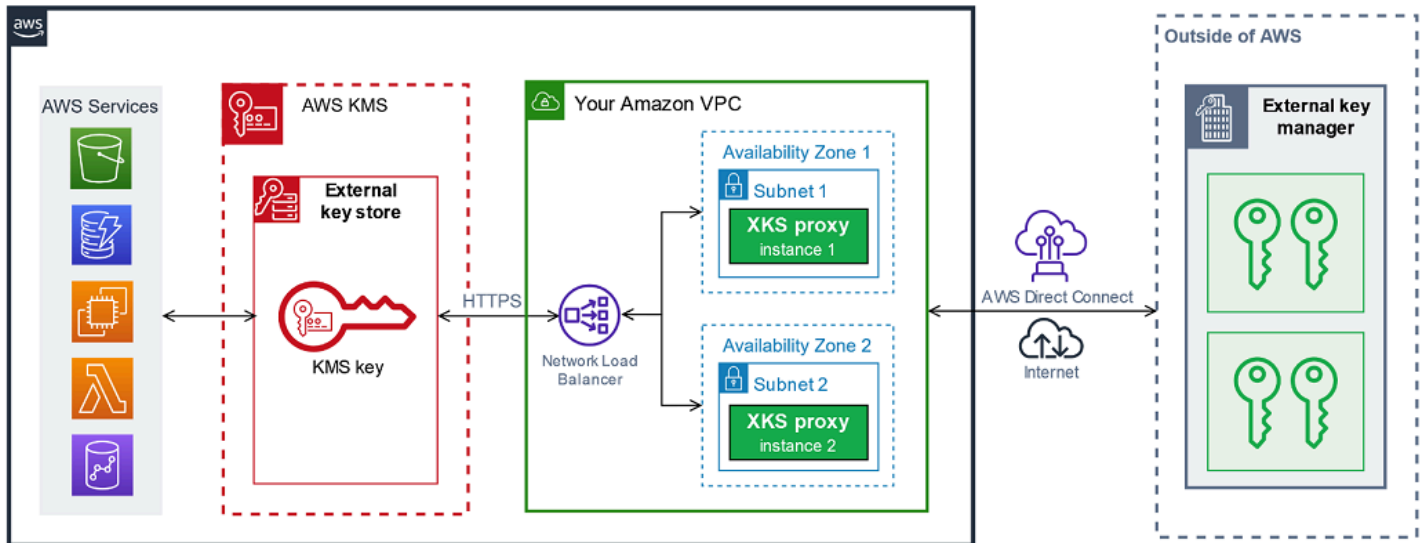
AWS KMS si connette al proxy di archiviazione chiavi esterno (proxy XKS) creando un endpoint di interfaccia verso un servizio endpoint Amazon VPC da te creato e configurato. Sei responsabile della [creazione del servizio endpoint VPC](#) e della connessione del VPC al gestore delle chiavi esterne.

Il tuo servizio endpoint può utilizzare qualsiasi opzione [network-to-AmazonVPC supportata](#) per le comunicazioni, tra cui. [AWS Direct Connect](#)

Questa opzione di connettività è più complessa da configurare e gestire. Ma utilizza AWS PrivateLink, il che consente di AWS KMS connettersi privatamente al tuo Amazon VPC e al tuo proxy di archiviazione chiavi esterno senza utilizzare la rete Internet pubblica.

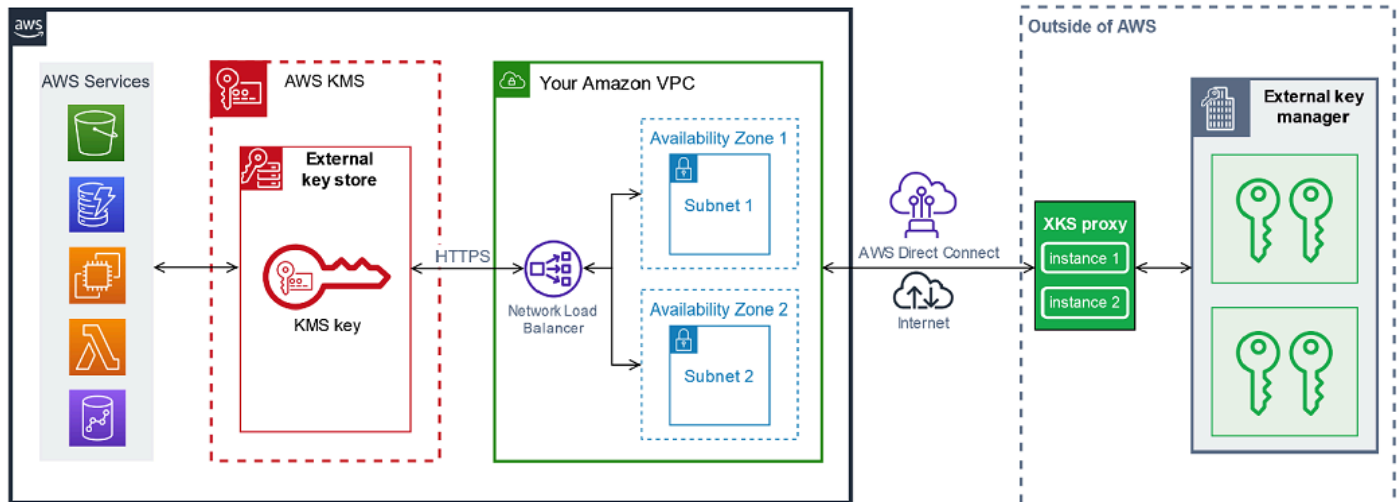
Puoi posizionare il proxy dell'archivio delle chiavi esterne in Amazon VPC.

XKS proxy hosted in Amazon VPC



In alternativa, puoi localizzare il tuo proxy di archiviazione chiavi esterno all'esterno Cloud AWS e utilizzare il servizio endpoint Amazon VPC solo per comunicazioni sicure con AWS KMS

XKS proxy connected via Amazon VPC endpoint service



Puoi anche connettere un archivio di chiavi esterno a un servizio endpoint Amazon VPC di proprietà di un altro Account AWS. Entrambi Account AWS necessitano delle [autorizzazioni necessarie](#) per consentire le comunicazioni tra AWS KMS e il servizio endpoint VPC.

Ulteriori informazioni:

- Rivedi il processo di creazione di un archivio delle chiavi esterne, incluso [l'assemblaggio dei prerequisiti](#). Ti aiuterà a verificare di disporre di tutti i componenti necessari per la creazione dell'archivio delle chiavi esterne.
- Scopri come [controllare l'accesso all'archivio delle chiavi esterne](#), comprese le autorizzazioni richieste dagli amministratori e dagli utenti dell'archivio.
- Scopri le [CloudWatch metriche e le dimensioni di Amazon](#) registrate per gli AWS KMS archivi di chiavi esterni. Ti consigliamo di creare allarmi per monitorare l'archivio delle chiavi esterne, in modo da poter rilevare fin dal principio eventuali segnali relativi a problemi operativi e prestazionali.

Configurazione della connettività del servizio endpoint VPC

Utilizza le indicazioni in questa sezione per creare e configurare AWS le risorse e i componenti correlati necessari per un archivio di chiavi esterno che utilizza la connettività del servizio [endpoint VPC](#). Le risorse elencate per questa opzione di connettività sono un supplemento alle [risorse necessarie per tutti gli archivi delle chiavi esterne](#). Dopo aver creato e configurato le risorse necessarie, puoi [creare l'archivio delle chiavi esterne](#).

Puoi localizzare il tuo proxy di archiviazione chiavi esterno nel tuo Amazon VPC o localizzare il proxy all'esterno AWS e utilizzare il servizio di endpoint VPC per la comunicazione.

Prima di iniziare, [verifica che sia necessario un archivio delle chiavi esterne](#). La maggior parte dei clienti può utilizzare chiavi KMS supportate da materiale chiave. AWS KMS

Note

Alcuni degli elementi necessari per la connettività del servizio endpoint VPC potrebbero essere inclusi nel gestore delle chiavi esterne. Inoltre, il software potrebbe avere requisiti di configurazione aggiuntivi. Prima di creare e configurare le AWS risorse in questa sezione, consulta la documentazione del proxy e del gestore delle chiavi.

Argomenti

- [Requisiti per la connettività del servizio endpoint VPC](#)
- [Fase 1: creazione di un Amazon VPC e delle sottoreti](#)
- [Fase 2: Creare un gruppo target](#)
- [Fase 3: Creare un sistema di bilanciamento del carico di rete](#)

- [Fase 4: Creare un servizio endpoint VPC](#)
- [Fase 5: Verifica il tuo nome di dominio DNS privato](#)
- [Fase 6: Autorizzazione AWS KMS alla connessione al servizio endpoint VPC](#)

Requisiti per la connettività del servizio endpoint VPC

Se scegli la connettività del servizio endpoint VPC per l'archivio delle chiavi esterne, sono necessarie le seguenti risorse.

- Un Amazon VPC collegato al gestore delle chiavi esterne. Deve avere almeno due [sottoreti](#) private in due zone di disponibilità diverse.

Puoi utilizzare un Amazon VPC esistente, a condizione che [soddisfi i requisiti](#) per l'utilizzo con un archivio delle chiavi esterne. Più archivi delle chiavi esterne possono condividere un Amazon VPC, ma ogni archivio deve avere il proprio servizio endpoint VPC e il proprio nome DNS privato.

- Un [servizio endpoint Amazon VPC basato su un AWS PrivateLink](#) con un [Network Load Balancer](#) e un [gruppo di destinazione](#).

Il servizio endpoint non può richiedere l'accettazione. Inoltre, devi aggiungere AWS KMS come principale consentito. Ciò consente di AWS KMS creare endpoint di interfaccia in modo che possa comunicare con il proxy di archiviazione delle chiavi esterno.

- Un nome DNS privato per il servizio endpoint VPC univoco nella Regione AWS.

Il nome DNS privato deve essere un sottodominio di un dominio pubblico di livello superiore. Ad esempio, se il nome DNS privato è `myproxy-private.xks.example.com`, deve essere un sottodominio di un dominio pubblico come `xks.example.com` o `example.com`.

Devi [verificare la proprietà](#) del dominio DNS per il nome DNS privato.

- Un certificato TLS emesso da un'[autorità di certificazione pubblica supportata](#) per il proxy dell'archivio delle chiavi esterne.

Il nome comune del soggetto (CN) sul certificato TLS deve corrispondere al nome DNS privato. Ad esempio, se il nome DNS privato è `myproxy-private.xks.example.com`, il CN sul certificato TLS deve essere `myproxy-private.xks.example.com` o `*.xks.example.com`.

- Per ridurre al minimo la latenza di rete, create i AWS componenti nel [supporto Regione AWS](#) più vicino al gestore di [chiavi esterno](#). Se possibile, scegli una regione con un tempo di andata e ritorno (RTT) della rete di 35 millisecondi o meno.

Per informazioni su tutti i requisiti di un archivio delle chiavi esterne, consulta [Assemblare i prerequisiti](#).

Fase 1: creazione di un Amazon VPC e delle sottoreti

La connettività del servizio endpoint VPC richiede un Amazon VPC connesso al gestore delle chiavi esterne con almeno due sottoreti private. Puoi creare un Amazon VPC o utilizzarne uno esistente che soddisfi i requisiti per gli archivi delle chiavi esterne. Per informazioni sulla creazione di un nuovo Amazon VPC, consulta [Creazione di un VPC](#) nella Guida per l'utente di Amazon Virtual Private Cloud.

Requisiti per Amazon VPC

Il servizio endpoint Amazon VPC deve avere le seguenti proprietà per funzionare con archivi di chiavi esterni.

- Deve trovarsi in una [regione supportata](#) come archivio di chiavi esterno.
- Richiede almeno due sottoreti private, ognuna in una zona di disponibilità diversa.
- L'intervallo di indirizzi IP privati di Amazon VPC non deve sovrapporsi all'intervallo di indirizzi IP privati del data center che ospita il [gestore delle chiavi esterne](#).
- Tutti i componenti devono essere utilizzati IPv4.

Le opzioni per connettere Amazon VPC al proxy dell'archivio delle chiavi esterne sono molteplici. Scegli un'opzione che soddisfi le tue esigenze di prestazioni e sicurezza. Per un elenco, consulta [Connect your VPC ad altre reti](#) e Opzioni di connettività [Network-to-Amazon VPC](#). Per ulteriori dettagli, consulta [Direct Connect](#) e la [Guida per l'utente di AWS Site-to-Site VPN](#).

Creazione di un Amazon VPC per l'archivio delle chiavi esterne

Utilizza le istruzioni seguenti per creare un Amazon VPC per l'archivio delle chiavi esterne. Un Amazon VPC è necessario solo se scegli l'opzione di [connettività del servizio endpoint VPC](#). Puoi utilizzare un Amazon VPC esistente, a condizione che soddisfi i requisiti di un archivio delle chiavi esterne.

Segui le istruzioni nell'argomento [Creazione di VPC, sottoreti e altre risorse VPC](#) utilizzando i seguenti valori obbligatori. Per gli altri campi, accetta i valori predefiniti e immetti i nomi come richiesto.

Campo	Valore
IPv4 blocco CIDR	Inserisci gli indirizzi IP per il VPC. L'intervallo di indirizzi IP privati di Amazon VPC non deve sovrapporsi all'intervallo di indirizzi IP privati del data center che ospita il gestore delle chiavi esterne .
Numero di zone di disponibilità () AZs	2 o più
Numero di sottoreti pubbliche	Non è necessario indicare alcun valore (0)
Numero di sottoreti private	Una per ogni zona di disponibilità
Gateway NAT	Non è necessario indicare alcun valore.
Endpoint VPC	Non è necessario indicare alcun valore.
Enable DNS hostnames (Abilita hostname DNS)	Sì
Abilita risoluzione DNS	Sì

Assicurati di testare la comunicazione VPC. Ad esempio, se il tuo proxy di archiviazione chiavi esterno non si trova nel tuo Amazon VPC, crea un' EC2 istanza Amazon nel tuo Amazon VPC, verifica che Amazon VPC possa comunicare con il tuo proxy di archiviazione chiavi esterno.

Connessione del VPC al gestore delle chiavi esterne

Connetti il VPC al data center che ospita il gestore delle chiavi esterne utilizzando una delle [opzioni di connettività di rete](#) supportate da Amazon VPC. Assicurati che l' EC2 istanza Amazon nel VPC (o il proxy dell'archivio chiavi esterno, se si trova nel VPC) possa comunicare con il data center e il gestore delle chiavi esterno.

Fase 2: Creare un gruppo target

Prima di creare il servizio endpoint VPC richiesto, crea i componenti necessari, vale a dire un Network Load Balancer e un gruppo di destinazione. Il Network Load Balancer distribuisce le richieste tra più destinazioni integre, ognuna delle quali può soddisfare la richiesta. In questo passaggio, crea un gruppo di destinazione con almeno due host per il proxy dell'archivio delle chiavi esterne e registra gli indirizzi IP con il gruppo di destinazione.

Segui le istruzioni nell'argomento [Configurazione di un gruppo di destinazione](#) utilizzando i seguenti valori obbligatori. Per gli altri campi, accetta i valori predefiniti e immetti i nomi come richiesto.

Campo	Valore
Target type (Tipo di destinazione)	Indirizzi IP
Protocollo	TCP
Porta	443
Tipo di indirizzo IP	IPv4
VPC	Scegli il VPC in cui creare il servizio endpoint VPC per l'archivio delle chiavi esterne.
Protocollo e percorso di controllo dell'integrità	Il protocollo e il percorso di controllo dell'integrità saranno diversi a seconda della configurazione del proxy dell'archivio delle chiavi esterne. Consulta la documentazione del gestore delle chiavi esterne o del proxy dell'archivio delle chiavi esterne. Per informazioni generali sulla configurazione dei controlli dell'integrità per i gruppi di destinazione, consulta Controlli dell'integrità per i gruppi di destinazioni nella Guida per l'utente di Elastic Load Balancing per Network Load Balancer.
Rete	Altro indirizzo IP privato
IPv4 indirizzo	Gli indirizzi privati del proxy dell'archivio delle chiavi esterne

Campo	Valore
Porte	443

Fase 3: Creare un sistema di bilanciamento del carico di rete

Il Network Load Balancer distribuisce il traffico di rete, comprese le richieste provenienti da AWS KMS al proxy dell'archivio delle chiavi esterne, fino alle destinazioni configurate.

Segui le istruzioni nell'argomento [Configurare un sistema di bilanciamento del carico e un ascoltatore](#) per configurare e aggiungere un ascoltatore e creare un sistema di bilanciamento del carico utilizzando i seguenti valori obbligatori. Per gli altri campi, accetta i valori predefiniti e immetti i nomi come richiesto.

Campo	Valore
Schema	Interno
Tipo di indirizzo IP	IPv4
Mappatura della rete	Scegli il VPC in cui creare il servizio endpoint VPC per l'archivio delle chiavi esterne.
Mapping	Scegli entrambe le zone di disponibilità (almeno due) configurate per le sottoreti VPC. Verifica i nomi delle sottoreti e l'indirizzo IP privato.
Protocollo	TCP
Porta	443
Azione predefinita: Inoltra a	Scegli il gruppo di destinazione per il Network Load Balancer.

Fase 4: Creare un servizio endpoint VPC

In genere, la creazione di un endpoint è destinata a un servizio. Tuttavia, quando crei un servizio endpoint VPC, sei il fornitore e AWS KMS crei un endpoint per il tuo servizio. Per un archivio delle

chiavi esterne, crea un servizio endpoint VPC con il Network Load Balancer creato nel passaggio precedente. Il servizio endpoint VPC può trovarsi nello stesso Account AWS archivio di chiavi esterno o in un altro. Account AWS

Più archivi delle chiavi esterne possono condividere un Amazon VPC, ma ogni archivio deve avere il proprio servizio endpoint VPC e il proprio nome DNS privato.

Segui le istruzioni nell'argomento [Creazione di un servizio endpoint](#) per creare il servizio endpoint VPC con i seguenti valori obbligatori. Per gli altri campi, accetta i valori predefiniti e immetti i nomi come richiesto.

Campo	Valore
Nuovo tipo di load balancer	Rete
Sistemi di bilanciamento del carico disponibili	Scegli il Network Load Balancer creato nella fase precedente. Se il nuovo sistema di bilanciamento del carico non compare nell'elenco, verifica che il suo stato sia attivo. Potrebbero essere necessari alcuni minuti prima che lo stato del sistema di bilanciamento del carico passi dal provisioning ad attivo.
Accettazione richiesta	Falso. Deseleziona la casella di controllo. Non richiedono l'accettazione. AWS KMS non può connettersi al servizio endpoint VPC senza un'accettazione manuale. Se è richiesta l'accettazione, i tentativi di creare l'archivio delle chiavi esterne falliscono con un'eccezione <code>XksProxyInvalidConfigurationException</code>.
Abilita nome DNS privato	Associa un nome DNS privato al servizio
Nome DNS privato	Inserisci un nome DNS privato che sia univoco nella Regione AWS. Il nome DNS privato deve essere un sottodominio di un dominio pubblico di livello superiore. Ad esempio, se il nome DNS privato è <code>myproxy-private.xks.example.com</code>, deve essere un sottodominio di un dominio pubblico come <code>xks.example.com</code> o <code>example.com</code>.

Campo	Valore
	Questo nome DNS privato deve corrispondere al nome comune del soggetto (CN) nel certificato TLS configurato sul proxy dell'archivio delle chiavi esterne. Ad esempio, se il nome DNS privato è <code>myproxy-private.xks.example.com</code>, il CN sul certificato TLS deve essere <code>myproxy-private.xks.example.com</code> o <code>*.xks.example.com</code>. Se il certificato e il nome DNS privato non corrispondono, i tentativi di connettere un archivio delle chiavi esterne al relativo proxy hanno esito negativo con un codice di errore di connessione di <code>XKS_PROXY_INVALID_TLS_CONFIGURATION</code>. Per informazioni dettagliate, vedi Errori di configurazione generale.
Tipi di indirizzo IP supportati	IPv4

Fase 5: Verifica il tuo nome di dominio DNS privato

Quando crei il servizio endpoint VPC, lo stato di verifica del dominio è `pendingVerification`. Prima di creare un archivio delle chiavi esterne con il servizio endpoint VPC, tale stato deve essere `verified`. Per verificare di essere il proprietario del dominio associato al nome DNS privato, devi creare un record TXT in un server DNS pubblico.

Ad esempio, se il nome DNS privato per il tuo servizio endpoint VPC `myproxy-private.xks.example.com` è, devi creare un record TXT in un dominio pubblico, ad esempio `example.com` o, a seconda di `xks.example.com` quale sia pubblico. AWS PrivateLink cerca il record TXT prima e poi `xks.example.com` `example.com`.

Tip

Dopo aver aggiunto un record TXT, potrebbero essere necessari alcuni minuti prima che il valore di `Domain verification status` (Stato di verifica del dominio) passi da `pendingVerification` a `verify`.

Per iniziare, individua lo stato di verifica del dominio utilizzando uno dei metodi seguenti. I valori validi sono `verified`, `pendingVerification` e `failed`.

- Nella [console Amazon VPC](#), scegli Endpoint services (Servizi endpoint), quindi seleziona il servizio endpoint. Nel riquadro dei dettagli, vedi Domain verification status (Stato di verifica del dominio).
- Usa l'[DescribeVpcEndpointServiceConfigurations](#) operazione. Il valore State si trova nel campo `ServiceConfigurations.PrivateDnsNameConfiguration.State`.

Se lo stato della verifica non è `verified`, segui le istruzioni nell'argomento [Verifica della proprietà del dominio](#) per aggiungere un record TXT al server DNS del dominio e verificare che il record TXT sia pubblicato. Quindi controlla nuovamente lo stato della verifica.

Non è necessario creare un record A per il nome del dominio DNS privato. Quando AWS KMS crea un endpoint di interfaccia per il tuo servizio endpoint VPC AWS PrivateLink, crea automaticamente una zona ospitata con il record A richiesto per il nome di dominio privato nel VPC. AWS KMS Per gli archivi delle chiavi esterne con connettività del servizio endpoint VPC, ciò accade quando [colleghi l'archivio delle chiavi esterne](#) al relativo proxy.

Fase 6: Autorizzazione AWS KMS alla connessione al servizio endpoint VPC

Consulta le seguenti procedure per gestire le autorizzazioni del servizio endpoint Amazon VPC. Ogni passaggio dipende dalla connettività e dalla configurazione tra l'archivio chiavi esterno, il servizio endpoint VPC e. Account AWS

Same Account AWS

Se il servizio endpoint VPC è di proprietà dello stesso Account AWS archivio chiavi esterno, è necessario aggiungerlo AWS KMS all'elenco dei principali indirizzi consentiti per il servizio endpoint VPC. Ciò consente di AWS KMS creare endpoint di interfaccia per il servizio endpoint VPC. Se non AWS KMS è un'opzione principale consentita, i tentativi di creare un archivio di chiavi esterno falliranno con un'eccezione. `XksProxyVpcEndpointServiceNotFoundException`

Segui le istruzioni nell'argomento [Gestione delle autorizzazioni](#) della Guida di AWS PrivateLink. Utilizza il seguente valore obbligatorio.

Campo	Valore
AWS KMS	<code>cks.kms.<region>.amazonaws.com</code>
ARN	Ad esempio, <code>cks.kms.us-east-1.amazonaws.com</code>

Cross Account AWS

Quando il tuo servizio endpoint VPC è di proprietà di un altro, Account AWS devi aggiungere entrambi AWS KMS e il tuo account all'elenco Consenti principali. Ciò consente AWS KMS e al tuo archivio di chiavi esterno di creare endpoint di interfaccia per il tuo servizio endpoint VPC. Se AWS KMS non rappresenta un principale consentito, i tentativi di creare un archivio delle chiavi esterne avranno esito negativo con un'eccezione `XksProxyVpcEndpointServiceNotFoundException`. Dovrai fornire l' Account AWS ARN in cui risiede l'archivio delle chiavi esterno.

Segui le istruzioni nell'argomento [Gestione delle autorizzazioni](#) della Guida di AWS PrivateLink . Utilizza il seguente valore obbligatorio.

Campo	Valore
AWS KMS ARN	<code>cks.kms.<region>.amazonaws.com</code> Ad esempio, <code>cks.kms.us-east-1.amazonaws.com</code>
Account AWS ARN	<code>arn:aws:iam:: 111122223333 :role/role_name</code> Ad esempio, <code>arn:aws:iam:: 123456789012 :role/cks_role</code>

Successivo: [Creare un archivio di chiavi esterno](#)

Creare un archivio di chiavi esterno

È possibile creare uno o più archivi di chiavi esterni in ciascuna Account AWS regione. Ogni archivio chiavi esterno deve essere associato a un gestore di chiavi esterno a e a un proxy di AWS archiviazione chiavi esterno (proxy XKS) che media la comunicazione tra AWS KMS e il gestore di chiavi esterno. Per informazioni dettagliate, vedi [Scegli un'opzione di connettività proxy per l'archivio chiavi esterno](#). Prima di iniziare, [verifica che sia necessario un archivio delle chiavi esterne](#). La maggior parte dei clienti può utilizzare chiavi KMS supportate da AWS KMS materiale chiave.

 Tip

Alcuni gestori delle chiavi esterne offrono un metodo più semplice per creare un relativo archivio. Per ulteriori informazioni, consulta la documentazione del gestore delle chiavi esterne.

Prima di creare l'archivio delle chiavi esterne, devi [assemblare i prerequisiti](#). Durante il processo di creazione, specifica le proprietà dell'archivio delle chiavi esterne. La cosa più importante è indicare se l'archivio chiavi esterno AWS KMS utilizza un [endpoint pubblico](#) o un servizio [endpoint VPC](#) per connettersi al proxy dell'archivio chiavi esterno. È inoltre necessario specificare i dettagli della connessione, incluso l'endpoint URI del proxy e il percorso all'interno di tale endpoint proxy AWS KMS da cui invia le richieste API al proxy.

Considerazioni

- KMS non può comunicare IPv6 con archivi di chiavi esterni.
- Se utilizzi la connettività degli endpoint pubblici, assicurati che sia in AWS KMS grado di comunicare con il tuo proxy su Internet utilizzando una connessione HTTPS. Ciò include la configurazione di TLS sul proxy dell'archivio chiavi esterno e la garanzia che tutti i firewall tra AWS KMS e il proxy consentano il IPv4 traffico da e verso la porta 443 del proxy. Durante la creazione di un archivio di chiavi esterno con connettività endpoint pubblica, AWS KMS verifica la connessione inviando una richiesta di stato al proxy dell'archivio chiavi esterno. Questo test verifica che l'endpoint sia raggiungibile e che il proxy accetti una richiesta firmata con le [credenziali di autenticazione proxy dell'archivio delle chiavi esterne](#). Se tale richiesta di test fallisce, l'operazione di creazione dell'archivio delle chiavi esterne ha esito negativo.
- Se utilizzi la connettività del servizio endpoint VPC, assicurati che il Network Load Balancer, il nome DNS privato e il servizio endpoint VPC siano operativi e configurati correttamente. Se il proxy dell'archivio chiavi esterno non si trova nel VPC, devi assicurarti che il servizio endpoint VPC possa comunicare con il proxy dell'archivio chiavi esterno. (AWS KMS verifica la connettività del servizio endpoint VPC quando si [collega l'archivio chiavi esterno al relativo proxy dell'archivio chiavi esterno](#).)
- AWS KMS registra i [CloudWatch parametri e le dimensioni di Amazon](#), in particolare per gli archivi di chiavi esterni. I grafici di monitoraggio basati su alcune di queste metriche vengono visualizzati nella AWS KMS console per ogni archivio di chiavi esterno. Ti consigliamo vivamente di utilizzare queste metriche per creare allarmi che monitorino il tuo archivio di chiavi esterno. in modo da poter

rilevare eventuali segnali relativi a problemi operativi e prestazionali prima che si verifichino. Per istruzioni, consulta [Monitora gli archivi di chiavi esterni](#).

- Gli archivi delle chiavi esterne sono soggetti a [quote di risorse](#). L'uso delle chiavi KMS in un archivio delle chiavi esterne è soggetto a [quote di richieste](#). Esamina queste quote prima di progettare l'implementazione dell'archivio delle chiavi esterne.

Note

Rivedi la tua configurazione per verificare eventuali dipendenze circolari che potrebbero impedirne il funzionamento.

Ad esempio, se crei il proxy dell'archivio chiavi esterno utilizzando AWS risorse, assicurati che il funzionamento del proxy non richieda la disponibilità di una chiave KMS in un archivio di chiavi esterno a cui si accede tramite quel proxy.

Tutti i nuovi archivi delle chiavi esterne vengono creati in uno stato disconnesso. Prima di poter creare chiavi KMS nell'archivio delle chiavi esterne, devi [collegarlo](#) al relativo proxy. Per modificare le proprietà dell'archivio delle chiavi esterne, [modifica le impostazioni](#).

Argomenti

- [Assemblare i prerequisiti](#)
- [Crea un nuovo archivio di chiavi esterno](#)

Assemblare i prerequisiti

Prima di creare un archivio chiavi esterno, è necessario assemblare i componenti richiesti, incluso il [gestore di chiavi esterno](#) che verrà utilizzato per supportare l'archivio chiavi esterno e il [proxy dell'archivio chiavi esterno](#) che traduce AWS KMS le richieste in un formato comprensibile al gestore di chiavi esterno.

I seguenti componenti sono necessari per tutti gli archivi delle chiavi esterne. Oltre a questi elementi, devi fornire anche i componenti necessari per supportare l'[opzione di connettività proxy dell'archivio delle chiavi esterne](#) scelta.

 Tip

Il gestore delle chiavi esterne potrebbe includere alcuni di questi componenti oppure potrebbero essere configurati automaticamente. Per ulteriori informazioni, consulta la documentazione del gestore delle chiavi esterne.

[Se state creando l'archivio di chiavi esterno nella AWS KMS console, avete la possibilità di caricare un file di configurazione proxy basato su JSON che specifica il percorso URI del proxy le credenziali di autenticazione del proxy.](#) Alcuni proxy dell'archivio delle

chiavi esterne generano automaticamente questo file. Per maggiori dettagli, consulta la documentazione relativa al proxy dell'archivio delle chiavi esterne o al gestore delle chiavi esterne.

Gestore delle chiavi esterne

Ogni archivio delle chiavi esterne richiede almeno un'istanza [del gestore delle chiavi esterne](#). Può trattarsi di un modulo di sicurezza hardware (HSM) fisico o virtuale o di un software di gestione delle chiavi.

Sebbene sia possibile utilizzare un unico gestore delle chiavi, ti consigliamo di impiegare almeno due istanze correlate che condividono chiavi crittografiche per motivi di ridondanza. L'archivio delle chiavi esterne non richiede l'uso esclusivo del gestore delle chiavi esterne. Tuttavia, il gestore delle chiavi esterno deve avere la capacità di gestire la frequenza prevista delle richieste di crittografia e decrittografia provenienti dai AWS servizi che utilizzano le chiavi KMS nell'archivio di chiavi esterno per proteggere le risorse. Il gestore delle chiavi esterne deve essere configurato per gestire fino a 1.800 richieste al secondo e per rispondere a ciascuna richiesta entro il timeout di 250 millisecondi. Ti consigliamo di posizionare il gestore delle chiavi esterno vicino a un in Regione AWS modo che il tempo di andata e ritorno della rete (RTT) sia pari o inferiore a 35 millisecondi.

Se il proxy dell'archivio delle chiavi esterne lo consente, puoi modificare il gestore delle chiavi esterne associato al proxy, tuttavia il nuovo gestore deve essere un backup o uno snapshot con lo stesso materiale della chiave. Se la chiave esterna associata a una chiave KMS non è più disponibile per il proxy dell'archivio chiavi esterno, non è AWS KMS possibile decrittografare il testo cifrato crittografato con la chiave KMS.

Il gestore delle chiavi esterne deve essere accessibile al proxy dell'archivio delle chiavi esterne. Se la [GetHealthStatus](#) risposta del proxy riporta che tutte le istanze del gestore di chiavi esterno

lo sono `Unavailable`, tutti i tentativi di creare un archivio di chiavi esterno falliscono con un [XksProxyUriUnreachableException](#)

Proxy dell'archivio delle chiavi esterne

Devi specificare un [proxy dell'archivio delle chiavi esterne](#) (proxy XKS) conforme ai requisiti di progettazione della [Specifica API relativa al proxy dell'archivio delle chiavi esterne di AWS KMS](#). È possibile sviluppare o acquistare un proxy di archiviazione chiavi esterno oppure utilizzare un proxy di archiviazione chiavi esterno fornito o integrato nel gestore di chiavi esterno. AWS KMS consiglia di configurare il proxy di archiviazione delle chiavi esterno per gestire fino a 1800 richieste al secondo e rispondere entro il timeout di 250 millisecondi per ogni richiesta. Si consiglia di posizionare il gestore delle chiavi esterno vicino a un in Regione AWS modo che il tempo di andata e ritorno della rete (RTT) sia pari o inferiore a 35 millisecondi.

Puoi utilizzare un proxy dell'archivio delle chiavi esterne per più archivi, ma ogni archivio deve disporre di un endpoint URI univoco e di un percorso all'interno del proxy dell'archivio delle chiavi esterne per le relative richieste.

Se utilizzi la connettività del servizio endpoint VPC, puoi collocare il proxy dell'archivio delle chiavi esterne in Amazon VPC, ma ciò non è necessario. Puoi localizzare il proxy all'esterno AWS, ad esempio nel tuo data center privato, e utilizzare il servizio endpoint VPC solo per comunicare con il proxy.

Credenziali di autenticazione proxy

Per creare un archivio delle chiavi esterne, devi specificare le credenziali di autenticazione proxy dell'archivio (`XksProxyAuthenticationCredential`).

È necessario stabilire una [credenziale di autenticazione](#) (`XksProxyAuthenticationCredential`) per AWS KMS il proxy dell'archivio chiavi esterno. AWS KMS si autentica sul proxy firmando le relative richieste utilizzando il [processo Signature Version 4 \(SigV4\)](#) con la credenziale di autenticazione proxy del key store esterno. Puoi specificare le credenziali di autenticazione durante la creazione dell'archivio delle chiavi esterne e [modificarle](#) in qualsiasi momento. Se il proxy effettua la rotazione delle credenziali, assicurati di aggiornare i valori delle credenziali per l'archivio delle chiavi esterne.

Le credenziali di autenticazione proxy sono composte da due parti. Per l'archivio delle chiavi esterne, devi fornire entrambe.

- ID chiave di accesso: identifica la chiave di accesso segreta. Puoi fornire questo ID come testo non crittografato.

- Chiave di accesso segreta: la parte segreta della credenziale. AWS KMS crittografa la chiave di accesso segreta nella credenziale prima di archivarla.

Le credenziali SigV4 AWS KMS utilizzate per firmare le richieste al proxy dell'archivio chiavi esterno non sono correlate alle credenziali SigV4 associate ai principali degli account. AWS Identity and Access Management AWS Non riutilizzare le credenziali SigV4 IAM per il proxy dell'archivio delle chiavi esterne.

Connettività proxy

Per creare un archivio delle chiavi esterne, devi specificare l'opzione di connettività proxy (XksProxyConnectivity).

AWS KMS può comunicare con il tuo proxy di archiviazione delle chiavi esterno utilizzando un [endpoint pubblico](#) o un servizio endpoint [Amazon Virtual Private Cloud \(Amazon VPC\)](#). Sebbene un endpoint pubblico sia più semplice da configurare e gestire, potrebbe non soddisfare i requisiti di sicurezza per ogni installazione. Se scegli l'opzione di connettività del servizio endpoint Amazon VPC, devi creare e gestire i componenti richiesti, tra cui un Amazon VPC con almeno due sottoreti in due diverse zone di disponibilità, un servizio endpoint VPC con un Network Load Balancer e un gruppo di destinazione e un nome DNS privato per il servizio endpoint VPC.

Puoi [modificare l'opzione di connettività proxy](#) dell'archivio delle chiavi esterne. Tuttavia, devi assicurarti che il materiale della chiave associato alle chiavi KMS sia sempre disponibile nell'archivio. Altrimenti, AWS KMS non può decrittografare alcun testo cifrato crittografato con tali chiavi KMS.

Per informazioni relative all'opzione di connettività proxy migliore per l'archivio delle chiavi esterne, consulta [Scegli un'opzione di connettività proxy per l'archivio chiavi esterno](#). Per informazioni sulla creazione e sulla configurazione della connettività del servizio endpoint VPC, consulta [Configurazione della connettività del servizio endpoint VPC](#).

Endpoint dell'URI proxy

Per creare un archivio di chiavi esterno, è necessario specificare l'endpoint (XksProxyUriEndpoint) da AWS KMS utilizzare per inviare le richieste al proxy dell'archivio chiavi esterno.

Il protocollo deve essere HTTPS. AWS KMS comunica IPv4 sulla porta 443. Non specificare la porta nel valore dell'endpoint URI proxy.

- [Connettività dell'endpoint pubblico](#): specifica l'endpoint disponibile per il proxy dell'archivio delle chiavi esterne. Tale endpoint deve essere raggiungibile prima di creare l'archivio delle chiavi esterne.
- [Connettività del servizio endpoint VPC](#): specifica `https://` seguito dal nome DNS privato del servizio endpoint VPC.

Il certificato del server TLS configurato sul proxy dell'archivio delle chiavi esterne deve corrispondere al nome di dominio nell'endpoint URI proxy ed essere emesso da un'autorità di certificazione supportata per gli archivi delle chiavi esterne. Per un elenco, consulta [Autorità di certificazione attendibili](#). L'autorità di certificazione richiederà una prova della proprietà del dominio prima di emettere il certificato TLS.

Il nome comune del soggetto (CN) sul certificato TLS deve corrispondere al nome DNS privato. Ad esempio, se il nome DNS privato è `myproxy-private.xks.example.com`, il CN sul certificato TLS deve essere `myproxy-private.xks.example.com` o `*.xks.example.com`.

Puoi [modificare l'endpoint dell'URI proxy](#), tuttavia devi assicurarti che il proxy dell'archivio delle chiavi esterne abbia accesso al materiale della chiave associato alle chiavi KMS nell'archivio. Altrimenti, AWS KMS non può decrittografare alcun testo cifrato crittografato con quelle chiavi KMS.

Requisiti di unicità

- Il valore combinato di endpoint dell'URI proxy (`XksProxyUriEndpoint`) e percorso URI proxy (`XksProxyUriPath`) deve essere univoco nel tuo Account AWS e nella regione.
- Gli archivi delle chiavi esterne con connettività dell'endpoint pubblico possono condividere lo stesso endpoint URI proxy, a condizione che abbiano valori diversi per il percorso URI proxy.
- Un archivio di chiavi esterno con connettività endpoint pubblica non può utilizzare lo stesso valore di endpoint URI proxy di qualsiasi archivio di chiavi esterno con connettività dei servizi endpoint VPC nello stesso archivio Regione AWS, anche se gli archivi di chiavi si trovano in archivi diversi. Account AWS
- Ogni archivio delle chiavi esterne con connettività all'endpoint VPC deve avere il proprio nome DNS privato. L'endpoint URI proxy (nome DNS privato) deve essere univoco nella regione and. Account AWS

Percorso URI proxy

Per creare un archivio di chiavi esterno, è necessario specificare il percorso di base del proxy di archiviazione delle chiavi esterno fino al proxy [richiesto](#). API Il valore deve iniziare con / e deve terminare con /kms/xks/v1, dove v1 rappresenta la versione dell' AWS KMS API per il proxy dell'archivio di chiavi esterno. Questo percorso può includere un prefisso facoltativo tra gli elementi richiesti, ad esempio /example-prefix/kms/xks/v1. Per trovare questo valore, consulta la documentazione del proxy dell'archivio delle chiavi esterne.

AWS KMS invia le richieste proxy all'indirizzo specificato dalla concatenazione dell'endpoint URI del proxy e del percorso URI del proxy. Ad esempio, se l'endpoint URI del proxy è `https://myproxy.xks.example.com` e il percorso URI del proxy è `/kms/xks/v1`, AWS KMS invia le relative richieste API proxy a `https://myproxy.xks.example.com/kms/xks/v1`

Puoi [modificare il percorso URI proxy](#), tuttavia devi assicurarti che il proxy dell'archivio delle chiavi esterne abbia accesso al materiale della chiave associato alle chiavi KMS nell'archivio. In caso contrario, AWS KMS non è possibile decrittografare alcun testo cifrato crittografato con tali chiavi KMS.

Requisiti di unicità

- Il valore combinato di endpoint dell'URI proxy (`XksProxyUriEndpoint`) e percorso URI proxy (`XksProxyUriPath`) deve essere univoco nel tuo Account AWS e nella regione.

Servizio endpoint VPC

Specifica il nome del servizio endpoint Amazon VPC utilizzato per comunicare con il proxy dell'archivio delle chiavi esterne. Questo componente è richiesto solo per gli archivi delle chiavi esterne che utilizzano la connettività del servizio endpoint VPC. Per informazioni sull'impostazione e sulla configurazione del servizio endpoint VPC per un archivio delle chiavi esterne, consulta [Configurazione della connettività del servizio endpoint VPC](#).

Il servizio endpoint VPC deve avere le seguenti proprietà:

- Il servizio endpoint VPC può risiedere nello stesso o in un altro Account AWS archivio di chiavi esterno.
 - Il servizio endpoint VPC deve risiedere nello stesso archivio delle chiavi Regione AWS esterno.
 - Dovrai fornire l' Account AWS ID del servizio endpoint VPC se si trova in un altro. Account AWS

- Deve avere un Network Load Balancer (NLB) connesso ad almeno due sottoreti, ognuna in una zona di disponibilità diversa.
- L'elenco dei principali consentiti per il servizio endpoint VPC deve includere AWS KMS l'entità del servizio per la regione `cks.kms.<region>.amazonaws.com`; ad esempio `cks.kms.us-east-1.amazonaws.com`
- Se il tuo servizio endpoint Amazon VPC è di proprietà di una persona Account AWS diversa da quella Account AWS proprietaria dell'archivio di chiavi esterne (XKS), dovrai anche consentire a XKS l'accesso al servizio endpoint VPC. A tale scopo, [consenti l' Account AWS ID XKS come principale](#) per il servizio endpoint Amazon VPC.
- Non deve richiedere l'accettazione delle richieste di connessione.
- Deve avere un nome DNS privato all'interno di un dominio pubblico di livello superiore. Ad esempio, potresti avere un nome DNS privato `myproxy-private.xks.example.com` nel dominio `xks.example.com` pubblico.

Il nome DNS privato per un archivio delle chiavi esterne con connettività del servizio endpoint VPC deve essere univoco nella Regione AWS.

- Lo [stato di verifica del dominio](#) per il dominio del nome DNS privato deve essere `verified`.
- Il certificato del server TLS configurato sul proxy dell'archivio delle chiavi esterne deve indicare il nome host DNS privato in cui l'endpoint è raggiungibile.

Requisiti di unicità

- Gli archivi delle chiavi esterne con connettività all'endpoint VPC possono condividere un Amazon VPC, ma ogni archivio deve avere il proprio servizio endpoint VPC e il proprio nome DNS privato.

File di configurazione proxy

Un file di configurazione proxy è un file facoltativo basato su JSON che contiene valori per le proprietà del [percorso URI proxy](#) e delle [credenziali di autenticazione proxy](#) dell'archivio delle chiavi esterne. Quando crei o [modifichi un archivio delle chiavi esterne](#) nella console AWS KMS, puoi caricare un file di configurazione proxy per fornire i valori di configurazione dell'archivio. L'utilizzo di questo file consente di evitare errori correlati alle operazioni di digitazione e di copia e incolla, garantendo che i valori nell'archivio delle chiavi esterne corrispondano ai valori del relativo proxy.

I file di configurazione proxy vengono generati dal proxy dell'archivio delle chiavi esterne. Per scoprire se il proxy dell'archivio delle chiavi esterne offre un file di configurazione proxy, consulta la relativa documentazione.

Di seguito è riportato un esempio di un file di configurazione proxy ben formato con valori fittizi.

```
{
  "XksProxyUriPath": "/example-prefix/kms/xks/v1",
  "XksProxyAuthenticationCredential": {
    "AccessKeyId": "ABCDE12345670EXAMPLE",
    "RawSecretAccessKey": "0000EXAMPLEFA5FT0mCc3DrGUe2sti527BitkQ0Zr9M09+vE="
  }
}
```

Puoi caricare un file di configurazione del proxy solo quando crei o modifichi un archivio di chiavi esterno nella console. AWS KMS Non è possibile utilizzarlo con le [UpdateCustomKeyStore](#) operazioni [CreateCustomKeyStore](#), ma è possibile utilizzare i valori nel file di configurazione del proxy per assicurarsi che i valori dei parametri siano corretti.

Crea un nuovo archivio di chiavi esterno

Dopo aver assemblato i prerequisiti necessari, è possibile creare un nuovo archivio di chiavi esterno nella AWS KMS console o utilizzando l'[CreateCustomKeyStore](#) operazione.

Utilizzo della console AWS KMS

Prima di creare un archivio di chiavi esterno, [scegli il tipo di connettività proxy](#) e assicurati di aver creato e configurato tutti i [componenti richiesti](#). Se hai bisogno di aiuto per trovare uno dei valori richiesti, consulta la documentazione del proxy dell'archivio delle chiavi esterne o del software di gestione delle chiavi.

Note

Quando crei un archivio di chiavi esterno in Console di gestione AWS, puoi caricare un file di configurazione proxy basato su JSON con i valori per il [percorso URI del proxy](#) e le credenziali di [autenticazione del proxy](#). Alcuni proxy generano automaticamente questo file, ma non è obbligatorio.

1. [Accedi a Console di gestione AWS e apri la console AWS Key Management Service \(AWS KMS\) su /kms. https://console.aws.amazon.com](#)

2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel pannello di navigazione, scegli Custom key stores (Archivi delle chiavi personalizzate), External key stores (Archivi delle chiavi esterne).
4. Scegli Create external key store (Crea archivio delle chiavi esterne).
5. Immetti un nome descrittivo per l'archivio delle chiavi esterne. Il nome deve essere univoco tra tutti gli archivi delle chiavi esterne nel tuo account.

 Important

Non includere informazioni riservate o sensibili in questo campo. Questo campo può essere visualizzato in testo semplice nei log e in altri output. CloudTrail

6. Scegli il tipo di [connettività proxy](#).

La scelta della connettività proxy determina i [componenti necessari](#) per il proxy dell'archivio delle chiavi esterne. Per assistenza durante la scelta, consulta [Scegli un'opzione di connettività proxy per l'archivio chiavi esterno](#).

- a. Seleziona Servizio endpoint VPC per più account se il tuo servizio endpoint VPC risiede in un altro. Account AWS Quindi inserisci l' Account AWS ID del proprietario dell'endpoint VPC nel campo ID account proprietario del servizio endpoint VPC.
- b. Scegli o inserisci il nome del [servizio endpoint VPC](#) per questo archivio delle chiavi esterne. Questo passaggio viene visualizzato solo quando il tipo di connettività proxy dell'archivio chiavi esterno è il servizio endpoint VPC.

Il servizio endpoint VPC e il suo VPCs devono soddisfare i requisiti per un archivio di chiavi esterno. Per informazioni dettagliate, vedi [the section called “Assemblare i prerequisiti”](#).

7. Scegli o inserisci il nome del [servizio endpoint VPC](#) per questo archivio delle chiavi esterne. Questo passaggio viene visualizzato solo quando il tipo di connettività proxy è servizio endpoint VPC.

Il servizio endpoint VPC e il suo VPCs devono soddisfare i requisiti per un archivio di chiavi esterno. Per informazioni dettagliate, vedi [the section called “Assemblare i prerequisiti”](#).

8. Inserisci l'[endpoint URI proxy](#). Il protocollo deve essere HTTPS. AWS KMS comunica IPv4 sulla porta 443. Non specificare la porta nel valore dell'endpoint URI proxy.

Se AWS KMS riconosce il servizio endpoint VPC specificato nel passaggio precedente, completa questo campo per te.

Per la connettività dell'endpoint pubblico, inserisci un URI endpoint disponibile pubblicamente. Per la connettività dell'endpoint VPC, inserisci `https://` seguito dal nome DNS privato del servizio endpoint VPC.

9. Per inserire i valori relativi al prefisso del [percorso URI proxy](#) e le [credenziali di autenticazione proxy](#), carica un file di configurazione proxy o inserisci i valori manualmente.
 - Se disponi di un [file di configurazione proxy](#) facoltativo che contiene i valori del [percorso URI proxy](#) e delle [credenziali di autenticazione proxy](#), scegli Upload configuration file (Carica file di configurazione). Segui le istruzioni per caricare il file.

Quando il file viene caricato, la console visualizza i valori del file in campi modificabili. Puoi modificare i valori in questo momento o [modificarli](#) dopo la creazione dell'archivio delle chiavi esterne.

Per visualizzare il valore della chiave di accesso segreta, scegli Show secret access key (Mostra chiave di accesso segreta).

- Se non hai a disposizione un file di configurazione proxy, puoi inserire manualmente i valori del percorso URI proxy e delle credenziali di autenticazione proxy.
 - a. Se non disponi di un file di configurazione proxy, puoi inserire l'URI proxy manualmente. La console fornisce il valore/1 richiesto. kms/xks/v

Se il [percorso URI proxy](#) comprende un prefisso facoltativo, ad esempio `example-prefix` in `/example-prefix/kms/xks/v1`, inseriscilo nel campo Proxy URI path prefix (Prefisso del percorso URI proxy). In caso contrario, lascia vuoto il campo.

- b. Se non disponi di un file di configurazione proxy, puoi inserire le [credenziali di autenticazione proxy](#) manualmente. Sono necessari sia l'ID chiave di accesso che la chiave di accesso segreta.
 - In Proxy credential: Access key ID (Credenziali proxy: ID chiave di accesso), inserisci l'ID chiave di accesso delle credenziali di autenticazione proxy. L'ID della chiave di accesso identifica la chiave di accesso segreta.
 - In Proxy credential: Secret access key (Credenziali proxy: chiave di accesso segreta), inserisci la chiave di accesso segreta delle credenziali di autenticazione proxy.

Per visualizzare il valore della chiave di accesso segreta, scegli Show secret access key (Mostra chiave di accesso segreta).

Questa procedura non imposta o modifica le credenziali di autenticazione stabilite sul proxy dell'archivio delle chiavi esterne, ma associa semplicemente tali valori all'archivio. Per informazioni sull'impostazione, la modifica e la rotazione delle credenziali di autenticazione proxy, consulta la documentazione del proxy dell'archivio delle chiavi esterne o del software di gestione delle chiavi.

Se le credenziali di autenticazione proxy cambiano, [modifica l'impostazione delle credenziali](#) per l'archivio delle chiavi esterne.

10. Scegli Create external key store (Crea archivio delle chiavi esterne).

Quando la procedura ha esito positivo, il nuovo archivio delle chiavi esterne viene visualizzato nell'elenco degli archivi delle chiavi esterne dell'account e della regione. Se ha esito negativo, viene visualizzato un messaggio di errore che descrive il problema e fornisce istruzioni su come risolverlo. Per ulteriori informazioni, consulta [CreateKey errori per la chiave esterna](#).

Successivo: i nuovi archivi delle chiavi esterne non sono connessi automaticamente. Prima di poter creare AWS KMS keys nell'archivio chiavi esterno, è necessario [connettere l'archivio chiavi esterno](#) al relativo proxy dell'archivio chiavi esterno.

Utilizzo dell' AWS KMS API

È possibile utilizzare l'[CreateCustomKeyStore](#) operazione per creare un nuovo archivio di chiavi esterno. Per assistenza nell'individuazione dei valori per i parametri richiesti, consulta la documentazione del proxy dell'archivio delle chiavi esterne o del software di gestione delle chiavi.

Tip

Non puoi caricare un [file di configurazione proxy](#) quando utilizzi l'operazione CreateCustomKeyStore. Tuttavia, puoi utilizzare i valori presenti nel file di configurazione proxy per assicurarti che i valori dei parametri siano corretti.

Per creare un archivio delle chiavi esterne, l'operazione CreateCustomKeyStore richiede i valori di parametro seguenti.

- **CustomKeyStoreName**: un nome descrittivo per l'archivio delle chiavi esterne univoco nell'account.

 Important

Non includere informazioni riservate o sensibili in questo campo. Questo campo può essere visualizzato in testo semplice nei CloudTrail log e in altri output.

- **CustomKeyStoreType**: specifica `EXTERNAL_KEY_STORE`.
- [**XksProxyConnectivity**](#): specifica `PUBLIC_ENDPOINT` o `VPC_ENDPOINT_SERVICE`.
- [**XksProxyAuthenticationCredential**](#): specifica sia l'ID chiave di accesso che la chiave di accesso segreta.
- [**XksProxyUriEndpoint**](#): l'endpoint utilizzato da AWS KMS per comunicare con il proxy dell'archivio delle chiavi esterne.
- [**XksProxyUriPath**](#)— Il percorso all'interno del proxy verso il proxy. APIs
- [**XksProxyVpcEndpointServiceName**](#): obbligatorio solo quando il valore di `XksProxyConnectivity` è `VPC_ENDPOINT_SERVICE`.

 Note

Se utilizzate la AWS CLI versione 1.0, eseguite il comando seguente prima di specificare un parametro con un valore HTTP o HTTPS, ad esempio il `XksProxyUriEndpoint` parametro.

```
aws configure set cli_follow_urlparam false
```

In caso contrario, la AWS CLI versione 1.0 sostituisce il valore del parametro con il contenuto trovato in quell'indirizzo URI, causando il seguente errore:

```
Error parsing parameter '--xks-proxy-uri-endpoint': Unable to retrieve  
https:// : received non 200 status code of 404
```

Gli esempi seguenti utilizzano valori fittizi. Prima di eseguire il comando, sostituiteli con valori validi per l'archivio delle chiavi esterne.

Crea un archivio delle chiavi esterne con connettività dell'endpoint pubblico.

```
$ aws kms create-custom-key-store
  --custom-key-store-name ExampleExternalKeyStorePublic \
  --custom-key-store-type EXTERNAL_KEY_STORE \
  --xks-proxy-connectivity PUBLIC_ENDPOINT \
  --xks-proxy-uri-endpoint https://myproxy.xks.example.com \
  --xks-proxy-uri-path /kms/xks/v1 \
  --xks-proxy-authentication-credential
AccessKeyId=<value>,RawSecretAccessKey=<value>
```

Crea un archivio delle chiavi esterne con connettività del servizio endpoint VPC.

```
$ aws kms create-custom-key-store
  --custom-key-store-name ExampleExternalKeyStoreVPC \
  --custom-key-store-type EXTERNAL_KEY_STORE \
  --xks-proxy-connectivity VPC_ENDPOINT_SERVICE \
  --xks-proxy-vpc-endpoint-service-name com.amazonaws.vpce.us-east-1.vpce-svc-
example \
  --xks-proxy-uri-endpoint https://myproxy-private.xks.example.com \
  --xks-proxy-uri-path /kms/xks/v1 \
  --xks-proxy-authentication-credential
AccessKeyId=<value>,RawSecretAccessKey=<value>
```

Se l'operazione riesce, `CreateCustomKeyStore` restituisce l'ID store chiavi personalizzate, come illustrato nel seguente esempio di risposta.

```
{
  "CustomKeyId": cks-1234567890abcdef0
}
```

Se l'operazione ha esito negativo, correggi l'errore indicato dall'eccezione e riprova. Per ulteriori informazioni, consulta [Risoluzione dei problemi relativi all'archivio delle chiavi esterne](#).

Successivo: Per utilizzare l'archivio delle chiavi esterne, [connettilo al relativo proxy dell'archivio delle chiavi esterne](#).

Modifica delle proprietà dell'archivio chiavi esterno

Puoi modificare le proprietà selezionate di un archivio delle chiavi esterne esistente.

Quando l'archivio delle chiavi esterne è connesso o disconnesso, puoi modificare solo alcune proprietà. Per modificare le altre proprietà, [disconnetti l'archivio delle chiavi esterne](#) dal relativo proxy.

Lo [stato di connessione](#) dell'archivio delle chiavi esterne deve essere DISCONNECTED. Quando un archivio delle chiavi esterne è disconnesso, puoi gestire l'archivio e le relative chiavi KMS, ma non puoi creare o utilizzare le chiavi KMS nell'archivio delle chiavi esterne. Per trovare lo [stato di connessione](#) del tuo archivio di chiavi esterno, utilizza l'[DescribeCustomKeyStores](#) operazione o consulta la sezione Configurazione generale nella pagina dei dettagli dell'archivio di chiavi esterno.

Prima di aggiornare le proprietà dell'archivio di chiavi esterno, AWS KMS invia una [GetHealthStatus](#) richiesta al proxy dell'archivio chiavi esterno utilizzando i nuovi valori. Se la richiesta ha esito positivo, indica che è possibile connettersi e autenticarsi a un proxy dell'archivio delle chiavi esterne con i valori delle proprietà aggiornati. Se la richiesta non riesce, l'operazione di modifica ha esito negativo con un'eccezione che identifica l'errore.

Al termine dell'operazione di modifica, i valori delle proprietà aggiornati per l'archivio di chiavi esterno vengono visualizzati nella AWS KMS console e nella [DescribeCustomKeyStores](#) risposta. Tuttavia, possono essere necessari fino a cinque minuti affinché le modifiche diventino effettive.

[Se modifichi l'archivio di chiavi esterno nella AWS KMS console, hai la possibilità di caricare un file di configurazione del proxy basato su JSON che specifica il percorso dell'URI del proxye le credenziali di autenticazione del proxy.](#) Alcuni proxy dell'archivio delle chiavi esterne generano automaticamente questo file. Per maggiori dettagli, consulta la documentazione relativa al proxy dell'archivio delle chiavi esterne o al gestore delle chiavi esterne.

Warning

I valori delle proprietà aggiornati devono connettere l'archivio delle chiavi esterne a un proxy per lo stesso gestore delle chiavi esterne dei valori precedenti o per un backup o uno snapshot del gestore con le stesse chiavi crittografiche. Se l'archivio delle chiavi esterne perde definitivamente l'accesso alle chiavi esterne associate alle relative chiavi KMS, il testo criptato crittografato con tali chiavi esterne è irrecuperabile. In particolare, la modifica della connettività proxy di un archivio di chiavi esterno può AWS KMS impedire l'accesso alle chiavi esterne.

Tip

Alcuni gestori delle chiavi esterne offrono un metodo più semplice per modificare le proprietà dell'archivio delle chiavi esterne. Per ulteriori informazioni, consulta la documentazione del gestore delle chiavi esterne.

Puoi modificare le seguenti proprietà di un archivio delle chiavi esterne.

Proprietà modificabili dell'archivio delle chiavi esterne	Qualsiasi stato di connessione	Richiede lo stato Disconnesso
Il nome dello store delle chiavi personalizzate Un nome descrittivo per l'archivio delle chiavi personalizzate.  Important Non includere informazioni riservate o sensibili in questo campo. Questo campo può essere visualizzato in testo semplice nei CloudTrail log e in altri output.	✓	
Credenziali di autenticazione proxy () XksProxyAuthenticationCredential Devi specificare sia l'ID chiave di accesso che la chiave di accesso segreta, anche se modifichi un solo elemento.	✓	
Percorso URI del proxy () XksProxyUriPath	✓	
Connettività proxy (XksProxyConnectivity) Devi inoltre aggiornare l'endpoint URI proxy. Se stai passando alla connettività del servizio endpoint VPC, devi specificare un nome del servizio endpoint VPC del proxy.		✓
Endpoint URI proxy () XksProxyUriEndpoint Se modifichi l'URI endpoint proxy, potrebbe anche essere necessario modificare il certificato TLS associato.		✓

Proprietà modificabili dell'archivio delle chiavi esterne	Qualsiasi stato di connessione	Richiede lo stato Disconnesso
Nome del servizio endpoint VPC proxy () XksProxyVpcEndpointServiceName Questo campo è obbligatorio per la connettività del servizio endpoint VPC		
Titolare del servizio endpoint VPC () XksProxyVpcEndpointServiceOwner Questo campo è obbligatorio per la connettività del servizio endpoint VPC		

Modifica le proprietà del tuo archivio di chiavi esterno

È possibile modificare le proprietà dell'archivio di chiavi esterno nella AWS KMS console o utilizzando l'[UpdateCustomKeyStore](#) operazione.

Utilizzo della AWS KMS console

Quando si modifica un archivio di chiavi, è possibile modificare qualsiasi valore modificabile. Alcune modifiche richiedono la disconnessione dell'archivio delle chiavi esterne dal relativo proxy.

Se stai modificando il percorso URI proxy o le credenziali di autenticazione proxy, puoi inserire i nuovi valori o caricare un [file di configurazione proxy](#) dell'archivio delle chiavi esterne che includa i nuovi valori.

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) su <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel pannello di navigazione, scegli Custom key stores (Archivi delle chiavi personalizzate), External key stores (Archivi delle chiavi esterne).
4. Scegli l'archivio delle chiavi che desideri modificare.

- Se necessario, disconnetti l'archivio delle chiavi esterne dal relativo proxy. Dal menu Key store actions (Operazioni per l'archivio delle chiavi), scegli Disconnect (Disconnetti).
5. Dal menu Key store actions (Operazioni per l'archivio delle chiavi), scegli Edit (Modifica).
 6. Modifica una o più proprietà configurabili dell'archivio delle chiavi esterne. Puoi inoltre caricare un [file di configurazione proxy](#) dell'archivio delle chiavi esterne con i valori per il percorso URI proxy e le credenziali di autenticazione proxy. Puoi utilizzare tale file di configurazione proxy anche se alcuni valori specificati nel file non sono stati modificati.
 7. Scegli Update external key store (Aggiornamento dell'archivio delle chiavi esterne).
 8. Esamina l'avviso e, se decidi di continuare, confermalo, quindi scegli Update external key store (Aggiornamento dell'archivio delle chiavi esterne).

Se la procedura ha esito positivo, un messaggio descrive le proprietà modificate. Se ha esito negativo, viene visualizzato un messaggio di errore che descrive il problema e fornisce istruzioni su come risolverlo.

9. Se necessario, connetti nuovamente l'archivio delle chiavi esterne. Dal menu Key store actions (Operazioni per l'archivio delle chiavi), scegli Connect (Connetti).

Puoi lasciarlo disconnesso, ma in questo stato, non puoi creare o utilizzare le chiavi KMS nell'archivio delle chiavi esterne per [operazioni di crittografia](#).

Utilizzo dell' AWS KMS API

Per modificare le proprietà di un archivio di chiavi esterno, utilizzare l'[UpdateCustomKeyStore](#) operazione. Puoi modificare più proprietà di un archivio delle chiavi esterne con la stessa operazione. Se l'operazione ha esito positivo, AWS KMS restituisce una risposta HTTP 200 e un oggetto JSON senza proprietà.

Utilizza il parametro CustomKeyStoreId per identificare l'archivio delle chiavi esterne. Utilizza gli altri parametri per modificare le proprietà. Non puoi utilizzare un [file di configurazione proxy](#) con l'operazione UpdateCustomKeyStore. Il file di configurazione del proxy è supportato solo dalla AWS KMS console. Tuttavia, puoi utilizzare il file di configurazione proxy per determinare i valori dei parametri corretti per il proxy dell'archivio delle chiavi esterne.

Gli esempi in questa sezione utilizzano [AWS Command Line Interface \(AWS CLI\)](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

Prima di iniziare, [se necessario](#), [disconnetti l'archivio delle chiavi esterne](#) dal relativo proxy. Dopo l'aggiornamento, se necessario, [connetti nuovamente l'archivio](#) al proxy dell'archivio delle chiavi esterne. Puoi lasciare l'archivio delle chiavi esterne disconnesso, ma devi connetterlo per poter creare nuove chiavi KMS o utilizzare le chiavi KMS esistenti nell'archivio delle chiavi per operazioni di crittografia.

Note

Se utilizzi la AWS CLI versione 1.0, esegui il comando seguente prima di specificare un parametro con un valore HTTP o HTTPS, ad esempio il `XksProxyUriEndpoint` parametro.

```
aws configure set cli_follow_urlparam false
```

In caso contrario, la AWS CLI versione 1.0 sostituisce il valore del parametro con il contenuto trovato in quell'indirizzo URI, causando il seguente errore:

```
Error parsing parameter '--xks-proxy-uri-endpoint': Unable to retrieve  
https:// : received non 200 status code of 404
```

Modifica del nome dell'archivio delle chiavi esterne

Il primo esempio utilizza l'[UpdateCustomKeyStore](#) operazione per modificare il nome descrittivo dell'archivio di chiavi esterno in `XksKeyStore`. Il comando utilizza il parametro `CustomKeyStoreId` per identificare lo store delle chiavi personalizzate e `CustomKeyStoreName` per specificarne il nuovo nome. Sostituisci tutti i valori di esempio con i valori effettivi per l'archivio delle chiavi esterne.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --new-  
custom-key-store-name XksKeyStore
```

Modifica delle credenziali di autenticazione proxy

L'esempio seguente aggiorna le credenziali di autenticazione proxy utilizzate da AWS KMS per l'autenticazione nel proxy dell'archivio delle chiavi esterne. Puoi utilizzare un comando simile a questo per aggiornare le credenziali se vengono ruotate sul proxy.

Aggiorna prima le credenziali nel proxy dell'archivio delle chiavi esterne. Quindi, utilizza questa funzione per segnalare la modifica ad AWS KMS. (Il proxy supporterà brevemente sia la vecchia che la nuova credenziale, in modo da avere il tempo di aggiornare le credenziali). AWS KMS

Nelle credenziali devi specificare sia l'ID chiave di accesso che la chiave di accesso segreta, anche se viene modificato un solo valore.

I primi due comandi impostano le variabili per contenere i valori delle credenziali. Le operazioni `UpdateCustomKeyStore` utilizzano il parametro `CustomKeyId` per identificare l'archivio delle chiavi esterne. Utilizza il parametro `XksProxyAuthenticationCredential` con i relativi campi `AccessKeyId` e `RawSecretAccessKey` per specificare le nuove credenziali. Sostituisci tutti i valori di esempio con i valori effettivi per l'archivio delle chiavi esterne.

```
$ accessKeyId=access key id
$ secretAccessKey=secret access key

$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
  --xks-proxy-authentication-credential \
    AccessKeyId=$accessKeyId,RawSecretAccessKey=$secretAccessKey
```

Modifica del percorso URI proxy

L'esempio seguente aggiorna il percorso URI proxy (`XksProxyUriPath`). La combinazione dell'endpoint URI del proxy e del percorso URI del proxy deve essere unica nella Account AWS regione `and`. Sostituisci tutti i valori di esempio con i valori effettivi per l'archivio delle chiavi esterne.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
  --xks-proxy-uri-path /kms/xks/v1
```

Passaggio alla connettività del servizio endpoint VPC

L'esempio seguente utilizza l'[UpdateCustomKeyStore](#) operazione per modificare il tipo di connettività del proxy dell'archivio chiavi esterno in `VPC_ENDPOINT_SERVICE`. Per apportare questa modifica, devi specificare i valori richiesti per la connettività del servizio endpoint VPC, incluso il nome del servizio endpoint VPC (`XksProxyVpcEndpointServiceName`) e un valore dell'endpoint URI proxy (`XksProxyUriEndpoint`) che includa il nome DNS privato per il servizio endpoint VPC. Sostituisci tutti i valori di esempio con i valori effettivi per l'archivio delle chiavi esterne.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
  --xks-proxy-connectivity "VPC_ENDPOINT_SERVICE" \
  --xks-proxy-uri-endpoint https://myproxy-private.xks.example.com \
  --xks-proxy-vpc-endpoint-service-name com.amazonaws.vpce.us-east-1.vpce-
svc-example
```

Passaggio alla connettività dell'endpoint pubblico

L'esempio seguente modifica il tipo di connettività del proxy dell'archivio delle chiavi esterne in `PUBLIC_ENDPOINT`. Quando apporti questa modifica, devi aggiornare il valore dell'endpoint URI proxy (`XksProxyUriEndpoint`). Sostituisci tutti i valori di esempio con i valori effettivi per l'archivio delle chiavi esterne.

Note

La connettività dell'endpoint VPC offre una maggiore sicurezza rispetto alla connettività dell'endpoint pubblico. Prima di passare alla connettività dell'endpoint pubblico, prendi in considerazione altre opzioni, tra cui il posizionamento del proxy dell'archivio delle chiavi esterne on-premise e l'utilizzo del VPC solo per la comunicazione.

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
  --xks-proxy-connectivity "PUBLIC_ENDPOINT" \
  --xks-proxy-uri-endpoint https://myproxy.xks.example.com
```

Visualizza gli archivi di chiavi esterni

È possibile visualizzare gli archivi di chiavi esterni in ogni account e regione utilizzando la AWS KMS console o utilizzando l'[DescribeCustomKeyStores](#) operazione.

Quando visualizzi un archivio delle chiavi esterne, hai accesso alle seguenti informazioni:

- Informazioni di base sull'archivio delle chiavi, tra cui nome descrittivo, ID, tipo di archivio e data di creazione.
- Informazioni di configurazione per il [proxy dell'archivio delle chiavi esterne](#), tra cui [tipo di connettività](#), [endpoint URI proxy](#), [percorso URI proxy](#) e [ID della chiave di accesso](#) delle [credenziali di autenticazione proxy](#) correnti.
- Se il proxy dell'archivio delle chiavi esterne utilizza la [connettività del servizio endpoint VPC](#), la console visualizza il nome del servizio endpoint VPC.
- Lo [stato di connessione](#) corrente.

 Note

Il valore Disconnected (Disconnesso) dello stato di connessione indica che l'archivio delle chiavi esterne non è mai stato connesso oppure che è stato intenzionalmente disconnesso dal relativo proxy. Se tuttavia i tentativi di utilizzare una chiave KMS in un archivio delle chiavi esterne connesso non riescono, è possibile che vi sia un problema con l'archivio o con il cluster. Per assistenza, consulta [Errori di connessione all'archivio delle chiavi esterne](#).

- Una sezione di [monitoraggio](#) con grafici delle [CloudWatch metriche di Amazon](#) progettati per aiutarti a rilevare e risolvere problemi con il tuo archivio di chiavi esterno. Per informazioni sull'interpretazione dei grafici, sul loro utilizzo nella pianificazione e risoluzione dei problemi e sulla creazione di CloudWatch allarmi in base alle metriche dei grafici, consulta. [Monitora gli archivi di chiavi esterni](#)

Proprietà dell'archivio delle chiavi esterne

Le seguenti proprietà di un archivio di chiavi esterno sono visibili nella console e nella AWS KMS risposta. [DescribeCustomKeyStores](#)

Proprietà dell'archivio delle chiavi personalizzate

I seguenti valori vengono visualizzati nella sezione Configurazione generale della pagina di dettaglio di ogni archivio di chiavi personalizzato. Queste proprietà si applicano a tutti gli archivi di chiavi personalizzati, inclusi gli archivi di AWS CloudHSM chiavi e gli archivi di chiavi esterni.

ID dello store delle chiavi personalizzate

Un ID univoco che viene AWS KMS assegnato all'archivio chiavi personalizzato.

Il nome dello store delle chiavi personalizzate

Nome descrittivo assegnato all'archivio delle chiavi personalizzate durante la sua creazione. Puoi modificare questo valore in qualsiasi momento.

Tipo di archivio delle chiavi personalizzate

Il tipo di archivio delle chiavi personalizzate. I valori validi sono AWS CloudHSM (AWS_CLOUDHSM) o External key store (Archivio delle chiavi esterne) (EXTERNAL_KEY_STORE). Il tipo di archivio non può essere modificato dopo la creazione.

Data di creazione

La data in cui è stato creato l'archivio delle chiavi personalizzate. Questa data viene visualizzata nell'ora locale per la Regione AWS.

Stato connessione

Indica se l'archivio delle chiavi personalizzate è connesso al relativo archivio del materiale della chiave. Lo stato della connessione è DISCONNECTED solo se l'archivio delle chiavi personalizzate non è mai stato collegato al relativo archivio del materiale della chiave o se è stato disconnesso intenzionalmente. Per informazioni dettagliate, vedi [the section called “Stato connessione”](#).

Proprietà di configurazione dell'archivio delle chiavi esterne

I seguenti valori vengono visualizzati nella sezione di configurazione del proxy dell'archivio chiavi esterno della pagina di dettaglio per ogni archivio di chiavi esterno e nell'XksProxyConfigurationelemento della [DescribeCustomKeyStores](#)risposta. Per una descrizione dettagliata di ogni campo, inclusi i requisiti di unicità e le informazioni utili per determinare il valore corretto per ogni campo, consulta [the section called “Assemblare i prerequisiti”](#) nell'argomento Creazione di un archivio delle chiavi esterne.

Connettività proxy

Indica se l'archivio delle chiavi esterne utilizza la [connettività dell'endpoint pubblico](#) o la [connettività del servizio endpoint VPC](#).

Endpoint dell'URI proxy

L'endpoint AWS KMS utilizzato per connettersi al [proxy dell'archivio chiavi esterno](#).

Percorso URI proxy

Il percorso dall'endpoint URI del proxy a cui AWS KMS invia le richieste [API proxy](#).

Credenziali proxy: ID chiave di accesso

Parte delle [credenziali di autenticazione proxy](#) che stabilisci nel proxy dell'archivio delle chiavi esterne. L'ID della chiave di accesso identifica la chiave di accesso segreta nelle credenziali.

AWS KMS utilizza il processo di firma SigV4 e la credenziale di autenticazione del proxy per firmare le sue richieste al proxy di archiviazione delle chiavi esterno. La credenziale contenuta nella firma consente al proxy dell'archivio chiavi esterno di autenticare le richieste per conto dell'utente da. AWS KMS

Nome del servizio endpoint VPC

Il nome del servizio endpoint Amazon VPC che supporta l'archivio delle chiavi esterne. Questo valore viene visualizzato solo quando l'archivio delle chiavi esterne utilizza la [connettività del servizio endpoint VPC](#). Puoi individuare il proxy dell'archivio delle chiavi esterne nel VPC oppure puoi utilizzare il servizio endpoint VPC per comunicare in modo sicuro con il proxy.

ID del proprietario del servizio endpoint VPC

L'ID del servizio endpoint Amazon VPC che supporta l'archivio di chiavi esterno. Questo valore viene visualizzato solo quando l'archivio delle chiavi esterne utilizza la [connettività del servizio endpoint VPC](#). Puoi individuare il proxy dell'archivio delle chiavi esterne nel VPC oppure puoi utilizzare il servizio endpoint VPC per comunicare in modo sicuro con il proxy.

Visualizza le proprietà del tuo archivio di chiavi esterne

È possibile visualizzare l'archivio di chiavi esterno e le proprietà associate nella AWS KMS console o utilizzando l'[DescribeCustomKeyStores](#) operazione.

Utilizzo della AWS KMS console

Per visualizzare gli archivi delle chiavi esterne in determinati account e regioni, utilizza la procedura seguente.

1. Accedi Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel pannello di navigazione, scegli Custom key stores (Archivi delle chiavi personalizzate), External key stores (Archivi delle chiavi esterne).
4. Per visualizzare le informazioni dettagliate su un archivio delle chiavi esterne, scegli il nome dell'archivio delle chiavi.

Utilizzando l'API AWS KMS

Per visualizzare gli archivi di chiavi esterni, utilizzate l'[DescribeCustomKeyStores](#) operazione. Per impostazione predefinita, questa operazione restituisce tutti gli store delle chiavi personalizzate presenti nell'account e nella regione. Puoi tuttavia utilizzare il parametro CustomKeyStoreName

o `CustomKeyId` (ma non entrambi) per limitare l'output a un determinato store delle chiavi personalizzate.

Per quanto riguarda gli archivi delle chiavi personalizzate, l'output include l'ID, il nome e il tipo dell'archivio delle chiavi personalizzate, oltre allo [lo stato di connessione](#) dell'archivio delle chiavi. Se lo stato della connessione è `FAILED`, l'output include un `ConnectionErrorCode` che descrive il motivo dell'errore. Per informazioni sull'interpretazione di `ConnectionErrorCode` per un archivio delle chiavi esterne, consulta [Codici di errore di connessione per archivi delle chiavi esterne](#).

Per gli archivi delle chiavi esterne, l'output include anche l'elemento `XksProxyConfiguration`. Questo elemento comprende il [tipo di connettività](#), l'[endpoint URI proxy](#), il [percorso URI proxy](#) e l'ID della chiave di accesso delle [credenziali di autenticazione proxy](#).

Gli esempi in questa sezione utilizzano [AWS Command Line Interface \(AWS CLI\)](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

Ad esempio, il comando seguente restituisce tutti gli store delle chiavi personalizzate presenti nell'account e nella regione. Per scorrere gli store delle chiavi personalizzate nell'output puoi utilizzare i parametri `Limit` e `Marker`.

```
$ aws kms describe-custom-key-stores
```

Il comando seguente utilizza il parametro `CustomKeyName` per ottenere solo l'archivio delle chiavi esterne di esempio con il nome descrittivo `ExampleXksPublic`. Tale archivio delle chiavi utilizza la connettività dell'endpoint pubblico ed è collegato al relativo proxy dell'archivio delle chiavi esterne.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksPublic
{
  "CustomKeyStores": [
    {
      "CustomKeyId": "cks-1234567890abcdef0",
      "CustomKeyName": "ExampleXksPublic",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-14T20:17:36.419000+00:00",
      "CustomKeyType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE12345670EXAMPLE",
        "Connectivity": "PUBLIC_ENDPOINT",
        "UriEndpoint": "https://xks.example.com:6443",
        "UriPath": "/example/prefix/kms/xks/v1"
```

```
    }  
  }  
]  
}
```

Tramite il comando seguente si ottiene un archivio delle chiavi esterne di esempio con connettività del servizio endpoint VPC. In questo esempio, l'archivio delle chiavi esterne è connesso al relativo proxy.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc  
{  
  "CustomKeyStores": [  
    {  
      "CustomKeyStoreId": "cks-9876543210fedcba9",  
      "CustomKeyStoreName": "ExampleXksVpc",  
      "ConnectionState": "CONNECTED",  
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",  
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",  
      "XksProxyConfiguration": {  
        "AccessKeyId": "ABCDE98765432EXAMPLE",  
        "Connectivity": "VPC_ENDPOINT_SERVICE",  
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",  
        "UriPath": "/example/prefix/kms/xks/v1",  
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"  
      }  
    }  
  ]  
}
```

Se [ConnectionState](#) è Disconnected, indica che un archivio delle chiavi esterne non è mai stato connesso oppure è stato intenzionalmente disconnesso dal relativo proxy. Se tuttavia i tentativi di utilizzare una chiave KMS in un archivio delle chiavi esterne connesso non riescono, è possibile che vi sia un problema con il proxy o altri componenti esterni.

Se il campo `ConnectionState` dell'archivio delle chiavi esterne è FAILED, la risposta `DescribeCustomKeyStores` include un elemento `ConnectionErrorCode` che descrive il motivo dell'errore.

Ad esempio, nell'output seguente, il `XKS_PROXY_TIMED_OUT` valore indica che AWS KMS può connettersi al proxy dell'archivio chiavi esterno, ma la connessione è fallita perché il proxy dell'archivio chiavi esterno non ha risposto AWS KMS nel tempo assegnato. Se visualizzi

ripetutamente questo codice di errore di connessione, informa il fornitore del proxy dell'archivio delle chiavi esterne. Per informazioni su questo e altri errori di connessione, consulta [Risoluzione dei problemi relativi all'archivio delle chiavi esterne](#).

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
  "CustomKeyStores": [
    {
      "CustomKeyId": "cks-9876543210fedcba9",
      "CustomKeyName": "ExampleXksVpc",
      "ConnectionState": "FAILED",
      "ConnectionErrorCode": "XKS_PROXY_TIMED_OUT",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
}
```

Monitora gli archivi di chiavi esterni

AWS KMS raccoglie le metriche per ogni interazione con un archivio di chiavi esterno e le pubblica nel tuo account. CloudWatch Questi parametri vengono utilizzati per generare i grafici nella sezione di monitoraggio della pagina dei dettagli relativa a ogni archivio delle chiavi esterne. L'argomento seguente descrive in dettaglio come utilizzare i grafici per identificare e risolvere i problemi operativi e di configurazione che influiscono sull'archivio delle chiavi esterne. Ti consigliamo di utilizzare le CloudWatch metriche per impostare allarmi che ti avvisino quando l'archivio chiavi esterno non funziona come previsto. Per ulteriori informazioni, consulta [Monitoraggio con Amazon CloudWatch](#).

Argomenti

- [Visualizzazione dei grafici](#)
- [Interpretazione dei grafici](#)

Visualizzazione dei grafici

Puoi visualizzare i grafici a diversi livelli di dettaglio. Per impostazione predefinita, ogni grafico utilizza un intervallo di tempo di tre ore e un [periodo](#) di aggregazione di cinque minuti. Puoi regolare la visualizzazione del grafico all'interno della console, ma le modifiche verranno ripristinate alle impostazioni predefinite quando la pagina dei dettagli dell'archivio delle chiavi esterne viene chiusa o quando il browser viene aggiornato. Per assistenza sulla CloudWatch terminologia di Amazon, consulta [Amazon CloudWatch concepts](#).

Visualizzazione dei dettagli di punti dati

I dati in ogni grafico vengono raccolti in base ai [parametri di AWS KMS](#). Per visualizzare ulteriori informazioni su un punto dati specifico, posiziona il puntatore del mouse sul punto dati del grafico a linee. Verrà visualizzato un popup con ulteriori informazioni sul parametro da cui è stato derivato il grafico. Ogni elemento dell'elenco visualizza il valore della [dimensione](#) registrato in quel punto dati. Il popup visualizza un valore nullo (—) se non sono disponibili dati di parametro per il valore della dimensione in quel punto dati. Alcuni grafici registrano più dimensioni e valori per un singolo punto dati. Altri grafici, come il [grafico di affidabilità](#), utilizzano i dati raccolti dal parametro per calcolare un valore univoco. Ogni elemento dell'elenco è associato a un colore diverso del grafico a linee.

Modifica dell'intervallo di tempo

Per modificare l'[intervallo temporale](#), seleziona uno degli intervalli di tempo predefiniti nell'angolo in alto a destra della sezione di monitoraggio. Questi intervalli predefiniti vanno da 1 ora a 1 settimana (1h [1 ora], 3h [3 ore], 12h [12 ore], 1d [1 giorno], 3d [3 giorni] oppure 1w [1 settimana]). In questo modo si regola l'intervallo di tempo per tutti i grafici. Se desideri visualizzare un grafico specifico in un intervallo di tempo diverso o se desideri impostare un intervallo di tempo personalizzato, ingrandisci il grafico o visualizzalo nella CloudWatch console Amazon.

Ingrandimento dei grafici

Puoi utilizzare la [funzione di ingrandimento mini-mappa](#) per concentrarti su sezioni di grafici a linee e porzioni impilate dei grafici senza passare tra la visualizzazione ingrandita e ridimensionata. Ad esempio, puoi utilizzare la funzione di ingrandimento mini-mappa per concentrarti su un picco di un grafico a linee, in modo da poter confrontare il picco con altri grafici nella sezione di monitoraggio dalla stessa sequenza temporale.

1. Seleziona e trascina l'area del grafico che desideri ingrandire, quindi rilasciala.
2. Per ripristinare lo zoom, seleziona l'icona Reset zoom, a forma di lente d'ingrandimento con un simbolo meno (-) all'interno.

Ingrandimento di un grafico

Per ingrandire un grafico, seleziona l'icona del menu nell'angolo in alto a destra di un singolo grafico e scegli **Enlarge** (Ingrandisci). Puoi anche selezionare l'icona di ingrandimento che appare accanto all'icona del menu quando passi il mouse su un grafico.

L'ingrandimento di un grafico consente di modificare ulteriormente la visualizzazione di un grafico specificando un periodo diverso, un intervallo di tempo personalizzato o un intervallo di aggiornamento. Queste modifiche verranno ripristinate alle impostazioni predefinite quando si chiude la vista ingrandita.

Modifica del periodo

1. Scegli il menu **Period options** (Opzioni periodo). Per impostazione predefinita, questo menu visualizza il valore 5 minuti.
2. Scegli un periodo; i periodi predefiniti vanno da 1 secondo a 30 giorni.

Ad esempio, puoi scegliere la visualizzazione di un minuto, che può essere utile durante la risoluzione dei problemi. In alternativa, scegli la visualizzazione meno dettagliata di un'ora. Può essere utile quando si visualizza un intervallo di tempo più ampio (ad esempio 3 giorni) in modo da poter vedere le tendenze nel tempo. Per ulteriori informazioni, consulta [Periodi](#) nella Amazon CloudWatch User Guide.

Modifica dell'intervallo di tempo o del fuso orario

1. Seleziona uno degli intervalli di tempo predefiniti, che partono da 1 ora fino a 1 settimana: 1h (1 ora), 3h (3 ore), 12h (12 ore), 1d (1 giorno), 3d (3 giorni) oppure 1w (1 settimana). In alternativa, è possibile scegliere **Personalizza** per impostare il tuo intervallo di tempo.
2. Scegli **Custom** (Personalizzato)
 - a. Intervallo di tempo: seleziona la scheda **Absolute** (Assoluto) nell'angolo in alto a sinistra della casella. Utilizza la selezione calendario o i campi di testo per specificare l'intervallo di tempo.
 - b. Fuso orario: scegli il menu a discesa nell'angolo in alto a destra della casella. È possibile modificare il fuso orario in UTC o Fuso orario locale.
3. Dopo aver specificato un intervallo di tempo, scegli **Applica**.

Modifica la frequenza di aggiornamento dei dati nel grafico

1. Scegli il menu **Refresh options** (Aggiorna opzioni) nell'angolo in alto a destra.

2. Scegli un intervallo di aggiornamento: Off (Disattivato), 10 Seconds (10 secondi), 1 Minute (1 minuto), 2 Minutes (2 minuti), 5 Minutes (5 minuti) o 15 Minutes (15 minuti).

Visualizza i grafici nella console Amazon CloudWatch

I grafici nella sezione di monitoraggio derivano da metriche predefinite pubblicate su Amazon AWS KMS. CloudWatch Puoi aprirli all'interno della CloudWatch console e salvarli nelle dashboard. CloudWatch Se disponi di più archivi di chiavi esterni, puoi aprire i rispettivi grafici CloudWatch e salvarli in un'unica dashboard per confrontarne lo stato e l'utilizzo.

Aggiungi alla dashboard CloudWatch

Seleziona Aggiungi alla dashboard nell'angolo in alto a destra per aggiungere tutti i grafici a una CloudWatch dashboard di Amazon. Puoi selezionare un pannello di controllo esistente o crearne uno nuovo. Per informazioni sull'utilizzo di questa dashboard per creare visualizzazioni personalizzate dei grafici e degli allarmi, consulta Using [Amazon CloudWatch dashboard nella Amazon CloudWatch User Guide](#).

Visualizza nelle metriche CloudWatch

Seleziona l'icona del menu nell'angolo in alto a destra di un singolo grafico e scegli Visualizza nelle metriche per visualizzare questo grafico nella CloudWatch console Amazon. Dalla CloudWatch console, puoi aggiungere questo grafico singolo a una dashboard e modificare intervalli di tempo, periodi e intervalli di aggiornamento. Per ulteriori informazioni, consulta la sezione [Grafica delle metriche](#) nella Amazon CloudWatch User Guide.

Interpretazione dei grafici

AWS KMS fornisce diversi grafici per monitorare lo stato dell'archivio di chiavi esterno all'interno della console. AWS KMS Questi grafici vengono configurati automaticamente e derivati dai [parametri di AWS KMS](#).

I dati del grafico vengono raccolti come parte delle chiamate effettuate all'archivio delle chiavi esterne e alle chiavi esterne. È possibile che i dati compaiano nei grafici in un intervallo di tempo in cui non sono state effettuate chiamate. Questi dati provengono dalle GetHealthStatus chiamate periodiche che AWS KMS effettuiamo per conto dell'utente per verificare lo stato del proxy dell'archivio chiavi esterno e del gestore di chiavi esterno. Se nei grafici viene visualizzato il messaggio No data available (Nessun dato disponibile), significa che non sono state registrate chiamate durante tale intervallo di tempo o che l'archivio delle chiavi esterne si trova in uno stato

[DISCONNECTED](#). Potresti riuscire a identificare l'ora in cui l'archivio delle chiavi esterne è stato disconnesso [regolando la visualizzazione](#) su un intervallo di tempo più ampio.

Argomenti

- [Total Requests \(Richieste totali\)](#)
- [Affidabilità](#)
- [Latenza](#)
- [Le 5 eccezioni principali](#)
- [Giorni alla scadenza del certificato](#)

Total Requests (Richieste totali)

Il numero totale di AWS KMS richieste ricevute per uno specifico archivio di chiavi esterno in un determinato intervallo di tempo. Usa questo grafico per determinare se sei a rischio di limitazione (della larghezza di banda della rete).

AWS KMS consiglia che il gestore delle chiavi esterno sia in grado di gestire fino a 1800 richieste di operazioni crittografiche al secondo. Se ti avvicini a 540.000 chiamate in un periodo di cinque minuti, sei a rischio di limitazione (della larghezza di banda della rete).

Puoi monitorare il numero di richieste di operazioni crittografiche sulle chiavi KMS nel tuo archivio di chiavi esterno, che limita la AWS KMS metrica. [ExternalKeyStoreThrottle](#)

Se ricevi errori `KMSInvalidStateException` molto frequenti con un messaggio che spiega che la richiesta è stata rifiutata "a causa di un tasso di richieste molto elevato", ciò potrebbe indicare che il gestore delle chiavi esterne o il proxy dell'archivio delle chiavi esterne non è in grado di tenere il passo con il tasso di richieste corrente. Se possibile, riduci il tasso di richiesta. Potresti anche prendere in considerazione la possibilità di richiedere una riduzione del valore della quota di richiesta dell'archivio delle chiavi personalizzate. La riduzione di questo valore di quota potrebbe aumentare la limitazione, ma ciò significa rifiutare rapidamente le richieste in eccesso prima che AWS KMS vengano inviate al proxy dell'archivio chiavi esterno o al gestore di chiavi esterno. Per richiedere una riduzione della quota, consulta la sezione [Centro Supporto AWS](#) e crea un caso.

Il grafico delle richieste totali deriva dal parametro [XksProxyErrors](#), che raccoglie dati sulle risposte riuscite e non riuscite che AWS KMS riceve dal proxy dell'archivio delle chiavi esterne. Quando si [visualizza un punto dati specifico](#), il pop-up mostra il valore della `CustomKeyStoreId` dimensione insieme al numero totale di AWS KMS richieste registrate in quel punto dati. Il valore di `CustomKeyStoreId` sarà sempre lo stesso.

Affidabilità

La percentuale di AWS KMS richieste per le quali il proxy dell'archivio chiavi esterno ha restituito una risposta corretta o un errore irreversibile. Utilizza questo grafico per valutare lo stato operativo del proxy dell'archivio delle chiavi esterne.

Quando il grafico mostra un valore inferiore al 100%, indica i casi in cui il proxy non ha risposto o ha risposto con un errore non irreversibile. Ciò può indicare problemi con la rete, lentezza del proxy o del gestore delle chiavi esterne o bug di implementazione.

Se la richiesta include credenziali non valide e il proxy risponde con un'eccezione `AuthenticationFailedException`, il grafico indicherà comunque un'affidabilità del 100% perché il proxy ha identificato un valore errato nella [richiesta dell'API proxy dell'archivio delle chiavi esterne](#) e quindi l'errore è previsto. Se la percentuale del grafico di affidabilità è del 100%, il proxy dell'archivio delle chiavi esterne risponde come previsto. Se il grafico mostra un valore inferiore al 100%, il proxy ha risposto con un errore non irreversibile o è scaduto. Ad esempio, se il proxy risponde con un'eccezione `ThrottlingException` a causa di un tasso di richiesta molto elevato, mostrerà una percentuale di affidabilità inferiore perché il proxy non è stato in grado di identificare un problema specifico nella richiesta che ne ha causato l'errore. Questo perché gli errori non irreversibili sono probabilmente problemi temporanei che possono essere risolti ripetendo la richiesta.

Le seguenti risposte di errore ridurranno la percentuale di affidabilità. Puoi utilizzare il grafico [Le 5 eccezioni principali](#) e il parametro [XksProxyErrors](#) per monitorare ulteriormente la frequenza con cui il proxy restituisce ogni errore non irreversibile.

- `InternalException`
- `DependencyTimeoutException`
- `ThrottlingException`
- `XksProxyUnreachableException`

Il grafico di affidabilità è derivato dalla [XksProxyErrors](#) metrica, che raccoglie i dati sulle risposte riuscite e non riuscite AWS KMS ricevute dal proxy dell'archivio chiavi esterno. La percentuale di affidabilità diminuirà solo se la risposta ha un valore `ErrorType` di `Retryable`. Quando si [visualizza un punto dati specifico](#), il pop-up mostra il valore della `CustomKeyStoreId` dimensione insieme alla percentuale di affidabilità per le AWS KMS richieste registrate in quel punto dati. Il valore di `CustomKeyStoreId` sarà sempre lo stesso.

Ti consigliamo di utilizzare la [XksProxyErrors](#) metrica per creare un CloudWatch allarme che ti avvisi di potenziali problemi di rete avvisandoti quando vengono registrati più di cinque errori ripetibili in un periodo di un minuto. Per ulteriori informazioni, consulta [Crea un allarme per errori ripetibili](#).

Latenza

Il numero di millisecondi impiegato da un proxy di archiviazione chiavi esterno per rispondere a una richiesta. AWS KMS Utilizza questo grafico per valutare le prestazioni del proxy dell'archivio delle chiavi esterne e del gestore delle chiavi esterne.

AWS KMS prevede che il proxy di archiviazione delle chiavi esterno risponda a ciascuna richiesta entro 250 millisecondi. In caso di timeout di rete, riproverà la richiesta una volta. AWS KMS Se il proxy restituisce ancora una volta un errore, la latenza registrata è il limite di timeout combinato per entrambi i tentativi di richiesta e il grafico mostrerà circa 500 millisecondi. In tutti gli altri casi in cui il proxy non risponde entro il limite di timeout di 250 millisecondi, la latenza registrata è di 250 millisecondi. Se il proxy è spesso in timeout durante le operazioni di crittografia e decrittografia, rivolgiti all'amministratore del proxy esterno. Per informazioni sulla risoluzione dei problemi di latenza, consulta [Errori di latenza e timeout](#).

Le risposte lente potrebbero anche indicare che il gestore delle chiavi esterno non è in grado di gestire il traffico della richiesta corrente. AWS KMS consiglia che il gestore delle chiavi esterno sia in grado di gestire fino a 1800 richieste di operazioni crittografiche al secondo. Se il gestore delle chiavi esterne non è in grado di gestire 1.800 richieste al secondo, prendi in considerazione la possibilità di richiedere una riduzione della [quota di richieste per le chiavi KMS in un archivio delle chiavi personalizzate](#). Le richieste relative alle operazioni di crittografia che utilizzano le chiavi KMS nell'archivio delle chiavi esterne anticiperanno rapidamente l'errore (fail fast) con un'[eccezione di limitazione](#) (della larghezza di banda della rete), anziché essere elaborate e successivamente rifiutate dal proxy o dal gestore delle chiavi esterne.

Il grafico della latenza è derivato dal parametro [XksProxyLatency](#). Quando [visualizzi un punto dati specifico](#), il popup mostra i valori delle dimensioni KmsOperation e XksOperation corrispondenti, oltre alla latenza media registrata per le operazioni in quel punto dati. Gli elementi dell'elenco sono ordinati dalla latenza più alta a quella più bassa.

Ti consigliamo di utilizzare la [XksProxyLatency](#) metrica per creare un CloudWatch allarme che ti avvisi quando la latenza si avvicina al limite di timeout. Per ulteriori informazioni, consulta [Crea un allarme per il timeout della risposta](#).

Le 5 eccezioni principali

Le cinque eccezioni principali per le operazioni di crittografia e di gestione non riuscite in un determinato intervallo di tempo. Usa questo grafico per tenere traccia degli errori più frequenti, in modo da poter assegnare priorità diverse agli interventi tecnici.

Questo conteggio include le eccezioni AWS KMS ricevute dal proxy dell'archivio chiavi esterno e quelle che vengono AWS KMS restituite internamente quando non è in grado di stabilire una comunicazione con il `XksProxyUnreachableException` proxy dell'archivio chiavi esterno.

Tassi elevati di errori non irreversibili potrebbero indicare errori di rete, mentre un'elevata percentuale di errori irreversibili potrebbe indicare un problema con la configurazione dell'archivio delle chiavi esterne. Ad esempio, un piccolo `AuthenticationFailedExceptions` indica una discrepanza tra le credenziali di autenticazione configurate nel AWS KMS proxy dell'archivio chiavi esterno. Per visualizzare la configurazione dell'archivio delle chiavi esterne, consulta [Visualizza gli archivi di chiavi esterni](#). Per modificare le impostazioni dell'archivio delle chiavi esterne, consulta [Modifica delle proprietà dell'archivio chiavi esterno](#).

Le eccezioni AWS KMS ricevute dal proxy dell'archivio chiavi esterno sono diverse dalle eccezioni che AWS KMS vengono restituite quando un'operazione non riesce. AWS KMS le operazioni crittografiche restituiscono un `KMSInvalidStateException` valore per tutti gli errori relativi alla configurazione esterna o allo stato di connessione dell'archivio di chiavi esterno. Per identificare il problema, utilizza il testo del messaggio di errore allegato.

La tabella seguente mostra le eccezioni che possono apparire nel grafico delle prime 5 eccezioni e le eccezioni corrispondenti che vengono restituite all'utente. AWS KMS

Tipi di errore	Eccezione visualizzata nel grafico	Eccezione che AWS KMS ti è stata restituita
Irreversibile	AccessDeniedException Per la risoluzione dei problemi, consultare Problemi relativi all'autorizzazione proxy .	CustomKeyStoreInvalidStateException in risposta alle operazioni <code>CreateKey</code> . KMSInvalidStateException in risposta alle operazioni di crittografia.

Tipi di errore	Eccezione visualizzata nel grafico	Eccezione che AWS KMS ti è stata restituita
Irreversibile	AuthenticationFailedException Per la risoluzione dei problemi, consultare Errori delle credenziali di autenticazione .	XksProxyIncorrectAuthenticationCredentialException in risposta alle operazioni <code>CreateCustomKeyStore</code> e <code>UpdateCustomKeyStore</code> . CustomKeyStoreInvalidStateException in risposta alle operazioni <code>CreateKey</code> . KMSInvalidStateException in risposta alle operazioni di crittografia.
Non irreversibile	DependencyTimeoutException Per la risoluzione dei problemi, consultare Errori di latenza e timeout .	XksProxyUriUnreachableException in risposta alle operazioni <code>CreateCustomKeyStore</code> e <code>UpdateCustomKeyStore</code> . CustomKeyStoreInvalidStateException in risposta alle operazioni <code>CreateKey</code> . KMSInvalidStateException in risposta alle operazioni di crittografia.

Tipi di errore	Eccezione visualizzata nel grafico	Eccezione che AWS KMS ti è stata restituita
Non irreversibile	InternalException Il proxy dell'archivio delle chiavi esterne ha rifiutato la richiesta perché non è in grado di comunicare con il gestore delle chiavi esterne. Verifica che la configurazione del proxy dell'archivio delle chiavi esterne sia corretta e che il gestore delle chiavi esterne sia disponibile.	XksProxyInvalidResponseException in risposta alle operazioni <code>CreateCustomKeyStore</code> e <code>UpdateCustomKeyStore</code> . CustomKeyStoreInvalidStateException in risposta alle operazioni <code>CreateKey</code> . KMSInvalidStateException in risposta alle operazioni di crittografia.
Irreversibile	InvalidCiphertextException Per la risoluzione dei problemi, consultare Errori di decrittografia .	KMSInvalidStateException in risposta alle operazioni di crittografia.
Irreversibile	InvalidKeyUsageException Per la risoluzione dei problemi, consultare Errori relativi alle operazioni di crittografia per la chiave esterna .	XksKeyInvalidConfigurationException in risposta alle operazioni <code>CreateKey</code> . KMSInvalidStateException in risposta alle operazioni di crittografia.

Tipi di errore	Eccezione visualizzata nel grafico	Eccezione che AWS KMS ti è stata restituita
Irreversibile	InvalidStateException Per la risoluzione dei problemi, consultare Errori relativi alle operazioni di crittografia per la chiave esterna .	XksKeyInvalidConfigurationException in risposta alle operazioni <code>CreateKey</code> . KMSInvalidStateException in risposta alle operazioni di crittografia.
Irreversibile	InvalidUriPathException Per la risoluzione dei problemi, consultare Errori di configurazione generale .	XksProxyInvalidConfigurationException in risposta alle operazioni <code>CreateCustomKeyStore</code> e <code>UpdateCustomKeyStore</code> . CustomKeyStoreInvalidStateException in risposta alle operazioni <code>CreateKey</code> . KMSInvalidStateException in risposta alle operazioni di crittografia.
Irreversibile	KeyNotFoundException Per la risoluzione dei problemi, consultare Errori relativi alla chiave esterna .	XksKeyNotFoundException in risposta alle operazioni <code>CreateKey</code> . KMSInvalidStateException in risposta alle operazioni di crittografia.

Tipi di errore	Eccezione visualizzata nel grafico	Eccezione che AWS KMS ti è stata restituita
Non irreversibile	ThrottlingException Il proxy dell'archivio delle chiavi esterne ha rifiutato la richiesta a causa di un tasso di richieste molto elevato. Riduci la frequenza delle chiamate utilizzando le chiavi KMS in questo archivio delle chiavi esterne.	XksProxyUriUnreachableException in risposta alle operazioni <code>CreateCustomKeyStore</code> e <code>UpdateCustomKeyStore</code> . CustomKeyStoreInvalidStateException in risposta alle operazioni <code>CreateKey</code> . KMSInvalidStateException in risposta alle operazioni di crittografia.
Irreversibile	UnsupportedOperationException Per la risoluzione dei problemi, consultare Errori relativi alle operazioni di crittografia per la chiave esterna.	XksKeyInvalidResponseException in risposta alle operazioni <code>CreateKey</code> . KMSInvalidStateException in risposta alle operazioni di crittografia.

Tipi di errore	Eccezione visualizzata nel grafico	Eccezione che AWS KMS ti è stata restituita
Irreversibile	ValidationException Per la risoluzione dei problemi, consultare Problemi relativi al proxy .	XksProxyInvalidResponseException in risposta alle operazioni CreateCustomKeyStore e UpdateCustomKeyStore . CustomKeyStoreInvalidStateException in risposta alle operazioni CreateKey . KMSInvalidStateException in risposta alle operazioni di crittografia.
Non irreversibile	XksProxyUnreachableException Se riscontri ripetutamente questo errore, verifica che il proxy dell'archivio delle chiavi esterne sia attivo e connesso alla rete. Verifica inoltre che il percorso URI e l'URI endpoint o il nome del servizio VPC all'interno dell'archivio delle chiavi esterne siano corretti.	XksProxyUriUnreachableException in risposta alle operazioni CreateCustomKeyStore e UpdateCustomKeyStore . CustomKeyStoreInvalidStateException in risposta alle operazioni CreateKey . KMSInvalidStateException in risposta alle operazioni di crittografia.

Il grafico delle 5 eccezioni principali deriva dal parametro [XksProxyErrors](#). Quando [visualizzi un punto dati specifico](#), il popup mostra il valore della dimensione ExceptionName, insieme al numero di

volte in cui l'eccezione è stata registrata in quel punto dati. Le cinque voci dell'elenco sono ordinate dall'eccezione più frequente alla meno frequente.

Ti consigliamo di utilizzare la [XksProxyErrors](#) metrica per creare un CloudWatch allarme che ti avvisi di potenziali problemi di configurazione avvisandoti quando vengono registrati più di cinque errori non ripetibili in un periodo di un minuto. Per ulteriori informazioni, consulta [Crea un allarme per errori irreparabili](#).

Giorni alla scadenza del certificato

Il numero di giorni che mancano alla scadenza del certificato TLS per l'endpoint proxy dell'archivio delle chiavi esterne (XksProxyUriEndpoint). Utilizza questo grafico per monitorare la scadenza imminente del certificato TLS.

Quando il certificato scade, AWS KMS non può comunicare con il proxy esterno dell'archivio chiavi. Tutti i dati protetti dalle chiavi KMS nell'archivio delle chiavi esterne diventano inaccessibili fino al rinnovo del certificato.

Il grafico dei giorni di scadenza del certificato deriva dal parametro [XksProxyCertificateDaysToExpire](#). Ti consigliamo vivamente di utilizzare questa metrica per creare un CloudWatch allarme che ti avvisi della scadenza imminente. La scadenza del certificato potrebbe impedirti di accedere alle risorse crittografate. Configura l'allarme in modo che l'organizzazione abbia la possibilità di rinnovare il certificato prima della sua scadenza. Per ulteriori informazioni, consulta [Crea un allarme per la scadenza del certificato](#).

Connect e disconnetti gli archivi di chiavi esterni

I nuovi archivi delle chiavi esterne non sono connessi. Per creare e utilizzare AWS KMS keys nell'archivio chiavi esterno, è necessario collegare l'archivio chiavi esterno al relativo [proxy di archiviazione chiavi esterno](#). Puoi connettere e disconnettere l'archivio delle chiavi esterne in qualsiasi momento e [visualizzare il relativo stato di connessione](#).

Mentre l'archivio chiavi esterno è disconnesso, AWS KMS non è possibile comunicare con il proxy dell'archivio chiavi esterno. Di conseguenza, puoi visualizzare e gestire l'archivio delle chiavi esterne e le relative chiavi KMS, ma non puoi creare chiavi KMS nell'archivio delle chiavi esterne o utilizzare le relative chiavi KMS in operazioni di crittografia. In alcuni casi, ad esempio si modificano le proprietà, potresti dover disconnettere l'archivio delle chiavi esterne, per cui ti consigliamo di pianificare le operazioni di conseguenza. La disconnessione dell'archivio chiavi potrebbe interrompere il funzionamento dei AWS servizi che utilizzano le relative chiavi KMS.

Non sei obbligato a connettere l'archivio delle chiavi esterne. Puoi lasciarlo disconnesso indefinitamente e connetterlo solo quando devi utilizzarlo. Puoi tuttavia testare la connessione periodicamente per verificare che le impostazioni sono corrette e che non vi sono problemi di connessione dello store.

Quando disconnetti un archivio delle chiavi personalizzate, le chiavi KMS nell'archivio diventano immediatamente inutilizzabili (in base alla coerenza finale). Tuttavia, le risorse crittografate con [chiavi di dati](#) protette dalla chiave KMS non sono interessate fino a quando la chiave KMS non viene nuovamente utilizzata, ad esempio per decrittografare la chiave dati. Questo problema riguarda i Servizi AWS, molti dei quali proteggono le risorse tramite le chiavi dati. Per informazioni dettagliate, consultare [In che modo le chiavi KMS inutilizzabili influiscono sulle chiavi dati](#).

Note

Gli archivi delle chiavi esterne presentano lo stato DISCONNECTED solo quando l'archivio non è mai stato connesso o se lo si disconnette esplicitamente. Lo stato CONNECTED non indica che l'archivio delle chiavi esterne o i relativi componenti di supporto funzionino in modo efficiente. Per informazioni relative alle prestazioni dei componenti dell'archivio delle chiavi esterne, consulta i grafici nella sezione Monitoraggio della pagina dei dettagli di ogni archivio delle chiavi esterne. Per informazioni dettagliate, consultare [Monitora gli archivi di chiavi esterni](#).

Il gestore delle chiavi esterno potrebbe fornire metodi aggiuntivi per interrompere e riavviare la comunicazione tra l'archivio chiavi AWS KMS esterno e il proxy dell'archivio chiavi esterno o tra il proxy dell'archivio chiavi esterno e il gestore di chiavi esterno. Per ulteriori informazioni, consulta la documentazione del gestore delle chiavi esterne.

Argomenti

- [Stato connessione](#)
- [Connect un key store esterno](#)
- [Disconnetti un archivio di chiavi esterno](#)

Stato connessione

La connessione e la disconnessione modificano lo stato di connessione dell'archivio delle chiavi personalizzate. I valori dello stato di connessione sono gli stessi per gli archivi di AWS CloudHSM chiavi e gli archivi di chiavi esterni.

Per visualizzare lo stato di connessione del tuo archivio chiavi personalizzato, utilizza l'[DescribeCustomKeyStores](#) operazione o la AWS KMS console. Il campo Connection state (Stato connessione) viene visualizzato in ogni tabella dell'archivio delle chiavi personalizzate, nella sezione General configuration (Configurazione generale) della pagina dei dettagli di ogni archivio e nella scheda Cryptographic configuration (Configurazione crittografica) delle chiavi KMS. Per informazioni dettagliate, consulta [Visualizza un archivio di AWS CloudHSM chiavi](#) e [Visualizza gli archivi di chiavi esterni](#).

Un archivio delle chiavi personalizzate può avere uno dei seguenti stati di connessione:

- **CONNECTED:** l'archivio delle chiavi personalizzate è connesso al relativo archivio del materiale della chiave. Puoi creare o utilizzare le chiavi KMS nell'archivio delle chiavi personalizzate.

L'archivio di chiavi di backup per un AWS CloudHSM key store è il AWS CloudHSM cluster associato. L'archivio del materiale della chiave per un archivio delle chiavi esterne è rappresentato da un proxy dell'archivio delle chiavi esterne e dal gestore delle chiavi esterne che supporta.

Uno stato CONNECTED (CONNESSO) indica che la connessione è riuscita e che l'archivio delle chiavi personalizzate non è stato disconnesso intenzionalmente. Non indica che la connessione sta funzionando correttamente. Per informazioni sullo stato del AWS CloudHSM cluster associato al tuo AWS CloudHSM key store, consulta [Ottenere le CloudWatch metriche AWS CloudHSM nella Guida per l' AWS CloudHSM utente](#). Per informazioni sullo stato e sul funzionamento dell'archivio delle chiavi esterne, consulta i grafici nella sezione Monitoring (Monitoraggio) della pagina dei dettagli di ogni archivio. Per informazioni dettagliate, consultare [Monitora gli archivi di chiavi esterni](#).

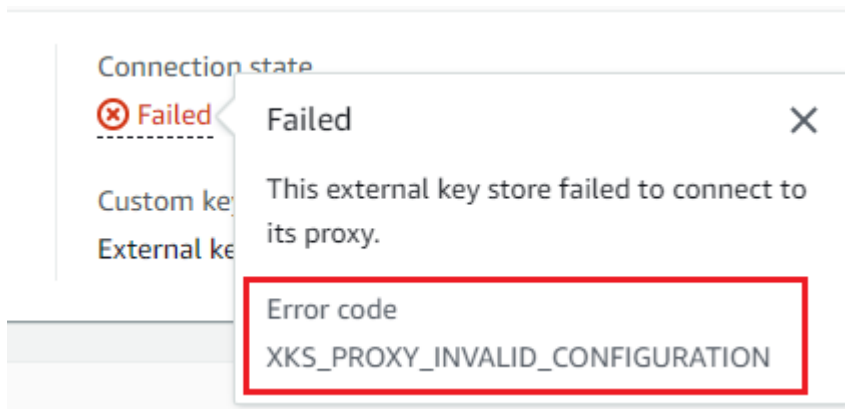
- **CONNECTING:** il processo di connessione di un archivio delle chiavi personalizzate è in corso. Si tratta di uno stato transitorio.
- **DISCONNECTED:** L'archivio chiavi personalizzato non è mai stato collegato al relativo supporto oppure è stato disconnesso intenzionalmente utilizzando la AWS KMS console o l'operazione [DisconnectCustomKeyStore](#).
- **DISCONNECTING:** il processo di disconnessione di un archivio delle chiavi personalizzate è in corso. Si tratta di uno stato transitorio.
- **FAILED:** tentativo di connessione dell'archivio delle chiavi personalizzate non riuscito. `ConnectionErrorCode` Nella [DescribeCustomKeyStores](#) risposta indica il problema.

Per connettere un archivio delle chiavi personalizzate, il relativo stato di connessione deve essere DISCONNECTED. Se lo stato della connessione è FAILED, utilizza `ConnectionErrorCode` per

identificare e risolvere il problema. Disconnetti quindi l'archivio delle chiavi personalizzate prima di provare a connetterlo di nuovo. Per informazioni sugli errori di connessione, consulta [Errori di connessione all'archivio delle chiavi esterne](#). Per informazioni sulla risposta a un codice di errore di connessione, consulta [Codici di errore di connessione per archivi delle chiavi esterne](#).

Per visualizzare il codice di errore della connessione:

- Nella [DescribeCustomKeyStores](#) risposta, visualizza il valore dell'`ConnectionErrorCode` elemento. Tale elemento appare nella risposta `DescribeCustomKeyStores` solo quando `ConnectionState` è nello stato `FAILED`.
- Per visualizzare il codice di errore di connessione nella AWS KMS console, vai alla pagina di dettaglio dell'archivio chiavi esterno e passa il mouse sul valore `Failed`.



Connect un key store esterno

Quando l'archivio delle chiavi esterne è connesso al relativo proxy, puoi [creare chiavi KMS nello stesso archivio delle chiavi esterne](#) e utilizzare le chiavi KMS esistenti in [operazioni di crittografia](#).

Il processo che collega un archivio delle chiavi esterne al relativo proxy varia in base alla connettività dell'archivio.

- Quando connetti un archivio di chiavi esterno con [connettività endpoint pubblica](#), AWS KMS invia una [GetHealthStatus richiesta](#) al proxy dell'archivio chiavi esterno per convalidare l'[endpoint URI del proxy, il percorso URI del proxy](#) e le credenziali di autenticazione del [proxy](#). Una risposta positiva da parte del proxy conferma che l'[endpoint dell'URI proxy](#) e il [percorso URI proxy](#) sono corretti e accessibili e che il proxy ha autenticato la richiesta firmata con le [credenziali di autenticazione proxy](#) per l'archivio delle chiavi esterne.
- Quando colleghi un key store esterno con [connettività del servizio endpoint VPC](#) al relativo proxy di archiviazione chiavi esterno, AWS KMS effettua le seguenti operazioni:

- Conferma che il dominio per il nome DNS privato specificato nell'[endpoint URI proxy](#) è [verificato](#).
- Crea un endpoint di interfaccia da un AWS KMS VPC al tuo servizio di endpoint VPC.
- Crea una zona ospitata privata per il nome DNS privato specificato nell'endpoint URI proxy
- Invia una [GetHealthStatusrichiesta al proxy](#) dell'archivio chiavi esterno. Una risposta positiva da parte del proxy conferma che l'[endpoint dell'URI proxy](#) e il [percorso URI proxy](#) sono corretti e accessibili e che il proxy ha autenticato la richiesta firmata con le [credenziali di autenticazione proxy](#) per l'archivio delle chiavi esterne.

L'operazione di connessione avvia il processo di connessione dell'archivio delle chiavi personalizzate, ma il collegamento di un archivio delle chiavi esterne al relativo proxy esterno richiede circa cinque minuti. Una risposta positiva dell'operazione di connessione non indica che l'archivio delle chiavi esterne sia connesso. Per confermare che la connessione è avvenuta correttamente, utilizza la AWS KMS console o l'[DescribeCustomKeyStores](#) operazione per visualizzare lo [stato della connessione](#) del tuo key store esterno.

Quando lo stato della connessione è FAILED 0, nella AWS KMS console viene visualizzato un codice di errore di connessione che viene aggiunto alla DescribeCustomKeyStore risposta. Per informazioni sull'interpretazione dei codici di errore di connessione, consulta [Codici di errore di connessione per archivi delle chiavi esterne](#).

Connect e riconnettiti al tuo key store esterno

È possibile connettere o ricollegare l'archivio chiavi esterno nella AWS KMS console o utilizzando l'[ConnectCustomKeyStore](#) operazione.

Utilizzo della console AWS KMS

È possibile utilizzare la AWS KMS console per connettere un archivio chiavi esterno al relativo proxy di archiviazione chiavi esterno.

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) su <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel pannello di navigazione, scegli Custom key stores (Archivi delle chiavi personalizzate), External key stores (Archivi delle chiavi esterne).
4. Scegli la riga relativa all'archivio delle chiavi esterne che vuoi connettere.

Se la voce [Connection state](#) (Stato connessione) dell'archivio delle chiavi esterne è FAILED (NON RIUSCITO), devi [disconnettere l'archivio](#) prima di connetterlo.

5. Dal menu Key store actions (Operazioni per l'archivio delle chiavi), scegli Connect (Connetti).

Il completamento del processo di connessione richiede in genere circa cinque minuti. Al termine dell'operazione, lo [stato di connessione](#) cambia in CONNECTED (CONNESSO).

Se lo stato della connessione è Failed (Non riuscito), passa il mouse sullo stato per visualizzare il codice di errore di connessione e la causa dell'errore. Per informazioni sulla risposta a un codice di errore di connessione, consulta [Codici di errore di connessione per archivi delle chiavi esterne](#). Per connettere un archivio delle chiavi esterne con uno stato di connessione Failed (Non riuscito), devi innanzitutto [disconnettere l'archivio delle chiavi personalizzate](#).

Utilizzando l'API AWS KMS

Per connettere un archivio di chiavi esterno disconnesso, utilizzare l'[ConnectCustomKeyStore](#) operazione.

Prima della connessione, lo [stato](#) dell'archivio delle chiavi esterne deve essere DISCONNECTED. Se lo stato di connessione corrente è FAILED, [disconnetti l'archivio delle chiavi esterne](#) e riconnettilo.

Il completamento del processo di connessione può richiedere fino a cinque minuti. Se l'operazione non genera rapidamente un errore, ConnectCustomKeyStore restituisce una risposta HTTP 200 e un oggetto JSON senza proprietà. Questa risposta iniziale non indica tuttavia che la connessione è riuscita. Per determinare se l'archivio chiavi esterno è connesso, vedi lo stato della connessione nella [DescribeCustomKeyStores](#) risposta.

Gli esempi in questa sezione utilizzano [AWS Command Line Interface \(AWS CLI\)](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

Per identificare l'archivio delle chiavi esterne, utilizza l'ID dell'archivio delle chiavi personalizzate. È possibile trovare l'ID nella pagina Custom key stores della console o utilizzando l'[DescribeCustomKeyStores](#) operazione. Prima di eseguire questo esempio, sostituisci l'ID di esempio con uno valido.

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

L'operazione ConnectCustomKeyStore non restituisce alcun ConnectionState nella risposta. Per verificare che l'archivio chiavi esterno sia connesso, utilizzare

l'[DescribeCustomKeyStores](#) operazione. Per impostazione predefinita, questa operazione restituisce tutti gli store delle chiavi personalizzate presenti nel tuo account e nella regione. Puoi tuttavia utilizzare il parametro `CustomKeyStoreName` o `CustomKeyStoreId` (ma non entrambi) per limitare la risposta a determinati store delle chiavi personalizzate. Un `ConnectionState` con valore `CONNECTED` indica che l'archivio delle chiavi esterne è connesso al relativo proxy.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
  "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
}
```

Se il valore `ConnectionState` nella risposta `DescribeCustomKeyStores` è `FAILED`, l'elemento `ConnectionErrorCode` indica il motivo dell'errore.

Nell'esempio seguente, il `XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND` valore di `ConnectionErrorCode` indica che non AWS KMS riesce a trovare il servizio endpoint VPC che utilizza per comunicare con il proxy dell'archivio chiavi esterno. Verifica che `XksProxyVpcEndpointServiceName` sia corretto, che l'entità del AWS KMS servizio sia un'entità consentita sul servizio endpoint Amazon VPC e che il servizio endpoint VPC non richieda l'accettazione delle richieste di connessione. Per informazioni sulla risposta a un codice di errore di connessione, consulta [Codici di errore di connessione per archivi delle chiavi esterne](#).

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
  "CustomKeyStores": [
    {
```

```
"CustomKeyStoreId": "cks-9876543210fedcba9",
"CustomKeyStoreName": "ExampleXksVpc",
"ConnectionState": "FAILED",
"ConnectionErrorCode": "XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND",
"CreationDate": "2022-12-13T18:34:10.675000+00:00",
"CustomKeyStoreType": "EXTERNAL_KEY_STORE",
"XksProxyConfiguration": {
  "AccessKeyId": "ABCDE98765432EXAMPLE",
  "Connectivity": "VPC_ENDPOINT_SERVICE",
  "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
  "UriPath": "/example/prefix/kms/xks/v1",
  "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
}
}
]
```

Disconnetti un archivio di chiavi esterno

Quando disconnetti un archivio delle chiavi esterne con [connettività del servizio endpoint VPC](#) dal relativo proxy dell'archivio delle chiavi esterne, AWS KMS elimina l'endpoint di interfaccia per il servizio endpoint VPC e rimuove l'infrastruttura di rete che ha creato per supportare la connessione. Non è richiesto alcun processo equivalente per gli archivi delle chiavi esterne con connettività dell'endpoint pubblico. Questa operazione non influisce sul servizio endpoint VPC o sui suoi componenti di supporto. Inoltre, non ha alcun impatto sul proxy dell'archivio delle chiavi esterne o su eventuali componenti esterni.

Mentre l'archivio chiavi esterno è disconnesso, AWS KMS non invia alcuna richiesta al proxy dell'archivio chiavi esterno. Lo stato di connessione dell'archivio delle chiavi esterne è **DISCONNECTED**. Le chiavi KMS nell'archivio delle chiavi esterne disconnesso presentano uno [stato UNAVAILABLE](#) (a meno che non siano in [attesa di eliminazione](#)), pertanto non possono essere utilizzate nelle operazioni di crittografia. Tuttavia, puoi comunque visualizzare e gestire l'archivio delle chiavi esterne e le relative chiavi KMS esistenti.

Lo stato disconnesso è progettato per essere temporaneo e reversibile. La riconnessione dell'archivio delle chiavi esterne può avvenire in qualsiasi momento. In genere, non è necessaria alcuna riconfigurazione. Tuttavia, se alcune proprietà del proxy dell'archivio delle chiavi esterne associato sono state modificate durante la disconnessione, ad esempio la rotazione delle [credenziali di autenticazione proxy](#), è necessario [modificare le impostazioni dell'archivio delle chiavi esterne](#) prima di riconnetterlo.

 Note

Quando un archivio delle chiavi personalizzate è disconnesso, tutti i tentativi di creare chiavi KMS nell'archivio delle chiavi personalizzate o di utilizzare le chiavi KMS in operazioni di crittografia avrà esito negativo. Questa azione può impedire agli utenti di archiviare e accedere ai dati sensibili.

Per valutare meglio l'effetto della disconnessione dell'archivio delle chiavi esterne, identifica le chiavi KMS e [determinane l'utilizzo precedente](#).

Puoi disconnettere un archivio delle chiavi esterne per i seguenti motivi:

- Per modificarne le proprietà. Puoi modificare il nome dell'archivio delle chiavi personalizzate, il percorso URI proxy e le credenziali di autenticazione proxy mentre l'archivio delle chiavi esterne è connesso. Tuttavia, per modificare il tipo di connettività proxy, l'endpoint dell'URI proxy o il nome del servizio endpoint VPC, devi prima disconnettere l'archivio delle chiavi esterne. Per informazioni dettagliate, consultare [Modifica delle proprietà dell'archivio chiavi esterno](#).
- Per interrompere tutte le comunicazioni tra AWS KMS e il proxy dell'archivio chiavi esterno. Puoi anche interrompere la comunicazione tra AWS KMS e il tuo proxy disabilitando l'endpoint o il servizio endpoint VPC. Inoltre, il proxy di archiviazione delle chiavi esterno o il software di gestione delle chiavi potrebbe fornire meccanismi aggiuntivi per AWS KMS impedire la comunicazione con il proxy o per impedire al proxy di accedere al gestore di chiavi esterno.
- Per disabilitare tutte le chiavi KMS nell'archivio delle chiavi esterne. È possibile [disabilitare e riattivare le chiavi KMS](#) in un archivio di chiavi esterno utilizzando la AWS KMS console o l'operazione. [DisableKey](#) Queste operazioni vengono completate rapidamente (in base alla coerenza finale), ma agiscono su una sola chiave KMS alla volta. La disconnessione modifica lo stato di chiave di tutte le chiavi KMS nell'archivio delle chiavi esterne in `Unavailable`, che ne impedisce l'utilizzo in qualsiasi operazione di crittografia.
- Per riparare un tentativo di connessione non riuscito. Se un tentativo di connessione di un archivio delle chiavi esterne ha esito negativo (il relativo stato di connessione è `FAILED`), devi disconnettere l'archivio prima di eseguire un nuovo tentativo di connessione.

Disconnetti l'archivio di chiavi esterno

È possibile disconnettere l'archivio di chiavi esterno nella AWS KMS console o utilizzando l'[DisconnectCustomKeyStore](#) operazione.

Utilizzo della console AWS KMS

È possibile utilizzare la AWS KMS console per connettere un archivio chiavi esterno al relativo proxy di archiviazione chiavi esterno. Questo processo dura circa 5 minuti.

1. Accedi a Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) su <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel pannello di navigazione, scegli Custom key stores (Archivi delle chiavi personalizzate), External key stores (Archivi delle chiavi esterne).
4. Scegli la riga relativa all'archivio delle chiavi esterne che vuoi disconnettere.
5. Dal menu Key store actions (Operazioni per l'archivio delle chiavi), scegli Disconnect (Disconnetti).

Al completamento dell'operazione, lo stato della connessione cambia da CONNECTED (CONNESSO) a DISCONNECTED (DISCONNESSO). Se l'operazione ha esito negativo, viene visualizzato un messaggio di errore che descrive il problema e fornisce istruzioni su come risolverlo. Per ulteriori informazioni, consulta [Errori di connessione all'archivio delle chiavi esterne](#).

Utilizzando l'API AWS KMS

Per disconnettere un archivio di chiavi esterno connesso, utilizzare l'[DisconnectCustomKeyStore](#) operazione. Se l'operazione ha esito positivo, AWS KMS restituisce una risposta HTTP 200 e un oggetto JSON senza proprietà. Il completamento del processo può richiedere fino a cinque minuti. Per trovare lo stato di connessione dell'archivio di chiavi esterno, utilizzate l'[DescribeCustomKeyStores](#) operazione.

Gli esempi in questa sezione utilizzano [AWS Command Line Interface \(AWS CLI\)](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato.

Questo esempio disconnette un archivio delle chiavi esterne con connettività del servizio endpoint VPC. Prima di eseguire questo comando, sostituisci l'ID dell'archivio delle chiavi personalizzate di esempio con uno valido.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Per verificare che l'archivio chiavi esterno sia disconnesso, utilizzare l'[DescribeCustomKeyStores](#) operazione. Per impostazione predefinita, questa operazione restituisce tutti gli store delle chiavi personalizzate presenti nel tuo account e nella regione. Puoi tuttavia utilizzare il parametro `CustomKeyStoreName` o `CustomKeyStoreId` (ma non entrambi) per limitare la risposta a determinati store delle chiavi personalizzate. Il `ConnectionState` con valore `DISCONNECTED` indica che questo archivio delle chiavi esterne di esempio non è più connesso al relativo proxy.

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
  "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "DISCONNECTED",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
}
```

Eliminare un archivio chiavi esterno

Quando si elimina un archivio di chiavi esterno, AWS KMS elimina tutti i metadati relativi all'archivio chiavi esterno da AWS KMS, incluse le informazioni sul proxy dell'archivio chiavi esterno. Questa operazione non influisce sul [proxy dell'archivio chiavi esterno](#), sul [gestore](#) delle [chiavi esterne](#), [sulle chiavi esterne](#) o su qualsiasi AWS risorsa creata per supportare l'archivio chiavi esterno, come un Amazon VPC o un servizio endpoint VPC.

Prima di eliminare un archivio delle chiavi esterne, devi [eliminare tutte le chiavi KMS](#) dall'archivio delle chiavi e [disconnettere l'archivio](#) dal relativo proxy. In caso contrario, i tentativi di eliminare l'archivio delle chiavi hanno esito negativo.

L'operazione di eliminazione di un archivio delle chiavi esterne è irreversibile, tuttavia puoi creare un nuovo archivio e associarlo allo stesso proxy dell'archivio delle chiavi esterne e allo stesso gestore delle chiavi esterne. Tuttavia, non è possibile ricreare le chiavi KMS di crittografia simmetrica nell'archivio di chiavi esterno, anche se si ha accesso allo stesso materiale di chiave esterna. AWS KMS include i metadati nel testo cifrato simmetrico unico per ogni chiave KMS. Questa funzionalità di sicurezza garantisce che solo la chiave KMS che ha crittografato i dati possa decrittografarli.

Anziché eliminare l'archivio delle chiavi esterne, puoi disconnetterlo. Mentre un archivio chiavi esterno è disconnesso, è possibile gestire l'archivio chiavi esterno e il relativo, AWS KMS keys ma non è possibile creare o utilizzare le chiavi KMS nell'archivio chiavi esterno. Puoi ricollegare l'archivio delle chiavi esterne in qualsiasi momento e utilizzare le chiavi KMS per crittografare e decrittografare i dati. Non è previsto alcun costo per un proxy dell'archivio delle chiavi esterne disconnesso o per le relative chiavi KMS non disponibili.

È possibile eliminare l'archivio di chiavi esterno nella AWS KMS console o utilizzando l'[DeleteCustomKeyStore](#) operazione.

Utilizzo della AWS KMS console

È possibile utilizzare la AWS KMS console per eliminare un archivio di chiavi esterno.

1. Accedi Console di gestione AWS e apri la console AWS Key Management Service (AWS KMS) in <https://console.aws.amazon.com/kms>.
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra della pagina.
3. Nel pannello di navigazione, scegli Custom key stores (Archivi delle chiavi personalizzate), External key stores (Archivi delle chiavi esterne).
4. Trova la riga che rappresenta l'archivio delle chiavi esterne da rimuovere. Se la voce Connection state (Stato connessione) dell'archivio delle chiavi esterne non è DISCONNECTED (DISCONNESSO), devi [disconnettere l'archivio delle chiavi esterne](#) prima di eliminarlo.
5. Dal menu Key store actions (Operazioni per l'archivio delle chiavi), scegli Delete (Elimina).

Se l'operazione riesce, viene visualizzato un messaggio di conferma e l'archivio delle chiavi esterne non è più visualizzato nel relativo elenco. Se l'operazione ha esito negativo, viene visualizzato un messaggio di errore che descrive il problema e fornisce istruzioni su come risolverlo. Per ulteriori informazioni, consulta [Risoluzione dei problemi relativi all'archivio delle chiavi esterne](#).

Utilizzando l'API AWS KMS

Per eliminare un archivio di chiavi esterno, utilizzare l'[DeleteCustomKeyStore](#) operazione. Se l'operazione ha esito positivo, AWS KMS restituisce una risposta HTTP 200 e un oggetto JSON senza proprietà.

Per iniziare, disconnetti l'archivio delle chiavi esterne. Prima di eseguire questo comando, sostituisci l'ID store chiavi personalizzate di esempio con uno valido.

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Dopo la disconnessione dell'archivio chiavi esterno, è possibile utilizzare l'[DeleteCustomKeyStore](#) operazione per eliminarlo.

```
$ aws kms delete-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

Per confermare che l'archivio chiavi esterno è stato eliminato, utilizzare l'[DescribeCustomKeyStores](#) operazione.

```
$ aws kms describe-custom-key-stores

{
  "CustomKeyStores": []
}
```

Se si specifica un nome o un ID di archivio chiavi personalizzato che non esiste più, AWS KMS restituisce un'CustomKeyStoreNotFoundException eccezione.

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
```

```
An error occurred (CustomKeyStoreNotFoundException) when calling the
DescribeCustomKeyStore operation:
```

Risoluzione dei problemi relativi all'archivio delle chiavi esterne

La risoluzione per la maggior parte dei problemi con gli archivi di chiavi esterni è indicata dal messaggio di errore AWS KMS visualizzato con ogni eccezione o dal [codice di errore di connessione](#) che viene AWS KMS restituito quando un tentativo di [connettere l'archivio chiavi esterno al relativo proxy dell'archivio](#) chiavi esterno ha esito negativo. Tuttavia, alcune questioni sono un po' più complesse.

Durante la diagnosi di un problema con un archivio delle chiavi esterne, individua innanzitutto la causa. Questa operazione consente di restringere la gamma di rimedi e rendere più efficiente la risoluzione dei problemi.

- **AWS KMS** — Il problema potrebbe essere interno AWS KMS, ad esempio un valore errato nella [configurazione dell'archivio chiavi esterno](#).
- **Esterno**: il problema potrebbe avere origine all'esterno AWS KMS, inclusi problemi con la configurazione o il funzionamento del proxy dell'archivio chiavi esterno, del gestore delle chiavi esterne, delle chiavi esterne o del servizio endpoint VPC.
- **Rete**: potrebbe essere un problema di connettività o di rete, ad esempio un problema con l'endpoint proxy, la porta, lo stack IP o il nome o dominio DNS privato.

Note

Quando le operazioni di gestione negli archivi delle chiavi esterne hanno esito negativo, generano diverse eccezioni. Tuttavia, le operazioni AWS KMS crittografiche si ripetono `KMSInvalidStateException` per tutti gli errori relativi alla configurazione esterna o allo stato di connessione dell'archivio di chiavi esterno. Per identificare il problema, utilizza il testo del messaggio di errore allegato.

L'[ConnectCustomKeyStore](#) operazione viene completata rapidamente prima del completamento del processo di connessione. Per determinare se il processo di connessione ha esito positivo, visualizza lo [stato della connessione](#) dell'archivio delle chiavi esterne. Se il processo di connessione fallisce, AWS KMS restituisce un [codice di errore di connessione](#) che spiega la causa e suggerisce una soluzione.

Argomenti

- [Strumenti per la risoluzione dei problemi degli archivi delle chiavi esterne](#)
- [Errori di configurazione](#)
- [Errori di connessione all'archivio delle chiavi esterne](#)
- [Errori di latenza e timeout](#)
- [Errori delle credenziali di autenticazione](#)
- [Errori relativi allo stato delle chiavi](#)
- [Errori di decrittografia](#)

- [Errori relativi alla chiave esterna](#)
- [Problemi relativi al proxy](#)
- [Problemi relativi all'autorizzazione proxy](#)

Strumenti per la risoluzione dei problemi degli archivi delle chiavi esterne

AWS KMS fornisce diversi strumenti che consentono di identificare e risolvere i problemi relativi all'archivio chiavi esterno e alle relative chiavi. Utilizza questi strumenti insieme alle funzioni fornite con il proxy dell'archivio delle chiavi esterne e il gestore delle chiavi esterne.

Note

Il proxy dell'archivio delle chiavi esterne e il gestore delle chiavi esterne potrebbero fornire metodi più semplici per creare e gestire l'archivio delle chiavi esterne e le relative chiavi KMS. Per ulteriori informazioni, consulta la documentazione degli strumenti esterni.

AWS KMS eccezioni e messaggi di errore

AWS KMS fornisce un messaggio di errore dettagliato su qualsiasi problema riscontrato. Puoi trovare ulteriori informazioni sulle AWS KMS eccezioni nell'[AWS Key Management Service API Reference](#) e. AWS SDKs Anche se utilizzi la AWS KMS console, potresti trovare utili questi riferimenti. Ad esempio, consulta l'elenco [Errori](#) per l'operazione `CreateCustomKeyStores`.

Per ottimizzare le prestazioni del proxy di archiviazione delle chiavi esterno, AWS KMS restituisce le eccezioni in base all'affidabilità del proxy entro un determinato periodo di aggregazione di 5 minuti. In caso di errore interno del server 500, servizio 503 non disponibile o timeout di connessione, viene restituito un proxy con elevata affidabilità `KMSInternalException` e attiva un nuovo tentativo automatico per garantire che le richieste vadano a buon fine. Tuttavia, viene restituito un proxy con bassa affidabilità. `KMSInvalidStateException` Per ulteriori informazioni, vedere [Monitoraggio di un archivio di chiavi esterno](#).

Se il problema si presenta in un AWS servizio diverso, ad esempio quando si utilizza una chiave KMS nell'archivio di chiavi esterno per proteggere una risorsa in un altro AWS servizio, il AWS servizio potrebbe fornire informazioni aggiuntive per aiutarti a identificare il problema. Se il AWS servizio non fornisce il messaggio, puoi visualizzare il messaggio di errore nei [CloudTrail log](#) che registrano l'uso della tua chiave KMS.

[CloudTrail logs](#)

Ogni operazione AWS KMS API, incluse le azioni nella AWS KMS console, viene registrata nei AWS CloudTrail log. AWS KMS registra una voce di registro per le operazioni riuscite e non riuscite. Per le operazioni con esito negativo, la voce di log include il nome dell'eccezione AWS KMS (`errorCode`) e il messaggio di errore (`errorMessage`). Puoi utilizzare queste informazioni per identificare e risolvere l'errore. Per vedere un esempio, consulta [Errore di decrittografia con una chiave KMS in un archivio delle chiavi esterne](#).

La voce di log include anche l'ID della richiesta. Se la richiesta ha raggiunto il proxy dell'archivio delle chiavi esterne, puoi utilizzare l'ID della richiesta nella voce di log per trovare la richiesta corrispondente nei log del proxy, se disponibili.

[CloudWatch metriche](#)

AWS KMS registra CloudWatch metriche Amazon dettagliate sul funzionamento e le prestazioni del tuo archivio di chiavi esterno, tra cui latenza, throttling, errori proxy, stato di gestore di chiavi esterno, il numero di giorni che mancano alla scadenza del certificato TLS e l'età riportata delle credenziali di autenticazione proxy. Puoi utilizzare queste metriche per sviluppare modelli di dati per il funzionamento del tuo archivio di chiavi esterno e CloudWatch allarmi che ti avvisano di problemi imminenti prima che si verifichino.

Important

AWS KMS consiglia di creare CloudWatch allarmi per monitorare le metriche dell'archivio di chiavi esterne. Questi allarmi ti avvisano dei primi segnali di problemi prima che si verifichino.

[Grafici di monitoraggio](#)

AWS KMS visualizza i grafici delle CloudWatch metriche dell'archivio chiavi esterno nella pagina di dettaglio di ogni archivio di chiavi esterno nella console. AWS KMS È possibile utilizzare i dati nei grafici per individuare l'origine degli errori, rilevare problemi imminenti, stabilire linee di base e perfezionare le soglie di allarme. CloudWatch Per informazioni dettagliate sull'interpretazione dei grafici di monitoraggio e sull'utilizzo dei relativi dati, consulta [Monitora gli archivi di chiavi esterni](#).

Visualizzazione di archivi delle chiavi esterne e chiavi KMS

AWS KMS visualizza informazioni dettagliate sugli archivi di chiavi esterni e sulle chiavi KMS nell'archivio chiavi esterno della AWS KMS console e nella risposta alle operazioni and.

[DescribeCustomKeyStoresDescribeKey](#) Queste visualizzazioni includono campi speciali per gli archivi delle chiavi esterne e le chiavi KMS con informazioni che è possibile utilizzare per la risoluzione dei problemi, come [lo stato di connessione](#) dell'archivio delle chiavi esterne e l'ID della chiave esterna associata alla chiave KMS. Per informazioni dettagliate, consultare [Visualizza gli archivi di chiavi esterni](#).

[Client di test proxy XKS](#)

AWS KMS fornisce un client di test open source che verifica che il proxy dell'archivio chiavi esterno sia conforme alla specifica dell'API [AWS KMS External Key Store Proxy](#). Puoi utilizzare tale client di test per identificare e risolvere i problemi relativi al proxy dell'archivio delle chiavi esterne.

Errori di configurazione

Durante la creazione di un archivio delle chiavi esterne, puoi specificare i valori delle proprietà che comprendono la configurazione dell'archivio, come le [credenziali di autenticazione proxy](#), l'[endpoint URI proxy](#), il [percorso URI proxy](#) e [il nome del servizio dell'endpoint VPC](#). Quando AWS KMS rileva un errore nel valore di una proprietà, l'operazione ha esito negativo e restituisce un errore che indica il valore difettoso.

Molti problemi di configurazione possono essere risolti correggendo il valore errato. Puoi correggere un percorso URI proxy o le credenziali di autenticazione proxy non valide senza disconnettere l'archivio delle chiavi esterne. Per le definizioni di questi valori, inclusi i requisiti di unicità, consulta [Assemblare i prerequisiti](#). Per istruzioni sull'aggiornamento di tali valori, consulta [Modifica delle proprietà dell'archivio chiavi esterno](#).

Per evitare errori con il percorso URI proxy e i valori delle credenziali di autenticazione proxy, quando crei o aggiorni l'archivio delle chiavi esterne, carica un [file di configurazione proxy](#) nella console AWS KMS. Si tratta di un file basato su JSON con il percorso URI proxy e i valori delle credenziali di autenticazione proxy forniti dal proxy o dal gestore delle chiavi esterne. Non è possibile utilizzare un file di configurazione proxy con le operazioni AWS KMS API, ma è possibile utilizzare i valori del file per fornire valori di parametro per le richieste API che corrispondano ai valori del proxy.

Errori di configurazione generale

Eccezioni: `CustomKeyStoreInvalidStateException` (`CreateKey`),
`KMSInvalidStateException` (operazioni di crittografia),
`XksProxyInvalidConfigurationException` (operazioni di gestione, ad eccezione di `CreateKey`)

Codici di errore di connessione: XKS_PROXY_INVALID_CONFIGURATION,
XKS_PROXY_INVALID_TLS_CONFIGURATION

Per gli archivi di chiavi esterni con [connettività pubblica degli endpoint](#), AWS KMS verifica i valori delle proprietà quando crei e aggiorni l'archivio di chiavi esterno. Per gli archivi delle chiavi esterne con [connettività del servizio endpoint VPC](#), AWS KMS testa i valori delle proprietà durante la connessione e l'aggiornamento dell'archivio delle chiavi esterne.

Note

L'operazione `ConnectCustomKeyStore`, che è asincrona, potrebbe avere esito positivo anche se il tentativo di connettere l'archivio delle chiavi esterne al relativo proxy fallisce. In tal caso non vi è alcuna eccezione, ma lo stato di connessione dell'archivio delle chiavi esterne è `Failed` (Non riuscito) e viene visualizzato un codice di errore di connessione che spiega il messaggio di errore. Per ulteriori informazioni, consulta [Errori di connessione all'archivio delle chiavi esterne](#).

Se AWS KMS rileva un errore nel valore di una proprietà, l'operazione ha esito negativo e restituisce `XksProxyInvalidConfigurationException` uno dei seguenti messaggi di errore.

Il proxy dell'archivio delle chiavi esterne ha rifiutato la richiesta a causa di un percorso URI non valido. Verifica il percorso URI dell'archivio delle chiavi esterne e aggiornalo, se necessario.

- Il [percorso URI del proxy](#) è il percorso di base per AWS KMS le richieste al proxy APIs. Se questo percorso non è corretto, tutte le richieste al proxy hanno esito negativo. Per [visualizzare il percorso URI proxy corrente](#) dell'archivio delle chiavi esterne, usa la console AWS KMS o l'operazione `DescribeCustomKeyStores`. Per trovare il percorso URI proxy corretto, consulta la documentazione del proxy dell'archivio delle chiavi esterne. Per informazioni sulla correzione del valore relativo al percorso URI proxy, consulta [Modifica delle proprietà dell'archivio chiavi esterno](#).
- Il percorso URI proxy per il proxy dell'archivio delle chiavi esterne può cambiare con gli aggiornamenti del proxy o del gestore delle chiavi esterne. Per informazioni relative a queste modifiche, consulta la documentazione del proxy o del gestore delle chiavi esterne.

XKS_PROXY_INVALID_TLS_CONFIGURATION

AWS KMS non è in grado di stabilire una connessione TLS al proxy dell'archivio delle chiavi esterne. Verifica la configurazione TLS e il relativo certificato.

- Tutti i proxy dell'archivio delle chiavi esterne richiedono un certificato TLS. Il certificato TLS deve essere emesso da un'autorità di certificazione (CA) pubblica supportata per gli archivi delle chiavi esterne. Per un elenco di quelle supportate CAs, consulta [Trusted Certificate Authorities](#) nella specifica dell'API proxy AWS KMS External Key Store.
- Per quanto riguarda la connettività dell'endpoint pubblico, il nome comune del soggetto (CN) sul certificato TLS deve corrispondere al nome di dominio nell'[endpoint URI proxy](#) per il proxy dell'archivio delle chiavi esterne. Ad esempio, se l'endpoint pubblico è `https://myproxy.xks.example.com` il TLS, il CN sul certificato TLS deve essere `myproxy.xks.example.com` o `*.xks.example.com`.
- Per la connettività del servizio endpoint VPC, il nome comune del soggetto (CN) sul certificato TLS deve corrispondere al nome DNS privato del [servizio endpoint VPC](#). Ad esempio, se il nome DNS privato è `myproxy-private.xks.example.com`, il CN sul certificato TLS deve essere `myproxy-private.xks.example.com` o `*.xks.example.com`.
- Il certificato TLS non deve essere scaduto. Per ottenere la data di scadenza di un certificato TLS, utilizza gli strumenti SSL, come [OpenSSL](#). Per monitorare la data di scadenza di un certificato TLS associato a un archivio di chiavi esterno, utilizza la [XksProxyCertificateDaysToExpire](#) CloudWatch metrica. Il numero di giorni che mancano alla data di scadenza della certificazione TLS viene visualizzato anche nella [sezione Monitoraggio della console](#). AWS KMS
- Se utilizzi la [connettività dell'endpoint pubblico](#), testa la configurazione SSL tramite gli strumenti di test SSL. Gli errori di connessione TLS possono derivare da un concatenamento errato dei certificati.

Errori di configurazione per la connettività del servizio endpoint VPC

Eccezioni: `XksProxyVpcEndpointServiceNotFoundException`,
`XksProxyVpcEndpointServiceInvalidConfigurationException`

Oltre ai problemi di connettività generali, potresti riscontrare i seguenti problemi durante la creazione, la connessione o l'aggiornamento di un archivio di chiavi esterno con connettività del servizio endpoint VPC. AWS KMS verifica i valori delle proprietà di un archivio di chiavi esterno con connettività del servizio endpoint VPC durante la [creazione](#), la [connessione](#) e l'[aggiornamento](#) dell'archivio chiavi esterno. Quando le operazioni di gestione falliscono a causa di errori di configurazione, generano le seguenti eccezioni:

XksProxyVpcEndpointServiceNotFoundException

Di seguito è riportata la possibile causa:

- Un nome del servizio endpoint VPC errato. Verifica che il nome del servizio endpoint VPC per l'archivio delle chiavi esterne sia corretto e corrisponda al valore dell'endpoint URI proxy per l'archivio delle chiavi esterne. Per trovare il nome del servizio endpoint VPC, usa la console Amazon [VPC](#) o l'operazione. [DescribeVpcEndpointServices](#) Per trovare il nome del servizio endpoint VPC e l'endpoint URI proxy di un key store esterno esistente, usa la AWS KMS console o l'operazione. [DescribeCustomKeyStores](#) Per informazioni dettagliate, consultare [Visualizza gli archivi di chiavi esterni](#).
- Il servizio endpoint VPC potrebbe trovarsi in un archivio chiavi Regione AWS diverso da quello esterno. Verifica che il servizio endpoint VPC e l'archivio delle chiavi esterne si trovino nella stessa regione. (Il nome esterno del nome della regione, ad esempio, fa parte del nome del servizio endpoint VPCus-east-1, ad esempio com.amazonaws.vpce.us-east-1.vpce-svc-example.) Per un elenco dei requisiti per il servizio endpoint VPC di un archivio delle chiavi esterne, consulta [Servizio endpoint VPC](#). Non puoi spostare un servizio endpoint VPC o un archivio delle chiavi esterne in una regione diversa, tuttavia puoi creare un nuovo archivio delle chiavi esterne nella stessa regione del servizio endpoint VPC. Per informazioni dettagliate, consulta [Configurazione della connettività del servizio endpoint VPC](#) e [Creare un archivio di chiavi esterno](#).
- AWS KMS non è un principale consentito per il servizio endpoint VPC. L'elenco Allow principals (Consenti entità principali) per il servizio endpoint VPC deve includere il valore cks.kms.<region>.amazonaws.com, ad esempio cks.kms.eu-west-3.amazonaws.com. Per istruzioni sull'aggiunta di questo valore, consulta [Gestione delle autorizzazioni](#) nella Guida di AWS PrivateLink.

XksProxyVpcEndpointServiceInvalidConfigurationException

Questo errore si verifica quando il servizio endpoint VPC non soddisfa uno dei seguenti requisiti:

- Il VPC richiede almeno due sottoreti private, ognuna in una zona di disponibilità diversa. Per assistenza sull'aggiunta di una sottorete al VPC, consulta [Creazione di una sottorete nel VPC](#) nella Guida per l'utente di Amazon VPC.

- Il [tipo di servizio endpoint VPC](#) deve utilizzare un Network Load Balancer, non un sistema di bilanciamento del carico gateway.
- L'accettazione non deve essere richiesta per il servizio endpoint VPC (Acceptance required [Accettazione richiesta] deve essere false). Se è richiesta l'accettazione manuale di ogni richiesta di connessione, AWS KMS non è possibile utilizzare il servizio endpoint VPC per connettersi al proxy dell'archivio chiavi esterno. Per maggiori dettagli, consulta [Accettare o rifiutare le richieste di connessione](#) nella Guida di AWS PrivateLink .
- Il servizio endpoint VPC deve avere un nome DNS privato che è un sottodominio di un dominio pubblico. Ad esempio, se il nome DNS privato è `https://myproxy-private.xks.example.com`, i domini `xks.example.com` o `example.com` devono disporre di un server DNS pubblico. Per visualizzare o modificare il nome DNS privato per il servizio endpoint VPC, consulta [Gestione dei nomi DNS per i servizi endpoint VPC](#) nella Guida di AWS PrivateLink .
- Il valore di Domain verification status (Stato di verifica del dominio) per il nome DNS privato deve essere `verified`. Per visualizzare e aggiornare lo stato di verifica del dominio del nome DNS privato, consulta [Fase 5: Verifica il tuo nome di dominio DNS privato](#). Potrebbero essere necessari alcuni minuti prima che venga visualizzato lo stato di verifica aggiornato dopo aver aggiunto il record di testo richiesto.

 Note

Un dominio DNS privato può essere verificato solo se è il sottodominio di un dominio pubblico. In caso contrario, lo stato di verifica del dominio DNS privato non cambia, anche dopo aver aggiunto il record TXT richiesto.

- Assicurati che tutti i firewall tra AWS KMS e il proxy dell'archivio chiavi esterno consentano il traffico da e verso la porta 443 del proxy. AWS KMS comunica sulla porta 443 tramite IPv4. Questo valore non è configurabile.
- Il nome DNS privato del servizio endpoint VPC deve corrispondere al valore dell'[endpoint URI proxy](#) per l'archivio delle chiavi esterne. Per un archivio delle chiavi esterne con connettività del servizio endpoint VPC, l'endpoint URI proxy deve essere `https://` seguito dal nome DNS privato del servizio endpoint VPC. Per visualizzare il valore dell'endpoint URI proxy, consulta [Visualizza gli archivi di chiavi esterni](#). Per modificare il valore dell'endpoint URI proxy, consulta [Modifica delle proprietà dell'archivio chiavi esterno](#).

Errori di connessione all'archivio delle chiavi esterne

Il [processo di connessione di un archivio delle chiavi esterne](#) al relativo proxy richiede circa cinque minuti. Se l'operazione non genera rapidamente un errore, l'operazione `ConnectCustomKeyStore` restituisce una risposta HTTP 200 e un oggetto JSON senza proprietà. Questa risposta iniziale non indica tuttavia che la connessione è riuscita. Per determinare se l'archivio delle chiavi esterne è connesso, visualizza lo [stato della connessione](#). Se la connessione fallisce, lo stato di connessione dell'archivio chiavi esterno cambia `FAILED` e AWS KMS restituisce un [codice di errore di connessione](#) che spiega la causa dell'errore.

Note

Quando lo stato di connessione di un archivio delle chiavi personalizzate è `FAILED`, devi disconnettere l'archivio prima di tentare di riconnetterlo. Non puoi connettere uno store delle chiavi personalizzate il cui stato di connessione è `FAILED`.

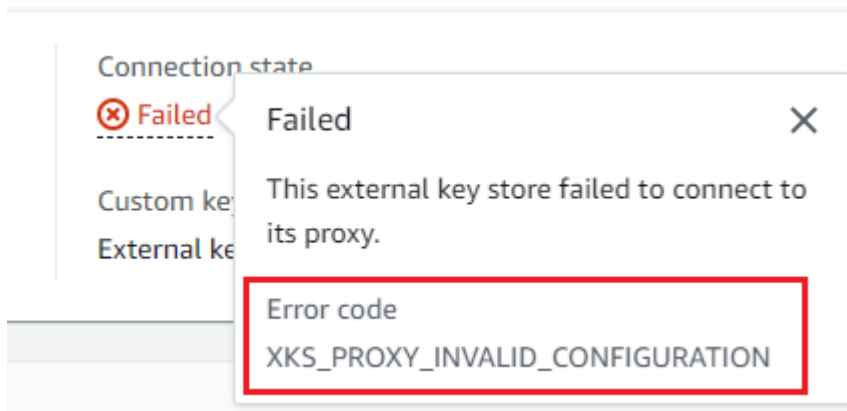
Per visualizzare lo stato di connessione di un archivio delle chiavi esterne:

- Nella [DescribeCustomKeyStores](#) risposta, visualizza il valore dell'`ConnectionState` elemento.
- Nella AWS KMS console, lo stato della connessione viene visualizzato nella tabella dell'archivio chiavi esterno. Inoltre, nella pagina dei dettagli di ogni archivio delle chiavi esterne, il campo `Connection state` (Stato connessione) viene visualizzato nella sezione `General configuration` (Configurazione generale).

Quando lo stato della connessione è `FAILED`, il codice di errore di connessione indica l'errore.

Per visualizzare il codice di errore della connessione:

- Nella [DescribeCustomKeyStores](#) risposta, visualizza il valore dell'`ConnectionErrorCode` elemento. Tale elemento appare nella risposta `DescribeCustomKeyStores` solo quando `ConnectionState` è nello stato `FAILED`.
- Per visualizzare il codice di errore di connessione nella AWS KMS console, vai alla pagina di dettaglio dell'archivio chiavi esterno e passa il mouse sul valore `Failed`.



Codici di errore di connessione per archivi delle chiavi esterne

I seguenti codici di errore di connessione si applicano agli archivi delle chiavi esterne

INTERNAL_ERROR

AWS KMS impossibile completare la richiesta a causa di un errore interno. Riprova la richiesta . Per le richieste ConnectCustomKeyStore, scollega lo store delle chiavi personalizzate prima di provare a connetterti di nuovo.

INVALID_CREDENTIALS

Uno o entrambi i valori XksProxyAuthenticationCredential non sono validi nel proxy dell'archivio delle chiavi esterne specificato.

NETWORK_ERRORS

Gli errori di rete AWS KMS impediscono la connessione dell'archivio chiavi personalizzato al relativo archivio chiavi di supporto.

XKS_PROXY_ACCESS_DENIED

AWS KMS alle richieste viene negato l'accesso al proxy dell'archivio chiavi esterno. Se tale proxy dispone di regole di autorizzazione, verifica che consentano ad AWS KMS di comunicare con il proxy per tuo conto.

XKS_PROXY_INVALID_CONFIGURATION

Un errore di configurazione impedisce all'archivio delle chiavi esterne di connettersi al relativo proxy. Verifica il valore di XksProxyUriPath.

XKS_PROXY_INVALID_RESPONSE

AWS KMS non è in grado di interpretare la risposta dal proxy dell'archivio chiavi esterno. Se visualizzi ripetutamente questo codice di errore di connessione, informa il fornitore del proxy dell'archivio delle chiavi esterne.

XKS_PROXY_INVALID_TLS_CONFIGURATION

AWS KMS non può connettersi al proxy dell'archivio chiavi esterno perché la configurazione TLS non è valida. Verifica che il proxy dell'archivio delle chiavi esterne supporti TLS 1.2 o 1.3. Inoltre, verifica che il certificato TLS sia ancora valido, che corrisponda al nome host nel valore `XksProxyUriEndpoint` e che sia firmato da un'autorità di certificazione affidabile inclusa nell'elenco delle [autorità di certificazione attendibili](#).

XKS_PROXY_NOT_REACHABLE

AWS KMS non riesce a comunicare con il proxy dell'archivio chiavi esterno. Verifica che `XksProxyUriEndpoint` e `XksProxyUriPath` siano corretti. Utilizza gli strumenti del proxy dell'archivio delle chiavi esterne per verificare che il proxy sia attivo e disponibile sulla rete. Inoltre, verifica che le istanze del gestore delle chiavi esterne funzionino correttamente. I tentativi di connessione hanno esito negativo e restituiscono questo codice di errore di connessione se il proxy segnala che tutte le istanze del gestore delle chiavi esterne non sono disponibili.

XKS_PROXY_TIMED_OUT

AWS KMS può connettersi al proxy dell'archivio di chiavi esterno, ma il proxy non risponde AWS KMS nel tempo assegnato. Se visualizzi ripetutamente questo codice di errore di connessione, informa il fornitore del proxy dell'archivio delle chiavi esterne.

XKS_VPC_ENDPOINT_SERVICE_INVALID_CONFIGURATION

La configurazione del servizio endpoint Amazon VPC non è conforme ai requisiti per un AWS KMS archivio di chiavi esterno.

- Il servizio endpoint VPC deve essere un servizio endpoint per gli endpoint di interfaccia nell'Account AWS del chiamante.
- Deve avere un Network Load Balancer (NLB) connesso ad almeno due sottoreti, ognuna in una zona di disponibilità diversa.
- L'`Allow principal` elenco deve includere il AWS KMS servizio principale per la regione `cks.kms.<region>.amazonaws.com`, ad esempio `cks.kms.us-east-1.amazonaws.com`
- Non deve richiedere l'[accettazione](#) delle richieste di connessione.

- Deve avere un nome DNS privato. Il nome DNS privato per un archivio delle chiavi esterne con connettività VPC_ENDPOINT_SERVICE deve essere univoco nella Regione AWS.
- Lo [stato di verifica](#) per il dominio del nome DNS privato deve essere `verified`.
- Il [certificato TLS](#) specifica il nome host DNS privato in cui è raggiungibile l'endpoint.

XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND

AWS KMS non riesce a trovare il servizio endpoint VPC che utilizza per comunicare con il proxy di archiviazione chiavi esterno. Verifica che `XksProxyVpcEndpointServiceName` sia corretto e che il principale del servizio AWS KMS disponga delle autorizzazioni consumer per il servizio endpoint Amazon VPC.

Errori di latenza e timeout

Eccezioni: `CustomKeyStoreInvalidStateException (CreateKey)`,
`KMSInvalidStateException` (operazioni di crittografia),
`XksProxyUriUnreachableException` (operazioni di gestione)

[Codici di errore di connessione](#): `XKS_PROXY_NOT_REACHABLE`, `XKS_PROXY_TIMED_OUT`

Quando non AWS KMS riesce a contattare il proxy entro l'intervallo di timeout di 250 millisecondi, restituisce un'eccezione. `CreateCustomKeyStoreXksProxyUriUnreachableException` torna. `UpdateCustomKeyStore` Le operazioni crittografiche restituiscono lo standard `KMSInvalidStateException` con un messaggio di errore che descrive il problema. Se `ConnectCustomKeyStore` fallisce, AWS KMS restituisce un [codice di errore di connessione](#) che descrive il problema.

Gli errori di timeout possono essere problemi temporanei che possono essere risolti ripetendo la richiesta. Se il problema persiste, verifica che il proxy dell'archivio delle chiavi esterne sia attivo e connesso alla rete e che l'endpoint URI proxy, il percorso URI proxy e il nome del servizio endpoint VPC (se presente) siano corretti. Inoltre, verifica che il gestore delle chiavi esterno sia vicino Regione AWS all'archivio delle chiavi esterno. Se è necessario aggiornare uno di questi valori, consulta [Modifica delle proprietà dell'archivio chiavi esterno](#).

Per tenere traccia dei modelli di latenza, utilizza la [XksProxyLatency](#) CloudWatch metrica e il grafico della latenza media (basato su tale metrica) nella [sezione Monitoraggio della console](#). AWS KMS Il proxy dell'archivio delle chiavi esterne potrebbe anche generare log e parametri in grado di tracciare la latenza e i timeout.

XksProxyUriUnreachableException

AWS KMS non è in grado di comunicare con il proxy dell'archivio chiavi esterno. Potrebbe trattarsi di un problema di rete temporaneo. Se visualizzi questo errore ripetutamente, verifica che il proxy dell'archivio delle chiavi esterne sia attivo e connesso alla rete e che l'URI endpoint sia corretto.

- Il proxy dell'archivio chiavi esterno non ha risposto a una richiesta API AWS KMS proxy entro l'intervallo di timeout di 250 millisecondi. Ciò potrebbe indicare un problema di rete temporaneo o un problema operativo o di prestazioni con il proxy. Se un nuovo tentativo non risolve il problema, informa l'amministratore del proxy dell'archivio delle chiavi esterne.

Gli errori di latenza e timeout si manifestano spesso come errori di connessione. Quando l'[ConnectCustomKeyStore](#) operazione ha esito negativo, lo stato di connessione dell'archivio chiavi esterno cambia FAILED e AWS KMS restituisce un codice di errore di connessione che spiega l'errore. Per un elenco di codici di errore di connessione e suggerimenti per la relativa risoluzione, consulta [Codici di errore di connessione per archivi delle chiavi esterne](#). Gli elenchi dei codici di connessione per All custom key stores (Tutti gli archivi delle chiavi personalizzate) e External key stores (Archivi delle chiavi esterne) si applicano agli archivi delle chiavi esterne. I seguenti errori di connessione sono correlati alla latenza e ai timeout.

XKS_PROXY_NOT_REACHABLE

oppure

CustomKeyStoreInvalidStateException , KMSInvalidStateException ,
XksProxyUriUnreachableException

AWS KMS non può comunicare con il proxy dell'archivio chiavi esterno. Verifica che il proxy dell'archivio delle chiavi esterne sia attivo e connesso alla rete e che il percorso URI e l'URI endpoint o il nome del servizio VPC siano corretti nell'archivio delle chiavi esterne.

Questo errore può verificarsi per i seguenti motivi:

- Il proxy dell'archivio delle chiavi esterne non è attivo o non è connesso alla rete.
- Si è verificato un errore nei valori [endpoint URI proxy](#), [percorso URI proxy](#) o del [nome del servizio endpoint VPC](#) (se applicabile) nella configurazione dell'archivio delle chiavi esterne. Per visualizzare la configurazione dell'archivio chiavi esterno, utilizza

l'[DescribeCustomKeyStores](#) operazione o [visualizza la pagina di dettaglio](#) dell'archivio chiavi esterno nella AWS KMS console.

- Potrebbe esserci un errore di configurazione di rete, ad esempio un errore di porta, nel percorso di rete tra AWS KMS e il proxy dell'archivio chiavi esterno. AWS KMS comunica con il proxy dell'archivio chiavi esterno sulla porta 443 tramite IPv4. Questo valore non è configurabile.
- Quando il proxy dell'archivio chiavi esterno segnala (in [GetHealthStatus](#) risposta) che tutte le istanze del gestore di chiavi esterno lo sono UNAVAILABLE, l'[ConnectCustomKeyStore](#) operazione fallisce e restituisce un valore di `ConnectionErrorCode XKS_PROXY_NOT_REACHABLE`. Per assistenza, consulta la documentazione del gestore delle chiavi esterne.
- Questo errore può derivare da una lunga distanza fisica tra il gestore di chiavi esterno e Regione AWS l'archivio chiavi esterno. La latenza del ping (network round-trip time (RTT)) tra il gestore delle chiavi Regione AWS e quello esterno non deve superare i 35 millisecondi. Potrebbe essere necessario creare un archivio di chiavi esterno in un Regione AWS data center più vicino al gestore di chiavi esterno o spostare il gestore di chiavi esterno in un data center più vicino al. Regione AWS

`XKS_PROXY_TIMED_OUT`

oppure

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` ,
`XksProxyUriUnreachableException`

AWS KMS ha rifiutato la richiesta perché il proxy dell'archivio delle chiavi esterne non ha risposto in tempo. Riprova la richiesta . Se visualizzi questo errore ripetutamente, segnalalo all'amministratore del proxy dell'archivio delle chiavi esterne.

Questo errore può verificarsi per i seguenti motivi:

- Questo errore può essere causato da una grande distanza fisica tra il gestore delle chiavi esterne e il proxy dell'archivio delle chiavi esterne. Se possibile, avvicina il proxy al gestore delle chiavi esterne.
- Gli errori di timeout possono verificarsi quando il proxy non è progettato per gestire il volume e la frequenza delle richieste da AWS KMS. Se le tue CloudWatch metriche indicano un problema persistente, avvisa l'amministratore proxy dell'archivio chiavi esterno.

- Gli errori di timeout possono verificarsi quando la connessione tra il gestore delle chiavi esterne e Amazon VPC per l'archivio delle chiavi esterne non funziona correttamente. Se lo utilizzi AWS Direct Connect, verifica che il tuo VPC e il gestore di chiavi esterno possano comunicare in modo efficace. Per assistenza nella risoluzione di eventuali problemi, consulta [Risoluzione dei problemi AWS Direct Connect](#) nella Guida per l'utente Direct Connect.

XKS_PROXY_TIMED_OUT

oppure

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` ,
`XksProxyUriUnreachableException`

Il proxy dell'archivio delle chiavi esterne non ha risposto alla richiesta nel tempo assegnato.

Riprova la richiesta . Se visualizzi questo errore ripetutamente, segnalalo all'amministratore del proxy dell'archivio delle chiavi esterne.

- Questo errore può essere causato da una grande distanza fisica tra il gestore delle chiavi esterne e il proxy dell'archivio delle chiavi esterne. Se possibile, avvicina il proxy al gestore delle chiavi esterne.

Errori delle credenziali di autenticazione

Eccezioni: `CustomKeyStoreInvalidStateException` (`CreateKey`),
`KMSInvalidStateException` (operazioni di crittografia),
`XksProxyIncorrectAuthenticationCredentialException` (operazioni di gestione diverse da `CreateKey`)

L'utente stabilisce e mantiene una credenziale di autenticazione per AWS KMS il proxy di archiviazione delle chiavi esterno. Quindi, quando AWS KMS crei un archivio di chiavi esterno, indichi i valori delle credenziali. Per modificare le credenziali di autenticazione, esegui questa operazione nel proxy dell'archivio delle chiavi esterne. Quindi [aggiorna le credenziali](#) per l'archivio delle chiavi esterne. Se il proxy effettua la rotazione delle credenziali, devi [aggiornarle](#).

Se il proxy dell'archivio delle chiavi esterne non autentica una richiesta firmata con le [credenziali di autenticazione proxy](#) per l'archivio delle chiavi esterne, l'effetto dipende dalla richiesta:

- `CreateCustomKeyStore` e `UpdateCustomKeyStore` hanno esito negativo con un'eccezione `XksProxyIncorrectAuthenticationCredentialException`.
- `ConnectCustomKeyStore` ha esito positivo, ma la connessione fallisce. Lo stato della connessione è `FAILED` e il codice di errore è `INVALID_CREDENTIALS`. Per informazioni dettagliate, consultare [Errori di connessione all'archivio delle chiavi esterne](#).
- Le operazioni crittografiche restituiscono tutti `KMSInvalidStateException` gli errori di configurazione esterni e gli errori dello stato di connessione in un archivio di chiavi esterno. Il messaggio di errore allegato descrive il problema.

Il proxy dell'archivio delle chiavi esterne ha rifiutato la richiesta perché non era in grado di autenticare AWS KMS. Verifica le credenziali dell'archivio delle chiavi esterne e aggiornale se necessario.

Questo errore può verificarsi per i seguenti motivi:

- L'ID della chiave di accesso o la chiave di accesso segreta per l'archivio delle chiavi esterne non corrisponde ai valori stabiliti nel proxy.

Per correggere questo errore, [aggiorna le credenziali di autenticazione proxy](#) per l'archivio delle chiavi esterne. Puoi apportare questa modifica senza disconnettere l'archivio delle chiavi esterne.

- Un proxy inverso tra AWS KMS e il proxy dell'archivio di chiavi esterno potrebbe manipolare le intestazioni HTTP in modo da invalidare le firme SigV4. Per correggere questo errore, informa l'amministratore del proxy.

Errori relativi allo stato delle chiavi

Eccezioni: `KMSInvalidStateException`

`KMSInvalidStateException` viene utilizzato per due scopi distinti per le chiavi KMS negli archivi delle chiavi personalizzate.

- Quando un'operazione di gestione, ad esempio `CancelKeyDeletion`, ha esito negativo e restituisce questa eccezione, indica che lo [stato](#) della chiave KMS non è compatibile con l'operazione.

- Quando un'[operazione di crittografia](#) su una chiave KMS in un archivio delle chiavi personalizzate ha esito negativo e restituisce un'eccezione `KMSInvalidStateException`, può indicare un problema con lo stato della chiave KMS. Tuttavia, le operazioni AWS KMS crittografiche restituiscono tutti gli errori di configurazione esterni e `KMSInvalidStateException` gli errori dello stato di connessione in un archivio di chiavi esterno. Per identificare il problema, utilizza il messaggio di errore che accompagna l'eccezione.

Per trovare lo stato della chiave richiesto per le operazioni di un' AWS KMS API, vedere [Stati chiave delle AWS KMS chiavi](#). Per trovare lo stato di chiave di una chiave KMS, nella pagina Chiavi gestite cliente, visualizza il campo Stato della chiave KMS. In alternativa, utilizza l'[DescribeKey](#) operazione e visualizzate l'`KeyState` elemento nella risposta. Per informazioni dettagliate, consultare [Identifica e visualizza le chiavi](#).

Note

Lo stato di una chiave KMS in un archivio delle chiavi esterne non indica lo stato della [chiave esterna](#) associata. Per informazioni sullo stato della chiave esterna, usa il gestore delle chiavi esterne e gli strumenti del proxy dell'archivio delle chiavi esterne.

`CustomKeyStoreInvalidStateException` si riferisce allo [stato di connessione](#) dell'archivio delle chiavi esterne, non allo [stato chiave](#) di una chiave KMS.

Un'operazione di crittografia su una chiave KMS in un archivio personalizzato potrebbe non riuscire perché lo stato della chiave KMS è `Unavailable` o `PendingDeletion`. (I tasti disattivati restituiscono `DisabledException`).

- Una chiave KMS ha uno stato `Disabled` chiave solo quando la disabiliti intenzionalmente nella AWS KMS console o utilizzando l'operazione. [DisableKey](#) Quando una chiave KMS è disabilitata, puoi visualizzare e gestire la chiave, ma non utilizzarla per operazioni di crittografia. Per risolvere il problema, abilita la chiave. Per informazioni dettagliate, consultare [Attivazione e disattivazione delle chiavi](#).
- Una chiave KMS ha uno stato chiave `Unavailable` quando l'archivio delle chiavi esterne viene disconnesso dal relativo proxy. Per correggere una chiave KMS non disponibile, [riconnetti l'archivio delle chiavi esterne](#). Dopo la riconnessione, per le chiavi KMS nell'archivio delle chiavi esterne viene ripristinato lo stato di chiave precedente, ovvero `Enabled` o `Disabled`.

Una chiave KMS ha uno stato di chiave PendingDeletion quando ne è stata programmata l'eliminazione e si trova nel periodo di attesa. Un errore di stato della chiave su una chiave KMS in attesa di eliminazione indica che la chiave non deve essere eliminata, perché viene utilizzata per la crittografia o è necessaria per la decrittografia. Per riattivare la chiave KMS, annulla l'eliminazione pianificata e [abilita la chiave](#). Per informazioni dettagliate, consultare [Pianifica l'eliminazione delle chiavi](#).

Errori di decrittografia

Eccezioni: KMSInvalidStateException

Quando un'operazione di [decrittografia](#) con una chiave KMS in un archivio di chiavi esterno non riesce, AWS KMS restituisce lo standard KMSInvalidStateException utilizzato dalle operazioni di crittografia per tutti gli errori di configurazione esterni e gli errori dello stato di connessione su un archivio di chiavi esterno. Il messaggio di errore indica il problema.

Per decrittografare un testo criptato con [doppia crittografia](#), il gestore delle chiavi esterne utilizza prima la chiave esterna per decrittografare il livello esterno. Quindi AWS KMS utilizza il materiale AWS KMS chiave contenuto nella chiave KMS per decrittografare lo strato interno del testo cifrato. Un testo criptato non valido o danneggiato può essere rifiutato dal gestore delle chiavi esterne o da AWS KMS.

I seguenti messaggi di errore accompagnano KMSInvalidStateException quando la decrittografia ha esito negativo. Indica un problema con il testo criptato o il contesto di crittografia opzionale nella richiesta.

Il proxy dell'archivio delle chiavi esterne ha rifiutato la richiesta perché il testo criptato specificato o i dati autenticati aggiuntivi sono danneggiati, mancanti o non validi.

- Quando il proxy dell'archivio chiavi esterno o il gestore di chiavi esterno segnala che un testo cifrato o il relativo contesto di crittografia non sono validi, in genere indica un problema con il testo cifrato o il contesto di crittografia nella richiesta inviata a Decrypt AWS KMS. Per Decrypt le operazioni, AWS KMS invia al proxy lo stesso testo cifrato e lo stesso contesto di crittografia che riceve nella richiesta. Decrypt

Questo errore potrebbe essere causato da un problema di rete in transito, ad esempio un bit capovolto. Riprova la richiesta Decrypt. Se il problema persiste, verifica che il testo criptato

non sia stato alterato o danneggiato. Inoltre, verifica che il contesto di crittografia nella Decrypt richiama AWS KMS corrisponda al contesto di crittografia nella richiesta che ha crittografato i dati.

Il testo criptato o il contesto di crittografia inviato dal proxy dell'archivio delle chiavi esterne per la decrittografia è danneggiato, mancante o non valido.

- Quando AWS KMS rifiuta il testo cifrato ricevuto dal proxy, indica che il gestore delle chiavi o il proxy esterno ha restituito un testo cifrato non valido o danneggiato a. AWS KMS

Questo errore potrebbe essere causato da un problema di rete in transito, ad esempio un bit capovolto. Riprova la richiesta Decrypt. Se il problema persiste, verifica che il gestore di chiavi esterno funzioni correttamente e che il proxy dell'archivio chiavi esterno non alteri il testo cifrato ricevuto dal gestore di chiavi esterno prima di restituirlo. AWS KMS

Errori relativi alla chiave esterna

Una [chiave esterna](#) è una chiave crittografica nel gestore delle chiavi esterne che funge da materiale della chiave per una chiave KMS. AWS KMS non è in grado di accedere direttamente alla chiave esterna, ma deve chiedere al gestore delle chiavi esterne (tramite il proxy dell'archivio delle chiavi esterne) di utilizzare la chiave esterna per crittografare i dati o decrittografare un testo criptato.

L'ID della chiave esterna viene specificato nel relativo gestore durante la creazione di una chiave KMS nell'archivio delle chiavi esterne. Non puoi modificare l'ID della chiave esterna dopo la creazione della chiave KMS. Per evitare problemi con la chiave KMS, puoi utilizzare l'operazione CreateKey per chiedere al proxy dell'archivio delle chiavi esterne di verificare l'ID e la configurazione della chiave esterna. Se la chiave esterna non [soddisfa i requisiti](#) per l'uso con una chiave KMS, l'operazione CreateKey ha esito negativo con un'eccezione e un messaggio di errore che identifica il problema.

Tuttavia, possono verificarsi problemi dopo la creazione della chiave KMS. Se un'operazione di crittografia fallisce a causa di un problema con la chiave esterna, l'operazione ha esito negativo e restituisce un'eccezione KMSInvalidStateException con un messaggio di errore che indica il problema.

CreateKey errori per la chiave esterna

Eccezioni: XksKeyAlreadyInUseException, XksKeyNotFoundException, XksKeyInvalidConfigurationException

L'[CreateKey](#) operazione tenta di verificare l'ID e le proprietà della chiave esterna fornita nel parametro External key ID (console) o XksKeyId (API). Questa procedura è progettata per rilevare gli errori in anticipo prima di provare a utilizzare la chiave esterna con la chiave KMS.

Chiave esterna in uso

Ogni chiave KMS in un archivio delle chiavi esterne deve utilizzare una chiave esterna diversa. Quando `CreateKey` riconosce che l'ID della chiave esterna (`XksKeyId`) per una chiave KMS non è univoco nell'archivio delle chiavi esterne, fallisce e restituisce un `XksKeyAlreadyInUseException`.

Se ne usi più di una IDs per la stessa chiave esterna, `CreateKey` non riconoscerà la chiave duplicata. Tuttavia, le chiavi KMS con la stessa chiave esterna non sono interoperabili perché hanno materiali AWS KMS chiave e metadati diversi.

Chiave esterna non trovata

Quando il proxy dell'archivio chiavi esterno segnala di non riuscire a trovare la chiave esterna utilizzando l'ID della chiave esterna (`XksKeyId`) per la chiave KMS, l'`CreateKey` operazione fallisce e restituisce il seguente messaggio `XksKeyNotFoundException` di errore.

Il proxy dell'archivio delle chiavi esterne ha rifiutato la richiesta perché non riusciva a trovare la chiave esterna.

Questo errore può verificarsi per i seguenti motivi:

- L'ID della chiave esterna (`XksKeyId`) per la chiave KMS potrebbe non essere valido. Per individuare l'ID utilizzato dal proxy dell'archivio delle chiavi esterne per identificare la chiave esterna, consulta la documentazione del proxy o del gestore delle chiavi esterne.
- La chiave esterna potrebbe essere stata eliminata dal gestore delle chiavi esterne. Per verificare, utilizza gli strumenti del gestore delle chiavi esterne. Se la chiave esterna viene eliminata definitivamente, usa una chiave esterna diversa con la chiave KMS. Per un elenco dei requisiti per la chiave esterna, consulta [Requisiti per una chiave KMS in un archivio delle chiavi esterne](#).

Requisiti della chiave esterna non soddisfatti

Quando il proxy dell'archivio delle chiavi esterne segnala che la chiave esterna non [soddisfa i requisiti](#) per l'uso con una chiave KMS, l'operazione `CreateKey` ha esito negativo e restituisce l'eccezione `XksKeyInvalidConfigurationException` con uno dei seguenti messaggi di errore.

La specifica della chiave per la chiave esterna deve essere `AES_256`. La specifica chiave della chiave esterna specificata è `<key-spec>`.

- La chiave esterna deve essere una chiave crittografica simmetrica a 256 bit con una specifica della chiave `AES_256`. Se la chiave esterna specificata è di tipo diverso, specifica l'ID di una chiave esterna che soddisfi questo requisito.

Lo stato della chiave esterna deve essere `ENABLED` (ABILITATO). Lo stato della chiave esterna specificata è `<status>`.

- La chiave esterna deve essere abilitata nel gestore delle chiavi esterne. Se la chiave esterna specificata non è abilitata, utilizza gli strumenti del gestore delle chiavi esterne per abilitarla o specifica una chiave esterna abilitata.

L'utilizzo della chiave per la chiave esterna deve includere `ENCRYPT` e `DECRYPT`. L'uso chiave della chiave esterna specificata è `<key-usage>`.

- La chiave esterna deve essere configurata per la crittografia e la decrittografia nel gestore delle chiavi esterne. Se la chiave esterna specificata non include queste operazioni, utilizza gli strumenti del gestore delle chiavi esterne per modificare le operazioni o specifica una chiave esterna diversa.

Errori relativi alle operazioni di crittografia per la chiave esterna

Eccezioni: `KMSInvalidStateException`

Quando il proxy dell'archivio delle chiavi esterne non riesce a trovare la chiave esterna associata alla chiave KMS o la chiave esterna non [soddisfa i requisiti](#) per l'uso con una chiave KMS, l'operazione di crittografia ha esito negativo.

I problemi relativi alla chiave esterna rilevati durante un'operazione di crittografia sono più difficili da risolvere rispetto ai problemi rilevati prima della creazione della chiave KMS. Non puoi modificare l'ID della chiave esterna dopo la creazione della chiave KMS. Se la chiave KMS non ha ancora crittografato alcun dato, puoi eliminare la chiave KMS e crearne una nuova con un ID diverso. Tuttavia, il testo cifrato generato con la chiave KMS non può essere decrittografato da nessun'altra chiave KMS, nemmeno da una con la stessa chiave esterna, poiché le chiavi avranno metadati chiave diversi e materiale chiave diverso. AWS KMS Al contrario, utilizza per quanto possibile gli strumenti del gestore delle chiavi esterne per risolvere il problema con la chiave esterna.

Quando il proxy dell'archivio delle chiavi esterne segnala un problema con la chiave esterna, le operazioni di crittografia restituiscono l'eccezione `KMSInvalidStateException` con un messaggio di errore che identifica il problema.

Chiave esterna non trovata

Quando il proxy dell'archivio di chiavi esterno segnala di non riuscire a trovare la chiave esterna utilizzando l'ID della chiave esterna (`XksKeyId`) per la chiave KMS, le operazioni crittografiche restituiscono un messaggio di errore con il seguente messaggio di errore.

`KMSInvalidStateException`

Il proxy dell'archivio delle chiavi esterne ha rifiutato la richiesta perché non riusciva a trovare la chiave esterna.

Questo errore può verificarsi per i seguenti motivi:

- L'ID della chiave esterna (`XksKeyId`) per la chiave KMS non è più valido.

Per trovare l'ID della chiave esterna associato alla chiave KMS, [visualizza i dettagli della chiave KMS](#). Per individuare l'ID utilizzato dal proxy dell'archivio delle chiavi esterne per identificare la chiave esterna, consulta la documentazione del proxy o del gestore delle chiavi esterne.

AWS KMS verifica l'ID della chiave esterna quando crea una chiave KMS in un archivio di chiavi esterno. Tuttavia, l'ID potrebbe non essere valido, soprattutto se il valore dell'ID della chiave esterna è un alias o un nome modificabile. Non puoi modificare l'ID della chiave esterna associato a una chiave KMS esistente. Per decrittografare un testo criptato generato con la chiave KMS, devi associare nuovamente la chiave esterna all'ID della chiave esterna esistente.

Se non hai ancora utilizzato la chiave KMS per crittografare i dati, puoi creare una nuova chiave KMS con un ID della chiave esterna valido. Tuttavia, se hai generato un testo criptato con la chiave KMS, non puoi usare nessun'altra chiave KMS per decrittografarlo, anche se utilizza la stessa chiave esterna.

- La chiave esterna potrebbe essere stata eliminata dal gestore delle chiavi esterne. Per verificare, utilizza gli strumenti del gestore delle chiavi esterne. Se possibile, prova a [recuperare il materiale della chiave](#) da una copia o da un backup del gestore delle chiavi esterne. Se la chiave esterna viene eliminata definitivamente, qualsiasi testo criptato generato con la chiave KMS associata è irrecuperabile.

Errori di configurazione della chiave esterna

Quando il proxy dell'archivio delle chiavi esterne segnala che la chiave esterna non [soddisfa i requisiti](#) per l'uso con una chiave KMS, l'operazione di crittografia genera l'eccezione `KMSInvalidStateException` con uno dei seguenti messaggi di errore.

Il proxy dell'archivio delle chiavi esterne ha rifiutato la richiesta perché la chiave esterna non supporta l'operazione richiesta.

- La chiave esterna deve supportare sia la crittografia che la decrittografia. Se l'utilizzo della chiave non include queste due operazioni, utilizza gli strumenti del gestore delle chiavi esterne per modificarlo.

Il proxy dell'archivio delle chiavi esterne ha rifiutato la richiesta perché la chiave esterna non è abilitata nel gestore delle chiavi esterne.

- La chiave esterna deve essere abilitata e disponibile per l'uso nel gestore delle chiavi esterne. Se lo stato della chiave esterna non è `Enabled`, utilizza gli strumenti del gestore delle chiavi esterne per abilitarlo.

Problemi relativi al proxy

Eccezioni:

`CustomKeyStoreInvalidStateException` (`CreateKey`), `KMSInvalidStateException` (operazioni di crittografia), `UnsupportedOperationException`, `XksProxyUriUnreachableException`, `XksProxyInvalidResponseException` (operazioni di gestione diverse da `CreateKey`)

Il proxy dell'archivio chiavi esterno media tutte le comunicazioni tra AWS KMS e il gestore di chiavi esterno. Traduce AWS KMS le richieste generiche in un formato comprensibile al gestore delle chiavi esterno. Se il proxy dell'archivio chiavi esterno non è conforme alla [specificazione dell'API AWS KMS External Key Store Proxy](#), o se non funziona correttamente o non è in grado di comunicare con esso AWS KMS, non sarai in grado di creare o utilizzare le chiavi KMS nel tuo archivio di chiavi esterno.

Sebbene molti errori menzionino il proxy dell'archivio delle chiavi esterne a causa del suo ruolo fondamentale nell'architettura dell'archivio, tali problemi potrebbero avere origine nel gestore delle chiavi esterne o nella chiave esterna.

I problemi descritti in questa sezione riguardano errori relativi alla progettazione o al funzionamento del proxy dell'archivio delle chiavi esterne. La risoluzione di questi problemi potrebbe richiedere una modifica al software proxy. Rivolgiti al tuo amministratore proxy. Per aiutarti nell'eseguire la diagnostica dei problemi relativi al proxy, AWS KMS mette a disposizione un [client di test XKS Proxy](#), un client open source che verifica la conformità del proxy dell'archivio delle chiavi esterne alla [Specificazione API relativa al proxy dell'archivio delle chiavi esterne di AWS KMS](#).

```
CustomKeyStoreInvalidStateException    , KMSInvalidStateException    o  
XksProxyUriUnreachableException
```

Il proxy dell'archivio delle chiavi esterne è in uno stato non integro. Se visualizzi questo messaggio ripetutamente, informa l'amministratore del proxy dell'archivio delle chiavi esterne.

- Questo errore può indicare un problema operativo o un errore software nel proxy dell'archivio delle chiavi esterne. Puoi trovare le voci di CloudTrail registro relative all'operazione AWS KMS API che ha generato ogni errore. Questo errore può essere risolto riprovando a eseguire l'operazione. Se persiste, contatta l'amministratore del proxy dell'archivio delle chiavi esterne.
- Quando il proxy dell'archivio di chiavi esterno segnala (in una [GetHealthStatus](#) risposta) che tutte le istanze del gestore di chiavi esterno lo sono `UNAVAILABLE`, i tentativi di creare o aggiornare un archivio di chiavi esterno falliscono con questa eccezione. Se l'errore persiste, consulta la documentazione del gestore delle chiavi esterne.

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` o `XksProxyInvalidResponseException`

AWS KMS non è in grado di interpretare la risposta dal proxy dell'archivio chiavi esterno. Se visualizzi questo errore ripetutamente, rivolgiti all'amministratore del proxy dell'archivio delle chiavi esterne.

- AWS KMS le operazioni generano questa eccezione quando il proxy restituisce una risposta indefinita che AWS KMS non può essere analizzata o interpretata. Questo errore può verificarsi occasionalmente a causa di problemi esterni temporanei o errori di rete sporadici. Se l'errore persiste, potrebbe indicare che il proxy dell'archivio delle chiavi esterne non è conforme alla [Specifica API relativa al proxy dell'archivio delle chiavi esterne di AWS KMS](#). Informa l'amministratore o il fornitore dell'archivio delle chiavi esterne.

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` o `UnsupportedOperationException`

Il proxy dell'archivio delle chiavi esterne ha rifiutato la richiesta in quanto non supporta l'operazione di crittografia richiesta.

- Il proxy dell'archivio chiavi esterno deve supportare tutti i [proxy APIs](#) definiti nella specifica dell'[API AWS KMS External Key Store Proxy](#). Questo errore indica che il proxy non supporta l'operazione correlata alla richiesta. Informa l'amministratore o il fornitore dell'archivio delle chiavi esterne.

Problemi relativi all'autorizzazione proxy

Eccezioni: `CustomKeyStoreInvalidStateException`, `KMSInvalidStateException`

Alcuni proxy degli archivi delle chiavi esterne implementano i requisiti di autorizzazione per l'uso delle relative chiavi esterne. Un proxy dell'archivio delle chiavi esterne è consentito, ma non obbligatorio, per progettare e implementare uno schema di autorizzazione che consenta a determinati utenti di richiedere operazioni particolari in determinate condizioni. Ad esempio, un proxy potrebbe consentire a un utente di eseguire la crittografia con una particolare chiave esterna, ma non di effettuare l'operazione inversa. Per ulteriori informazioni, consulta [Autorizzazione proxy dell'archivio delle chiavi esterne \(facoltativo\)](#).

L'autorizzazione del proxy si basa sui metadati AWS KMS inclusi nelle sue richieste al proxy. I campi `awsSourceVpc` e `awsSourceVpce` sono inclusi nei metadati solo quando la richiesta proviene da un endpoint VPC e solo quando il chiamante si trova nello stesso account della chiave KMS.

```
"requestMetadata": {  
  "awsPrincipalArn": string,  
  "awsSourceVpc": string, // optional  
  "awsSourceVpce": string, // optional  
  "kmsKeyArn": string,  
  "kmsOperation": string,  
  "kmsRequestId": string,  
  "kmsViaService": string // optional  
}
```

Quando il proxy rifiuta una richiesta a causa di un errore di autorizzazione, l'AWS KMS operazione correlata fallisce. `CreateKey` restituisce `CustomKeyStoreInvalidStateException`. AWS KMS le operazioni crittografiche ritornano `KMSInvalidStateException`. Entrambi utilizzano il messaggio di errore seguente:

Il proxy dell'archivio delle chiavi esterne ha negato l'accesso all'operazione. Verifica che l'utente e la chiave esterna siano autorizzati per questa operazione e riprova a eseguire la richiesta.

- Per risolvere l'errore, utilizza il gestore delle chiavi esterne o gli strumenti del proxy per determinare il motivo per cui l'autorizzazione non è riuscita. Quindi, aggiorna la procedura che ha causato un errore nella richiesta di autorizzazione o utilizza gli strumenti del proxy dell'archivio delle chiavi esterne per aggiornare la policy di autorizzazione. Non puoi risolvere questo errore in AWS KMS.

Sicurezza di AWS Key Management Service

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 e te. Il [modello di responsabilità condivisa](#) descrive questo modello come sicurezza del cloud e sicurezza nel cloud:

- **Sicurezza del cloud:** AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi 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 a AWS Key Management Service (AWS KMS), consulta [AWS Servizi nell'ambito del programma di conformitàAWS](#) .
- **Sicurezza nel cloud:** la tua responsabilità è determinata dal AWS servizio che utilizzi. Inoltre AWS KMS, oltre alla configurazione e all'utilizzo di AWS KMS keys, l'utente è responsabile di altri fattori, tra cui la sensibilità dei dati, i requisiti aziendali e le leggi e i regolamenti applicabili

Questa documentazione aiuta a capire come applicare il modello di responsabilità condivisa durante l'utilizzo AWS Key Management Service. Ti mostra come configurare per AWS KMS soddisfare i tuoi obiettivi di sicurezza e conformità.

Argomenti

- [Protezione dei dati in AWS Key Management Service](#)
- [Gestione delle identità e degli accessi per AWS Key Management Service](#)
- [Registrazione e monitoraggio AWS Key Management Service](#)
- [Convalida della conformità per AWS Key Management Service](#)
- [Resilienza in AWS Key Management Service](#)
- [Sicurezza dell'infrastruttura in AWS Key Management Service](#)

Protezione dei dati in AWS Key Management Service

AWS Key Management Service archivia e protegge le chiavi di crittografia per renderle altamente disponibili, fornendo al contempo un controllo degli accessi solido e flessibile.

Argomenti

- [Protezione del materiale della chiave](#)
- [Crittografia dei dati](#)
- [Riservatezza del traffico Internet](#)

Protezione del materiale della chiave

Per impostazione predefinita, AWS KMS genera e protegge il materiale delle chiavi crittografiche per le chiavi KMS. Inoltre, AWS KMS offre opzioni per il materiale chiave creato e protetto all'esterno di AWS KMS.

Protezione del materiale chiave generato in AWS KMS

Quando si crea una chiave KMS, per impostazione predefinita, AWS KMS genera e protegge il materiale crittografico per la chiave KMS.

Per proteggere il materiale chiave per le chiavi KMS, AWS KMS si affida a una flotta distribuita di moduli di sicurezza hardware convalidati dallo standard [FIPS 140-3 di livello di sicurezza 3](#) (). HSMs Ogni AWS KMS HSM è un'appliance hardware dedicata e autonoma progettata per fornire funzioni crittografiche dedicate per soddisfare i requisiti di sicurezza e scalabilità di AWS KMS. (I HSMs file AWS KMS utilizzati nelle regioni della Cina sono certificati dall'[OSCCA](#) e sono conformi a tutte le normative cinesi pertinenti, ma non sono convalidati nell'ambito del programma di convalida dei moduli crittografici FIPS 140-3.)

Il materiale della chiave per una chiave KMS è crittografato per impostazione predefinita quando viene generato nel modulo HSM. Il materiale della chiave viene decrittografato solo nella memoria volatile del modulo HSM e solo per i pochi millisecondi necessari per utilizzarlo in un'operazione crittografica. Ogni volta che il materiale chiave non viene utilizzato attivamente, viene crittografato all'interno dell'HSM e trasferito su uno storage persistente a bassa latenza e [altamente durevole](#) (99,449%), dove rimane separato e isolato dal HSM. Il materiale della chiave in testo normale mantiene sempre i [limiti di sicurezza](#) HSM, non viene mai scritto su disco né memorizzato su alcun supporto di memorizzazione. (L'unica eccezione è la chiave pubblica di una coppia di chiavi asimmetriche, che non è segreta.)

AWS afferma come principio di sicurezza fondamentale che non vi è alcuna interazione umana con materiale chiave crittografico in chiaro di alcun tipo e in alcun modo. Servizio AWS Non esiste alcun meccanismo che consenta a nessuno, compresi Servizio AWS gli operatori, di visualizzare, accedere

o esportare materiale chiave in testo semplice. Questo principio si applica anche in caso di guasti catastrofici ed eventi di ripristino di emergenza. Il materiale contenente le chiavi del cliente in testo semplice AWS KMS viene utilizzato per operazioni crittografiche all'interno dello standard AWS KMS FIPS 140-3 e viene convalidato HSMs solo in risposta alle richieste autorizzate inviate al servizio dal cliente o da un suo delegato.

Per le [chiavi gestite dal cliente](#), chi crea Account AWS la chiave è l'unico proprietario non trasferibile della chiave. L'account proprietario ha il controllo completo ed esclusivo delle policy di autorizzazione che controllano l'accesso alla chiave. Infatti Chiavi gestite da AWS, Account AWS ha il controllo completo sulle politiche IAM che autorizzano le richieste a. Servizio AWS

Protezione del materiale della chiave generato esternamente a AWS KMS

AWS KMS fornisce alternative al materiale chiave generato in AWS KMS.

Gli [archivi di chiavi personalizzati](#), una AWS KMS funzionalità opzionale, consentono di creare chiavi KMS supportate da materiale chiave generato e utilizzato all'esterno di AWS KMS. Le chiavi KMS negli [archivi AWS CloudHSM delle chiavi](#) sono supportate dalle chiavi dei moduli di sicurezza AWS CloudHSM hardware controllati dall'utente. Queste HSMs sono certificate secondo il livello di sicurezza [FIPS 140-2 o il livello di sicurezza 140-3](#). Le chiavi KMS negli [archivi di chiavi esterni](#) sono supportate dalle chiavi di un gestore di chiavi esterno che puoi controllare e gestire all'esterno AWS, ad esempio un HSM fisico nel tuo data center privato.

Un'altra caratteristica opzionale consente di [importare il materiale della chiave](#) per una chiave KMS. Per proteggere il materiale chiave importato mentre è in transito verso AWS KMS, si crittografa il materiale chiave utilizzando una chiave pubblica da una coppia di chiavi RSA generata in un AWS KMS HSM. Il materiale chiave importato viene decrittografato in un AWS KMS HSM e ricrittografato con una chiave simmetrica nell'HSM. Come tutto il materiale chiave, il materiale AWS KMS chiave importato in testo semplice non esce mai da quello non crittografato. HSMs Tuttavia, il cliente che ha fornito il materiale della chiave è responsabile dell'uso sicuro, della durabilità e della manutenzione del materiale della chiave esternamente a AWS KMS.

Crittografia dei dati

I dati contenuti sono AWS KMS costituiti dal materiale chiave di AWS KMS keys crittografia che rappresentano. Questo materiale chiave è disponibile in testo semplice solo all'interno dei moduli di sicurezza AWS KMS hardware (HSMs) e solo quando è in uso. In caso contrario, il materiale della chiave viene crittografato e memorizzato in uno storage persistente durevole.

Il materiale chiave AWS KMS generato per le chiavi KMS non esce mai dal limite del non crittografato. AWS KMS HSMs Non viene esportato o trasmesso in nessuna operazione API. AWS KMS L'eccezione è rappresentata dalle [chiavi multiregionali](#), in cui si AWS KMS utilizza un meccanismo di replica interregionale per copiare il materiale chiave di una chiave multiregionale da un HSM in un HSM in un altro Regione AWS . Regione AWS Per i dettagli, consulta [Processo di replica](#) per chiavi multiregionali in Dettagli crittografici. AWS Key Management Service

Argomenti

- [Crittografia a riposo](#)
- [Crittografia in transito](#)

Crittografia a riposo

AWS KMS genera materiale chiave per i moduli di sicurezza hardware conformi AWS KMS keys a [FIPS 140-3 di livello di sicurezza 3](#) (). HSMs L'unica eccezione è rappresentata dalle regioni della Cina, dove le chiavi HSMs AWS KMS utilizzate per generare le chiavi KMS sono conformi a tutte le normative cinesi pertinenti, ma non sono convalidate nell'ambito del programma di convalida dei moduli crittografici FIPS 140-3. Quando non in uso, il materiale della chiave viene crittografato da una chiave HSM e scritto in uno storage persistente e durevole. Il materiale chiave per le chiavi KMS e le chiavi di crittografia che proteggono il materiale chiave non viene mai rilasciato in formato non crittografato. HSMs

La cifratura e la gestione del materiale della chiave per le chiavi KMS sono eseguite interamente da AWS KMS.

Per maggiori dettagli, consulta [Working with AWS KMS keys](#) in AWS Key Management Service Cryptographic Details.

Crittografia in transito

Il materiale chiave AWS KMS generato per le chiavi KMS non viene mai esportato o trasmesso nelle AWS KMS operazioni API. AWS KMS utilizza [identificatori di chiave](#) per rappresentare le chiavi KMS nelle operazioni API. Allo stesso modo, il materiale chiave per le chiavi KMS negli [archivi di chiavi AWS KMS personalizzati](#) non è esportabile e non viene mai trasmesso nelle nostre operazioni API. AWS KMS AWS CloudHSM

[Tuttavia, alcune operazioni AWS KMS API restituiscono chiavi dati.](#) Inoltre, i clienti possono utilizzare le operazioni API per [importare il materiale chiave](#) per chiavi KMS selezionate.

Tutte le chiamate AWS KMS API devono essere firmate e trasmesse utilizzando Transport Layer Security (TLS). AWS KMS richiede TLS 1.2 e consiglia TLS 1.3 in tutte le regioni. AWS KMS supporta anche il TLS ibrido post-quantistico per gli endpoint di AWS KMS servizio in tutte le regioni, ad eccezione delle regioni della Cina. AWS KMS non supporta il TLS ibrido post-quantistico per gli endpoint FIPS in. AWS GovCloud (US) Le chiamate a AWS KMS richiedono anche una moderna suite di cifratura che supporti la perfect forward secrecy, il che significa che il compromesso di qualsiasi segreto, come una chiave privata, non compromette anche la chiave della sessione.

Se sono necessari moduli crittografici convalidati FIPS 140-3 per l'accesso AWS tramite un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per utilizzare AWS KMS endpoint standard o endpoint AWS KMS FIPS, i client devono supportare TLS 1.2 o versioni successive. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-3](#). [Per un elenco degli endpoint AWS KMS FIPS, AWS Key Management Service consulta endpoint e quote in.](#) Riferimenti generali di AWS

Le comunicazioni tra gli host AWS KMS del servizio HSMs sono protette utilizzando Elliptic Curve Cryptography (ECC) e Advanced Encryption Standard (AES) in uno schema di crittografia autenticato. Per ulteriori dettagli, vedere [Sicurezza delle comunicazioni interne](#) in Cryptographic Details. AWS Key Management Service

Riservatezza del traffico Internet

AWS KMS supporta una Console di gestione AWS serie di operazioni API che consentono di crearle, gestirle AWS KMS keys e utilizzarle nelle operazioni crittografiche.

AWS KMS supporta due opzioni di connettività di rete dalla rete privata a AWS.

- Una connessione IPsec VPN su Internet
- [AWS Direct Connect](#), che collega la rete interna a una Direct Connect posizione tramite un cavo Ethernet standard in fibra ottica.

Tutte le chiamate AWS KMS API devono essere firmate e trasmesse utilizzando Transport Layer Security (TLS). Le chiamate richiedono anche una moderna suite di cifratura che supporta la [perfect forward secrecy](#). Il traffico verso i moduli di sicurezza hardware (HSMs) che memorizzano il materiale chiave per le chiavi KMS è consentito solo da host AWS KMS API noti sulla rete AWS interna.

Per connetterti direttamente AWS KMS dal tuo cloud privato virtuale (VPC) senza inviare traffico su Internet pubblico, utilizza gli endpoint VPC, con tecnologia. [AWS PrivateLink](#) Per ulteriori informazioni, consulta [Connect a AWS KMS tramite un endpoint VPC](#).

AWS KMS supporta anche un'opzione [ibrida di scambio di chiavi post-quantistiche](#) per il protocollo di crittografia di rete Transport Layer Security (TLS). È possibile utilizzare questa opzione con TLS quando ci si connette agli endpoint API. AWS KMS

Gestione delle identità e degli accessi per AWS Key Management Service

AWS Identity and Access Management (IAM) ti aiuta a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse. AWS KMS Per ulteriori informazioni, consulta [Utilizzo delle politiche IAM con AWS KMS](#).

Le [politiche chiave](#) sono il meccanismo principale per controllare l'accesso alle chiavi KMS. AWS KMS Ogni chiave KMS deve avere una policy chiave. Puoi inoltre utilizzare le [policy IAM](#) e le [concessioni](#), insieme alle policy delle chiavi, per controllare l'accesso alle chiavi KMS. Per ulteriori informazioni, consulta [Accesso e autorizzazioni alle chiavi KMS](#).

Se utilizzi un Amazon Virtual Private Cloud (Amazon VPC), puoi [creare un endpoint VPC di interfaccia su cui basare la tecnologia](#). AWS KMS [AWS PrivateLink](#) Puoi anche utilizzare le policy degli endpoint VPC per determinare quali responsabili possono accedere al tuo AWS KMS endpoint, a quali chiamate API possono effettuare e a quale chiave KMS possono accedere.

Argomenti

- [AWS politiche gestite per AWS Key Management Service](#)
- [Utilizzo di ruoli collegati ai servizi per AWS KMS](#)

AWS politiche gestite per AWS Key Management Service

Una politica AWS gestita è una politica autonoma creata e amministrata da AWS. AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo [policy gestite dal cliente](#) specifiche per i propri casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consulta [Policy gestite da AWS](#) nella Guida per l'utente IAM.

AWS politica gestita: AWSKey ManagementServicePowerUser

È possibile allegare la policy AWSKeyManagementServicePowerUser alle identità IAM.

È possibile utilizzare una policy gestita da AWSKeyManagementServicePowerUser per assegnare ai principali IAM dell'account le autorizzazioni di un utente esperto. Gli utenti esperti possono creare chiavi KMS, utilizzare e gestire le chiavi KMS da loro create e visualizzare tutte le chiavi KMS e le identità IAM. I principali che dispongono della policy gestita AWSKeyManagementServicePowerUser possono ottenere le autorizzazioni anche da altre origini, incluse le policy delle chiavi, altre policy IAM e concessioni.

AWSKeyManagementServicePowerUser è una policy IAM AWS gestita. Per ulteriori informazioni sulle policy AWS gestite, consulta le [policy AWS gestite](#) nella IAM User Guide.

Note

Le autorizzazioni in questa policy specifiche di una chiave KMS, ad esempio `kms:TagResource` e `kms:GetKeyRotationStatus`, sono efficaci solo quando la policy delle chiavi per quella chiave KMS [consente esplicitamente all' Account AWS di utilizzare policy IAM](#) per controllare l'accesso alla chiave. Per determinare se un'autorizzazione è specifica di una chiave KMS, consultare [AWS KMS autorizzazioni](#) e cercare un valore di chiave KMS nella colonna Resources (Risorse).

Questa policy fornisce all'utente esperto le autorizzazioni per qualsiasi chiave KMS con una policy delle chiavi che consenta l'operazione. Per autorizzazioni multi-account, come `kms:DescribeKey` e `kms:ListGrants`, ciò potrebbe includere chiavi KMS in Account AWS non attendibili. Per informazioni dettagliate, consulta [Best practice per le policy IAM](#) e [Autorizzazione per gli utenti in altri account di utilizzare una chiave KMS](#). Per determinare se un'autorizzazione è valida per le chiavi KMS in altri account, consultare [AWS KMS autorizzazioni](#) e cercare un valore Yes (Sì) nella colonna Cross-account use (Utilizzo per più account).

Per consentire ai responsabili di visualizzare la AWS KMS console senza errori, il principale necessita del [tag: GetResources](#) permission, che non è incluso nella `AWSKeyManagementServicePowerUser` policy. È possibile concedere questa autorizzazione in una policy IAM separata.

La policy IAM gestita da [AWSKeyManagementServicePowerUser](#) deve includere le seguenti autorizzazioni:

- Consente ai principali di creare chiavi KMS. Poiché questo processo include l'impostazione della policy delle chiavi, gli utenti esperti possono concedere a se stessi e ad altri l'autorizzazione per utilizzare e gestire le chiavi KMS create.
- Consente ai principali di creare ed eliminare [alias](#) e [tag](#) in tutte le chiavi KMS. La modifica di un tag o alias può consentire o negare l'autorizzazione all'utilizzo e alla gestione della chiave KMS. Per informazioni dettagliate, vedi [ABAC per AWS KMS](#).
- Consente ai principali di ottenere informazioni dettagliate su tutte le chiavi KMS, compreso l'ARN della chiave, la configurazione di crittografia, la policy delle chiavi, gli alias, i tag e lo [stato di rotazione](#).
- Consente ai principali di elencare utenti, gruppi e ruoli IAM.
- Questa policy non consente ai principali di utilizzare o gestire le chiavi KMS che non hanno creato. Tuttavia, possono modificare alias e tag su tutte le chiavi KMS, il che potrebbe consentire o negare loro l'autorizzazione all'utilizzo o alla gestione di una chiave KMS.

Per visualizzare le autorizzazioni per questa politica, consulta [AWSKeyManagementServicePowerUser](#) il AWS Managed Policy Reference.

AWS politica gestita: `AWSService RoleForKeyManagementServiceCustomKeyStores`

Non è possibile collegare `AWSServiceRoleForKeyManagementServiceCustomKeyStores` alle entità IAM. Questa policy è associata a un ruolo collegato al servizio che AWS KMS autorizza a visualizzare AWS CloudHSM i cluster associati all'archivio delle AWS CloudHSM chiavi e a creare la rete per supportare una connessione tra l'archivio chiavi personalizzato e il relativo cluster. AWS CloudHSM Per ulteriori informazioni, consulta [Autorizzazione AWS KMS alla gestione AWS CloudHSM e alle risorse Amazon EC2](#).

AWS politica gestita: AWSService RoleForKeyManagementServiceMultiRegionKeys

Non è possibile collegare `AWSServiceRoleForKeyManagementServiceMultiRegionKeys` alle entità IAM. Questa policy è associata a un ruolo collegato al servizio che AWS KMS autorizza a sincronizzare qualsiasi modifica al materiale chiave di una chiave primaria multiregionale con le relative chiavi di replica. Per ulteriori informazioni, consulta [Autorizzazione AWS KMS alla sincronizzazione di chiavi multiregionali](#).

AWS KMS aggiornamenti alle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite AWS KMS da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per gli avvisi automatici sulle modifiche apportate alla pagina, iscriviti al feed RSS alla pagina AWS KMS [Cronologia dei documenti](#).

Modifica	Descrizione	Data
AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy : aggiornamento a policy esistente	AWS KMS ha aggiunto un campo <code>statement ID (Sid)</code> alla policy gestita nella versione di policy v2.	21 novembre 2024
AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy : aggiornamento a policy esistente	AWS KMS ha aggiunto <code>ec2:DescribeVpcs</code> , <code>ec2:DescribeNetworkAcls</code> , e <code>ec2:DescribeNetworkInterfaces</code> le autorizzazioni per monitorare le modifiche nel VPC che contiene AWS CloudHSM il cluster in modo da AWS KMS fornire messaggi di errore chiari in caso di errori.	10 novembre 2023
AWS KMS ha iniziato a tenere traccia delle modifiche	AWS KMS ha iniziato a tenere traccia delle modifiche per le sue politiche AWS gestite.	10 novembre 2023

Utilizzo di ruoli collegati ai servizi per AWS KMS

AWS Key Management Service utilizza ruoli collegati ai [servizi AWS Identity and Access Management](#) (IAM). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM a cui è collegato direttamente. AWS KMS I ruoli collegati ai servizi sono definiti AWS KMS e includono tutte le autorizzazioni richieste dal servizio per chiamare altri AWS servizi per tuo conto.

Un ruolo collegato al servizio semplifica la configurazione AWS KMS perché non è necessario aggiungere manualmente le autorizzazioni necessarie. AWS KMS definisce le autorizzazioni dei ruoli collegati ai servizi e, se non diversamente definito, solo può assumerne i ruoli. AWS KMS Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere allegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato ai servizi solo dopo aver eliminato le risorse correlate. In questo modo proteggi AWS KMS le tue risorse perché non puoi rimuovere inavvertitamente l'autorizzazione ad accedere alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta la sezione [Servizi AWS che funzionano con IAM](#) e cerca i servizi che riportano Sì nella colonna Ruolo associato ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Per visualizzare i dettagli sugli aggiornamenti ai ruoli collegati ai servizi discussi in questo argomento, vedere. [AWS KMS aggiornamenti alle politiche AWS gestite](#)

Argomenti

- [Autorizzazione AWS KMS alla gestione AWS CloudHSM e alle risorse Amazon EC2](#)
- [Autorizzazione AWS KMS alla sincronizzazione di chiavi multiregionali](#)

Autorizzazione AWS KMS alla gestione AWS CloudHSM e alle risorse Amazon EC2

Per supportare i tuoi AWS CloudHSM key store, hai AWS KMS bisogno dell'autorizzazione per ottenere informazioni sui tuoi AWS CloudHSM cluster. È inoltre necessaria l'autorizzazione per creare l'infrastruttura di rete che collega l'archivio delle AWS CloudHSM chiavi al relativo AWS CloudHSM cluster. Per ottenere queste autorizzazioni, AWS KMS crea il ruolo `AWSServiceRoleForKeyManagementServiceCustomKeyStores` collegato al servizio nel tuo Account AWS. Gli utenti che creano archivi di AWS CloudHSM chiavi devono disporre

dell'iam:CreateServiceLinkedRoleautorizzazione che consenta loro di creare ruoli collegati ai servizi.

Per visualizzare i dettagli sugli aggiornamenti della politica

AWSKeyManagementServiceCustomKeyStoresServiceRolePolicygestita, consulta. [AWS KMS aggiornamenti alle politiche AWS gestite](#)

Argomenti

- [Informazioni sul ruolo AWS KMS collegato al servizio](#)
- [Creazione del ruolo collegato ai servizi](#)
- [Modifica della descrizione di un ruolo collegato ai servizi](#)
- [Eliminazione del ruolo collegato ai servizi](#)

Informazioni sul ruolo AWS KMS collegato al servizio

Un [ruolo collegato ai servizi è un ruolo](#) IAM che concede a un AWS servizio l'autorizzazione a chiamare altri AWS servizi per tuo conto. È progettato per semplificare l'utilizzo delle funzionalità di più AWS servizi integrati senza dover creare e mantenere politiche IAM complesse. Per ulteriori informazioni, consulta [Utilizzo di ruoli collegati ai servizi per AWS KMS](#).

Per gli archivi AWS CloudHSM chiave, AWS KMS crea il ruolo

AWSServiceRoleForKeyManagementServiceCustomKeyStorescollegato ai servizi con la policy AWSKeyManagementServiceCustomKeyStoresServiceRolePolicygestita. Questa policy concede al ruolo le seguenti autorizzazioni:

- [CloudHSM:Describe*](#) — rileva le modifiche nel AWS CloudHSM cluster collegato al tuo archivio di chiavi personalizzato.
- [ec2: CreateSecurityGroup](#) — utilizzato quando [connetti un archivio di AWS CloudHSM chiavi](#) per creare il gruppo di sicurezza che abilita il flusso del traffico di rete tra e il cluster. AWS KMS AWS CloudHSM
- [ec2: AuthorizeSecurityGroupIngress](#) — utilizzato quando si [connette un AWS CloudHSM key store](#) per consentire l'accesso alla rete dal AWS KMS VPC che contiene AWS CloudHSM il cluster.
- [ec2: CreateNetworkInterface](#) — utilizzato quando si [connette un AWS CloudHSM key store](#) per creare l'interfaccia di rete utilizzata per la comunicazione tra AWS KMS e il cluster. AWS CloudHSM
- [ec2: RevokeSecurityGroupEgress](#) — utilizzato quando si [connette un archivio di AWS CloudHSM chiavi](#) per rimuovere tutte le regole in uscita dal gruppo di sicurezza creato. AWS KMS

- [ec2: DeleteSecurityGroup](#) — utilizzato quando si [disconnette un archivio di AWS CloudHSM chiavi](#) per eliminare i gruppi di sicurezza creati quando si è connesso l'archivio chiavi. AWS CloudHSM
- [ec2: DescribeSecurityGroups](#) — utilizzato per monitorare le modifiche nel gruppo di sicurezza AWS KMS creato nel VPC che contiene AWS CloudHSM il cluster in modo AWS KMS che possa fornire messaggi di errore chiari in caso di guasti.
- [ec2: DescribeVpcs](#) — utilizzato per monitorare le modifiche nel VPC che contiene AWS CloudHSM il cluster in modo da AWS KMS fornire messaggi di errore chiari in caso di guasti.
- [ec2: DescribeNetworkAcls](#) — utilizzato per monitorare i cambiamenti nella rete ACLs per il VPC che contiene AWS CloudHSM il cluster in modo AWS KMS che possa fornire messaggi di errore chiari in caso di guasti.
- [ec2: DescribeNetworkInterfaces](#) — utilizzato per monitorare le modifiche nelle interfacce di rete AWS KMS create nel VPC che contiene il AWS CloudHSM cluster in modo da AWS KMS fornire messaggi di errore chiari in caso di guasti.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudhsm:Describe*",
        "ec2:CreateNetworkInterface",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateSecurityGroup",
        "ec2:DescribeSecurityGroups",
        "ec2:RevokeSecurityGroupEgress",
        "ec2>DeleteSecurityGroup",
        "ec2:DescribeVpcs",
        "ec2:DescribeNetworkAcls",
        "ec2:DescribeNetworkInterfaces"
      ],
      "Resource": "*"
    }
  ]
}
```


Poiché solo il ruolo `AWSServiceRoleForKeyManagementServiceCustomKeyStores` collegato al servizio è affidabile `cks.kms.amazonaws.com`, solo può assumere questo ruolo collegato al servizio. AWS KMS Questo ruolo è limitato alle operazioni che AWS KMS richiedono la visualizzazione dei AWS CloudHSM cluster e la connessione di un archivio di AWS CloudHSM chiavi al cluster associato. AWS CloudHSM Non fornisce AWS KMS autorizzazioni aggiuntive. Ad esempio, AWS KMS non dispone dell'autorizzazione per creare, gestire o eliminare AWS CloudHSM i cluster o HSMs i backup.

Regioni

Come la funzionalità di archiviazione delle AWS CloudHSM chiavi, il `AWSServiceRoleForKeyManagementServiceCustomKeyStores` ruolo è supportato Regioni AWS ovunque AWS KMS ed AWS CloudHSM è disponibile. Per un elenco delle funzionalità Regioni AWS supportate da ciascun servizio, consulta [AWS Key Management Service Endpoints and Quotas](#) e [AWS CloudHSM endpoints and quotas](#) in. Riferimenti generali di Amazon Web Services

Per ulteriori informazioni su come AWS i servizi utilizzano i ruoli collegati ai servizi, consulta Using service-linked roles nella IAM [User Guide](#).

Creazione del ruolo collegato ai servizi

AWS KMS crea automaticamente il ruolo `AWSServiceRoleForKeyManagementServiceCustomKeyStores` collegato al servizio Account AWS quando crei un archivio di AWS CloudHSM chiavi, se il ruolo non esiste già. Non è possibile creare o ricreare direttamente questo ruolo collegato ai servizi.

Modifica della descrizione di un ruolo collegato ai servizi

Non puoi modificare il nome del ruolo o le istruzioni di policy nel ruolo collegato ai servizi `AWSServiceRoleForKeyManagementServiceCustomKeyStores`, ma puoi modificare la descrizione del ruolo. Per istruzioni, consulta [Modifica di un ruolo collegato ai servizi](#) nella Guida per l'utente di IAM.

Eliminazione del ruolo collegato ai servizi

AWS KMS non elimina il ruolo `AWSServiceRoleForKeyManagementServiceCustomKeyStores` collegato al servizio dal tuo Account AWS anche se hai [eliminato tutti i tuoi archivi di chiavi](#). AWS CloudHSM Sebbene al momento non esista una procedura per eliminare il ruolo

AWSServiceRoleForKeyManagementServiceCustomKeyStores collegato al servizio, non assume questo ruolo né ne utilizza le autorizzazioni a meno che AWS KMS non disponga di archivi di chiavi attivi. AWS CloudHSM

Autorizzazione AWS KMS alla sincronizzazione di chiavi multiregionali

Per supportare [le chiavi multiregionali](#), è AWS KMS necessaria l'autorizzazione per sincronizzare le [proprietà condivise](#) di una chiave primaria multiregionale con le relative chiavi di replica. Per ottenere queste autorizzazioni, AWS KMS crea il ruolo collegato al servizio in AWSServiceRoleForKeyManagementServiceMultiRegionKeys. Account AWS Gli utenti che creano chiavi multiregionali devono disporre dell'`iam:CreateServiceLinkedRole` autorizzazione che consenta loro di creare ruoli collegati ai servizi.

È possibile visualizzare l'[SynchronizeMultiRegionKey](#) CloudTrail evento che registra la AWS KMS sincronizzazione delle proprietà condivise nei registri. AWS CloudTrail

Per visualizzare i dettagli sugli aggiornamenti della politica AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy gestita, vedere. [AWS KMS aggiornamenti alle politiche AWS gestite](#)

Argomenti

- [Informazioni sul ruolo collegato ai servizi per le chiavi multi-regione](#)
- [Creazione del ruolo collegato ai servizi](#)
- [Modifica della descrizione di un ruolo collegato ai servizi](#)
- [Eliminazione del ruolo collegato ai servizi](#)

Informazioni sul ruolo collegato ai servizi per le chiavi multi-regione

Un [ruolo collegato ai servizi è un ruolo](#) IAM che concede a un AWS servizio l'autorizzazione a chiamare altri AWS servizi per tuo conto. È progettato per semplificare l'utilizzo delle funzionalità di più AWS servizi integrati senza dover creare e mantenere politiche IAM complesse.

Per le chiavi multiregionali, AWS KMS crea il ruolo AWSServiceRoleForKeyManagementServiceMultiRegionKeys collegato ai servizi con la AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy policy gestita. Questa policy dà al ruolo l'autorizzazione `kms:SynchronizeMultiRegionKey`, che consente di sincronizzare le proprietà condivise delle chiavi multi-regione.

Poiché solo il ruolo `AWSServiceRoleForKeyManagementServiceMultiRegionKeys` collegato al servizio è affidabile `arn:kms:amazonaws.com`, solo AWS KMS può assumere questo ruolo collegato al servizio. Questo ruolo è limitato alle operazioni che AWS KMS richiedono la sincronizzazione delle proprietà condivise in più regioni. Non fornisce autorizzazioni AWS KMS aggiuntive. Ad esempio, AWS KMS non dispone dell'autorizzazione per creare, replicare o eliminare alcuna chiave KMS.

Per ulteriori informazioni su come AWS i servizi utilizzano i ruoli collegati ai servizi, consulta [Using Service-Linked Roles](#) nella [IAM User Guide](#).

JSON

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Sid" : "KMSSynchronizeMultiRegionKey",
      "Effect" : "Allow",
      "Action" : [
        "kms:SynchronizeMultiRegionKey"
      ],
      "Resource" : "*"
    }
  ]
}
```

Creazione del ruolo collegato ai servizi

AWS KMS crea automaticamente il ruolo

`AWSServiceRoleForKeyManagementServiceMultiRegionKeys` collegato ai servizi in tuo Account AWS quando crei una chiave multiregionale, se il ruolo non esiste già. Non è possibile creare o ricreare direttamente questo ruolo collegato ai servizi.

Modifica della descrizione di un ruolo collegato ai servizi

Non è possibile modificare il nome del ruolo o le dichiarazioni politiche nel ruolo

`AWSServiceRoleForKeyManagementServiceMultiRegionKeys` collegato al servizio, ma è possibile modificare la descrizione del ruolo. Per istruzioni, consulta [Modifica di un ruolo collegato ai servizi](#) nella Guida per l'utente di IAM.

Eliminazione del ruolo collegato ai servizi

AWS KMS non elimina il ruolo

`AWSServiceRoleForKeyManagementServiceMultiRegionKeys` collegato al servizio

dal tuo Account AWS e non puoi eliminarlo. Tuttavia, AWS KMS non assume il

`AWSServiceRoleForKeyManagementServiceMultiRegionKeys` ruolo né utilizza alcuna delle sue

autorizzazioni a meno che non si disponga di chiavi multiregionali nella propria regione. Account

AWS

Registrazione e monitoraggio AWS Key Management Service

Il monitoraggio è una parte importante della comprensione della disponibilità, dello stato e dell'utilizzo delle AWS KMS keys in AWS KMS. Il monitoraggio aiuta a mantenere la sicurezza, l'affidabilità, la disponibilità e le prestazioni delle AWS soluzioni. AWS fornisce diversi strumenti per il monitoraggio delle chiavi KMS.

AWS CloudTrail Registri

Ogni chiamata a un'operazione AWS KMS API viene acquisita come evento in un AWS CloudTrail registro. Questi registri registrano tutte le chiamate API dalla AWS KMS console e le chiamate effettuate da AWS KMS e altri AWS servizi. Le chiamate API tra account, ad esempio una chiamata a utilizzare una chiave KMS in un altro account Account AWS, vengono registrate nei CloudTrail registri di entrambi gli account.

Durante la risoluzione dei problemi o il controllo puoi utilizzare il log per ricostruire il ciclo di vita di una chiave KMS. Puoi inoltre visualizzare la gestione e l'utilizzo della chiave KMS nelle operazioni di crittografia. Per ulteriori informazioni, consulta [the section called “Registrazione con AWS CloudTrail”](#).

CloudWatch Registri Amazon

Monitora, archivia e accedi ai tuoi file di registro da AWS CloudTrail e altre fonti. Per ulteriori informazioni, consulta la [Amazon CloudWatch User Guide](#).

Infatti AWS KMS, CloudWatch memorizza informazioni utili che ti aiutano a prevenire problemi con le tue chiavi KMS e le risorse che proteggono. Per ulteriori informazioni, consulta [the section called “Monitora le chiavi con CloudWatch”](#).

Amazon EventBridge

AWS KMS genera EventBridge eventi quando la chiave KMS viene [ruotata o eliminata](#) o il [materiale chiave importato nella chiave](#) KMS scade. Cerca AWS KMS eventi (operazioni API) e indirizzali a una o più funzioni o flussi di destinazione per acquisire informazioni sullo stato. Per ulteriori informazioni, consulta [the section called “Monitora le chiavi con Amazon EventBridge”](#) la [Amazon EventBridge User Guide](#).

CloudWatch Metriche Amazon

Puoi monitorare le tue chiavi KMS utilizzando i CloudWatch parametri, che raccolgono ed elaborano dati grezzi per trasformarli in metriche prestazionali. AWS KMS I dati vengono registrati a intervalli di due settimane in modo da poter visualizzare le tendenze delle informazioni correnti e cronologiche. Questo ti aiuta a capire come vengono usate le tue chiavi KMS e come il loro utilizzo cambia nel tempo. Per informazioni sull'utilizzo delle CloudWatch metriche per monitorare le chiavi KMS, consulta [AWS KMS metriche e dimensioni](#)

CloudWatch Allarmi Amazon

Osserva una singola modifica del parametro in un periodo di tempo specificato. Quindi esegui una o più operazioni basate sul valore del parametro relativo a una soglia per un certo numero di periodi. Ad esempio, puoi creare un CloudWatch allarme che viene attivato quando qualcuno tenta di utilizzare una chiave KMS la cui eliminazione è pianificata in un'operazione crittografica. Ciò indica che la chiave KMS è ancora in uso e probabilmente non dovrebbe essere eliminata. Per ulteriori informazioni, consulta [the section called “Creazione di un allarme”](#).

AWS Security Hub CSPM

È possibile monitorare AWS KMS l'utilizzo per verificare la conformità agli standard del settore della sicurezza e alle migliori pratiche utilizzando. AWS Security Hub CSPM Security Hub CSPM utilizza i controlli di sicurezza per valutare le configurazioni delle risorse e gli standard di sicurezza per aiutarti a rispettare vari framework di conformità. Per ulteriori informazioni, consulta [Controlli di AWS Key Management Service](#) nella Guida per l'utente di AWS Security Hub CSPM .

Convalida della conformità per AWS Key Management Service

I revisori esterni valutano la sicurezza e la conformità nell' AWS Key Management Service ambito di più programmi di AWS conformità. Questi includono SOC, PCI, FedRAMP, HIPAA e altri.

Argomenti

- [Documenti di conformità e sicurezza](#)
- [Ulteriori informazioni](#)

Documenti di conformità e sicurezza

I seguenti documenti di conformità e sicurezza riguardano AWS KMS. Per visualizzarli, usa [AWS Artifact](#).

- Cloud Computing Compliance Controls Catalogue (C5)
- ISO 27001:2013 Statement of Applicability (SoA)
- Certificazione ISO 27001:2013
- ISO 27017:2015 Statement of Applicability (SoA)
- Certificazione ISO 27017:2015
- ISO 27018:2015 Statement of Applicability (SoA)
- Certificazione ISO 27018:2014
- Certificazione ISO 9001:2015
- Attestazione di conformità allo standard DSS PCI e riepilogo delle responsabilità
- Service Organization Controls (SOC) 1 Report
- Service Organization Controls (SOC) 2 Report
- Service Organization Controls (SOC) 2 Report For Confidentiality
- FedRAMP-High

Per assistenza sull'utilizzo AWS Artifact, consulta [Scaricamento dei report in AWS Artifact](#).

Ulteriori informazioni

La vostra responsabilità di conformità durante l'utilizzo AWS KMS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. Se l'utilizzo di AWS KMS è soggetto alla conformità a uno standard pubblicato, AWS fornisce risorse per aiutarti a:

- [AWS Servizi che rientrano nell'ambito del programma di conformità](#): questa pagina elenca AWS i servizi che rientrano nell'ambito di programmi di conformità specifici. Per informazioni generali, consulta [Programmi di conformità di AWS](#).

- [Guide introduttive su sicurezza e conformità](#): queste guide all'implementazione illustrano considerazioni sull'architettura e forniscono passaggi per implementare ambienti di base incentrati sulla sicurezza e sulla conformità. AWS
- [AWS Risorse per la conformità](#): questa raccolta di cartelle di lavoro e guide potrebbe riguardare il settore e la località in cui operi.
- [AWS Config](#)— Questo AWS servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- [AWS Security Hub CSPM](#)— Questo AWS servizio offre una visione completa dello stato di sicurezza dell'utente all'interno. AWS Security Hub CSPM utilizza i controlli di sicurezza per valutare le AWS risorse e verificare la conformità rispetto agli standard e alle migliori pratiche del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta il riferimento ai controlli [CSPM di Security Hub](#).

Resilienza in AWS Key Management Service

L'infrastruttura AWS globale è costruita attorno a Regioni AWS zone di disponibilità. Regioni AWS forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, è possibile progettare e gestire applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture tradizionali a data center singolo o multiplo.

Oltre all'infrastruttura AWS globale, AWS KMS offre diverse funzionalità per supportare le esigenze di resilienza e backup dei dati. Per ulteriori informazioni sulle Regioni AWS zone di disponibilità, consulta [AWS Global Infrastructure](#).

Isolamento regionale

AWS Key Management Service (AWS KMS) è un servizio regionale autosufficiente disponibile per tutti. Regioni AWS La progettazione isolata a livello regionale AWS KMS garantisce che un problema di disponibilità in una regione Regione AWS non possa influire sul AWS KMS funzionamento in nessun'altra regione. AWS KMS è progettato per garantire zero tempi di inattività pianificati, con tutti gli aggiornamenti software e le operazioni di scalabilità eseguiti senza interruzioni e impercettibili.

Il AWS KMS [Service Level Agreement](#) (SLA) include un impegno di servizio del 99,999% per tutti i KMS. APIs Per portare a termine questo impegno, AWS KMS garantisce che tutti i dati e le

informazioni di autorizzazione necessarie per eseguire una richiesta API siano disponibili su tutti gli host regionali che ricevono la richiesta.

L'AWS KMS infrastruttura viene replicata in almeno tre zone di disponibilità () AZs in ogni regione. Per garantire che i guasti di più host non influiscano sulle AWS KMS prestazioni, AWS KMS è progettata per soddisfare il traffico dei clienti proveniente da qualsiasi area AZs geografica.

Le modifiche apportate alle proprietà o alle autorizzazioni di una chiave KMS vengono replicate in tutti gli host della regione per garantire che la richiesta successiva possa essere elaborata correttamente da qualunque host nella regione. Le richieste di [operazioni crittografiche](#) che utilizzano la chiave KMS vengono inoltrate a una flotta di moduli di sicurezza AWS KMS hardware (HSMs), ognuno dei quali può eseguire l'operazione con la chiave KMS.

Design multi-tenant

Il design multi-tenant di AWS KMS consente di soddisfare lo SLA di disponibilità del 99,999% e di sostenere tassi di richieste elevati, proteggendo al contempo la riservatezza delle chiavi e dei dati.

Sono implementati più meccanismi di applicazione dell'integrità per garantire che la chiave KMS specificata per l'operazione crittografica sia sempre quella utilizzata.

Il materiale della chiave in testo normale per le chiavi KMS è ampiamente protetto. Il materiale della chiave viene crittografato nel modulo HSM non appena viene creato e quest'ultimo viene spostato immediatamente in uno storage sicuro a bassa latenza. La chiave crittografata viene recuperata e decrittografata all'interno del modulo HSM solo nel momento in cui viene utilizzata. La chiave in testo normale rimane nella memoria HSM solo per il tempo necessario al completamento dell'operazione di crittografia. Dopo di che, viene nuovamente crittografata nel modulo HSM e la chiave crittografata viene restituita allo storage. Il materiale chiave in testo semplice non esce mai dall'archivio HSMs; non viene mai scritto su un dispositivo di archiviazione persistente.

Le migliori pratiche di resilienza in AWS KMS

Per ottimizzare la resilienza AWS KMS delle risorse, prendi in considerazione le seguenti strategie.

- Per il supporto della strategia di backup e ripristino di emergenza, valuta le chiavi multi-regione, costituite da chiavi KMS create in un'unica Regione AWS e replicate solo nelle regioni che hai specificato. Con le chiavi multiregionali, è possibile spostare risorse crittografate tra Regioni AWS (all'interno della stessa partizione) senza mai esporre il testo in chiaro e decrittografare la risorsa, quando necessario, in una delle regioni di destinazione. Le chiavi multi-regione correlate sono interoperabili perché condividono lo stesso materiale e lo stesso ID, ma hanno policy indipendenti

per il controllo degli accessi ad alta risoluzione. Per informazioni dettagliate, consulta [Chiavi multi-regione in AWS KMS](#).

- [Per proteggere le tue chiavi in un servizio multi-tenant come AWS KMS, assicurati di utilizzare i controlli di accesso, comprese le policy chiave e le policy IAM](#). Inoltre, puoi inviare le tue richieste AWS KMS utilizzando un endpoint con interfaccia VPC alimentato da AWS PrivateLink. In tal caso, tutte le comunicazioni tra Amazon VPC e AWS KMS vengono condotte interamente all'interno della AWS rete utilizzando un AWS KMS endpoint dedicato limitato al tuo VPC. Puoi proteggere ulteriormente queste richieste creando un livello di autorizzazione aggiuntivo tramite le [policy degli endpoint VPC](#). Per ulteriori dettagli, consulta [Connessione a AWS KMS mediante un endpoint VPC](#).

Sicurezza dell'infrastruttura in AWS Key Management Service

In quanto servizio gestito, AWS Key Management Service (AWS KMS) è protetto dalle procedure di sicurezza di rete AWS globali descritte in [Amazon Web Services: Overview of Security Processes](#).

Per accedere AWS KMS tramite la rete, puoi chiamare le operazioni AWS KMS API descritte nell'[AWS Key Management Service API Reference](#). AWS KMS richiede TLS 1.2 e consiglia TLS 1.3 in tutte le regioni. AWS KMS supporta anche il TLS ibrido post-quantistico per gli endpoint di AWS KMS servizio in tutte le regioni, ad eccezione delle regioni della Cina. AWS KMS non supporta il TLS ibrido post-quantistico per gli endpoint FIPS in. AWS GovCloud (US) Per utilizzare [endpoint standard AWS KMS](#) o [endpoint FIPS AWS KMS](#), i client devono supportare TLS 1.2 o versioni successive. I client devono, inoltre, supportare le suite di crittografia con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un chiave di accesso ID e una chiave di accesso segreta associata a un account principale IAM. In alternativa è possibile utilizzare [AWS Security Token Service](#) (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Puoi chiamare queste operazioni API da qualsiasi posizione di rete, ma AWS KMS supporta condizioni di policy globali che consentono di controllare l'accesso a una chiave KMS in base all'indirizzo IP di origine, al VPC e all'endpoint VPC. È possibile utilizzare le chiavi di condizione globali nelle policy delle chiavi e nelle policy IAM. Tuttavia, queste condizioni possono AWS impedire l'utilizzo della chiave KMS per tuo conto. Per informazioni dettagliate, vedi [AWS chiavi di condizione globali](#).

Ad esempio, la seguente dichiarazione politica chiave consente agli utenti che possono assumere il `KMSTestRole` ruolo di utilizzarla AWS KMS key per le [operazioni crittografiche](#) specificate, a meno che l'indirizzo IP di origine non sia uno degli indirizzi IP specificati nella politica.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::111122223333:role/KMSTestRole" },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": [
            "192.0.2.0/24",
            "203.0.113.0/24"
          ]
        }
      }
    }
  ]
}
```

Isolamento di host fisici

La sicurezza dell'infrastruttura fisica AWS KMS utilizzata è soggetta ai controlli descritti nella sezione [Sicurezza fisica e ambientale di Amazon Web Services: panoramica dei processi di sicurezza](#). Puoi trovare altri dettagli nei report di conformità e nei risultati degli audit di terze parti elencati nella sezione precedente.

AWS KMS è supportato da moduli di sicurezza hardware rinforzati dedicati (HSMs) progettati con controlli specifici per resistere agli attacchi fisici. HSMs Sono dispositivi fisici che non dispongono di

un livello di virtualizzazione, come un hypervisor, che condivide il dispositivo fisico tra diversi tenant logici. Il materiale chiave per AWS KMS keys viene archiviato solo nella memoria volatile di e solo mentre la HSMs chiave KMS è in uso. Questa memoria viene cancellata quando il modulo HSM non è in stato operativo, inclusi arresti e ripristini previsti e non intenzionali. Per informazioni dettagliate sul funzionamento di AWS KMS HSMs, vedere Dettagli [AWS Key Management Service crittografici](#).

Quote

Per renderlo AWS KMS reattivo e performante per tutti gli utenti, AWS KMS applica due tipi di quote, quote di risorse e quote di richiesta. Ogni quota viene calcolata in modo indipendente per ogni Regione di ciascun Account AWS.

[Tutte le AWS KMS quote sono regolabili, ad eccezione della quota di risorse di rotazione su richiesta e della quota di richiesta del key store.AWS CloudHSM](#) Per richiedere un aumento delle quote, consultare [Richiesta di aumento delle quote](#) nella Guida dell'utente di Service Quotas. Per richiedere una riduzione della quota, modificare una quota non elencata in Service Quotas o modificare una quota in Regione AWS cui Service Quotas AWS KMS for non è disponibile, [Supporto AWS visita il](#) Centro e crea un caso.

Argomenti

- [Quote delle risorse](#)
- [Quote di richieste](#)
- [Richieste di limitazione AWS KMS](#)

Quote delle risorse

AWS KMS stabilisce quote di risorse per garantire che possa fornire un servizio rapido e resiliente a tutti i nostri clienti. Alcune quote di risorse si applicano solo alle risorse create dall'utente, ma non alle risorse create dai AWS servizi per l'utente. Le risorse che utilizzi, ma che non sono nel tuo account Account AWS, come [Chiavi di proprietà di AWS](#), non vengono considerate nel calcolo di queste quote.

Se viene superato un limite di risorse, le richieste per creare una risorsa aggiuntiva di quel tipo generano il messaggio di errore `LimitExceededException`.

Tutte le quote di AWS KMS risorse sono regolabili, ad eccezione della quota di risorse a [rotazione su richiesta](#). Per richiedere un aumento delle quote, consultare [Richiesta di aumento delle quote](#) nella Guida dell'utente di Service Quotas. Per richiedere una riduzione della quota, modificare una quota non elencata in Service Quotas o modificare una quota in Regione AWS cui Service Quotas AWS KMS for non è disponibile, [Supporto AWS visita il](#) Centro e crea un caso.

La tabella seguente elenca e descrive le quote di AWS KMS risorse in ciascuna Account AWS regione.

Nome quota	Valore predefinito	Si applica a	Regolabile
AWS KMS keys	100.000	Chiavi gestite dal cliente	Sì
Alias per chiave KMS	10	Alias creati dal cliente	Sì
Autorizzazioni per chiave KMS	50.000	Chiavi gestite dal cliente	Sì
Quote di risorse per l'archivio delle chiavi personalizzate	10	Account AWS e regione	Sì
Rotazione su richiesta	50	Chiavi gestite dal cliente	No

Oltre alle quote di risorse, AWS KMS utilizza le quote di richiesta per garantire la reattività del servizio. Per informazioni dettagliate, vedi [the section called “Quote di richieste”](#).

AWS KMS keys: 100.000

Puoi avere fino a 100.000 [chiavi gestite dal cliente](#) in ciascuna Regione dell' Account AWS. Questa quota si applica a tutte le chiavi gestite dal cliente in tutte le Regioni AWS indipendentemente dalla [specifica della chiave](#) o dallo [stato della chiave](#). Ogni chiave KMS è considerata una risorsa. [Chiavi gestite da AWS](#) e [Chiavi di proprietà di AWS](#) non rientrano nel calcolo di questa quota.

Alias per chiave KMS: 50

Puoi associare fino a 50 [alias](#) a ciascuna [chiave gestita dal cliente](#). Gli alias AWS associati a [Chiavi gestite da AWS](#) non vengono conteggiati ai fini di questa quota. Questa quota potrebbe essere interessata quando [crei](#) o [aggiorni](#) un alias.

Note

La ResourceAliases condizione [kms:](#) è efficace solo quando la chiave KMS è conforme a questa quota. Se una chiave KMS supera questa quota, alle entità principali autorizzate a

usare la chiave KMS tramite la condizione `kms:ResourceAliases` viene negato l'accesso alla chiave KMS. Per informazioni dettagliate, vedi [Accesso negato a causa di quota alias](#).

La quota di chiavi Alias per KMS sostituisce la quota Alias per regione che limitava il numero totale di alias in ciascuna regione di un Account AWS. AWS KMS ha eliminato la quota Alias per regione.

Concessioni per chiave KMS: 50.000

Ogni [chiave gestita dal cliente](#) può avere fino a 50.000 [concessioni](#), incluse le concessioni create dai [servizi AWS integrati con AWS KMS](#). Questa quota non si applica alle [Chiavi gestite da AWS](#) o alle [Chiavi di proprietà di AWS](#).

Un effetto di questa quota è che non è possibile eseguire più di 50.000 operazioni concesse che utilizzano la stessa chiave KMS contemporaneamente. Una volta raggiunta la quota, puoi creare nuove concessioni sulla chiave KMS solo quando una concessione attiva viene ritirata o revocata.

Ad esempio, quando colleghi un volume Amazon Elastic Block Store (Amazon EBS) a un'istanza Amazon Elastic Compute Cloud (EC2 Amazon), il volume viene decrittografato in modo da poterlo leggere. Per ottenere l'autorizzazione per decrittare i dati, Amazon EBS crea una concessione per ogni volume. Pertanto, se tutti i volumi Amazon EBS utilizzano la stessa chiave KMS, non è possibile collegare più di 50.000 volumi contemporaneamente.

Quote di risorse per gli archivi delle chiavi personalizzate: 10

Puoi creare fino a 10 [archivi di chiavi personalizzati](#) in ciascuna regione Account AWS. Se si tenta di crearne altri, l'[CreateCustomKeyStore](#) operazione non riesce.

Questa quota si applica al numero totale di archivi delle chiavi personalizzate in ogni account e regione, inclusi tutti gli [archivi delle chiavi di AWS CloudHSM](#) e gli [archivi delle chiavi esterne](#), indipendentemente dallo stato della connessione.

Rotazione su richiesta: 10

È possibile eseguire la [rotazione delle chiavi su richiesta](#) un massimo di 10 volte per chiave KMS. Se si tenta di eseguire più rotazioni su richiesta, l'operazione non riesce. [RotateKeyOnDemand](#)

Questa quota non è regolabile. Non è possibile aumentarlo utilizzando Service Quotas o creando un case in. Supporto AWS Per evitare di raggiungere la quota di rotazione su richiesta, consigliamo di utilizzare la rotazione [automatica dei tasti ogni volta che è possibile](#).

Quote di richieste

AWS KMS stabilisce le quote per il numero di operazioni API richieste in ogni secondo. Le quote di richiesta differiscono a seconda del funzionamento dell'API, del Regione AWS e di altri fattori, come il tipo di chiave KMS. Quando superi una quota di richiesta API, limita la AWS KMS [richiesta](#).

Tutte le quote di AWS KMS richiesta sono regolabili, ad eccezione della quota di richiesta del [AWS CloudHSM key store](#). Per richiedere un aumento delle quote, consultare [Richiesta di aumento delle quote](#) nella Guida dell'utente di Service Quotas. Per richiedere una riduzione della quota, modificare una quota non elencata in Service Quotas o modificare una quota in Regione AWS cui Service Quotas AWS KMS for non è disponibile, [Supporto AWS visita il](#) Centro e crea un caso.

Se stai superando la quota richiesta per l'[GenerateDataKey](#) operazione, prendi in considerazione l'utilizzo della funzionalità di memorizzazione nella [cache delle chiavi di dati](#) di AWS Encryption SDK. Il riutilizzo delle chiavi dati potrebbe ridurre la frequenza delle richieste a AWS KMS.

Oltre alle quote di richiesta, AWS KMS utilizza le quote di risorse per garantire la capacità per tutti gli utenti. Per informazioni dettagliate, vedi [Quote delle risorse](#).

Per visualizzare le tendenze nei tassi di richiesta, utilizzare la [Console Service Quotas](#). Puoi anche creare un CloudWatch allarme [Amazon](#) che ti avvisi quando la frequenza delle richieste raggiunge una determinata percentuale del valore di quota. Per i dettagli, consulta [Gestisci le tariffe di richiesta AWS KMS API utilizzando Service Quotas e Amazon CloudWatch](#) nel blog sulla AWS sicurezza.

Argomenti

- [Richiedi quote per ogni operazione API AWS KMS](#)
- [Applicazione delle quote di richieste](#)
- [Quote condivise per le operazioni di crittografia](#)
- [Richieste API eseguite per tuo conto](#)
- [Richieste tra account](#)
- [Quote di richiesta per l'archivio delle chiavi personalizzate](#)

Richiedi quote per ogni operazione API AWS KMS

Questa tabella elenca il codice [di quota Service Quotas](#) e il valore predefinito per ogni quota di AWS KMS richiesta. Tutte le quote di AWS KMS richiesta sono regolabili, ad eccezione della quota di [richiesta del AWS CloudHSM key store](#).

Note

Potrebbe essere necessario scorrere orizzontalmente o verticalmente per visualizzare tutti i dati di questa tabella.

Nome quota	Valore predefinito (richieste al secondo)
Cryptographic operations (symmetric) request rate Si applica a: • Decrypt • Encrypt • GenerateDataKey • GenerateDataKeyWithoutPlainText • GenerateMac • GenerateRandom • ReEncrypt • VerifyMac	Queste quote condivise variano in base al tipo Regione AWS e al tipo di chiave KMS utilizzata nella richiesta. Ogni quota è calcolata separatamente. • 10.000 (condivise) • 20.000 (condivisi) nelle seguenti regioni: • Stati Uniti orientali (Ohio), us-east-2 • Asia Pacifico (Singapore), ap-southeast-1 • Asia Pacifico (Sydney), ap-southeast-2 • Asia Pacifico (Tokyo), ap-northeast-1 • Europa (Francoforte), eu-central-1 • Europa (Londra), eu-west-2 • 100.000 (condivisi) nelle seguenti regioni: • Stati Uniti orientali (Virginia settentrionale), us-east-1 • Stati Uniti occidentali (Oregon), us-west-2 • Europa (Irlanda), eu-west-1
Cryptographic operations (RSA) request rate Si applica a: • Decrypt • Encrypt • ReEncrypt	1.000 (condivise) per le chiavi RSA KMS

Nome quota	Valore predefinito (richieste al secondo)
• Sign • Verify	
Cryptographic operations (ML-DSA) request rate Si applica a: • Sign • Verify	1.000 (condivise) per le chiavi KMS ML-DSA
Cryptographic operations (ECC and SM2) request rate Si applica a: • Decrypt—supportato solo per le chiavi SM2 KMS (solo per le regioni della Cina) • DeriveSharedSecret • Encrypt—supportato solo per le chiavi SM2 KMS (solo per le regioni della Cina) • ReEncrypt —supportato solo per le chiavi SM2 KMS (solo per le regioni della Cina) • Sign • Verify	1.000 (condivise) per le chiavi KMS a curva ellittica (ECC) e (solo per le regioni SM2 della Cina)

Nome quota	Valore predefinito (richieste al secondo)
Custom key store request quotas Si applica a: • Decrypt • DeriveSharedSecret • Encrypt • GenerateDataKey • GenerateDataKeyWithoutPlainText • GenerateRandom • ReEncrypt	Le quote di richiesta per l'archivio delle chiavi personalizzate vengono calcolate separatamente per ogni archivio delle chiavi personalizzate • 1.800 (condivise) per ogni archivio di chiavi AWS CloudHSM • 1.800 (condiviso) per ogni archivio delle chiavi esterne
CancelKeyDeletion request rate	5
ConnectCustomKeyStore request rate	5
CreateAlias request rate	5
CreateCustomKeyStore request rate	5
CreateGrant request rate	50
CreateKey request rate	5
DeleteAlias request rate	15
DeleteCustomKeyStore request rate	5
DeleteImportedKeyMaterial request rate	15
DescribeCustomKeyStores request rate	5
DescribeKey request rate	2000

Nome quota	Valore predefinito (richieste al secondo)
DisableKey request rate	5
DisableKeyRotation request rate	5
DisconnectCustomKeyStore request rate	5
EnableKey request rate	5
EnableKeyRotation request rate	15
GenerateDataKeyPair (ECC_NIST_P256) request rate Si applica a: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100
GenerateDataKeyPair (ECC_NIST_P384) request rate Si applica a: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100
GenerateDataKeyPair (ECC_NIST_P521) request rate Si applica a: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100

Nome quota	Valore predefinito (richieste al secondo)
<code>GenerateDataKeyPair (ECC_SECG_P256K1) request rate</code> Si applica a: • <code>GenerateDataKeyPair</code> • <code>GenerateDataKeyPairWithoutPlaintext</code>	100
<code>GenerateDataKeyPair (RSA_2048) request rate</code> Si applica a: • <code>GenerateDataKeyPair</code> • <code>GenerateDataKeyPairWithoutPlaintext</code>	20
<code>GenerateDataKeyPair (RSA_3072) request rate</code> Si applica a: • <code>GenerateDataKeyPair</code> • <code>GenerateDataKeyPairWithoutPlaintext</code>	4
<code>GenerateDataKeyPair (RSA_4096) request rate</code> Si applica a: • <code>GenerateDataKeyPair</code> • <code>GenerateDataKeyPairWithoutPlaintext</code>	1

Nome quota	Valore predefinito (richieste al secondo)
GenerateDataKeyPair (SM2 – China Regions only) request rate	25
Si applica a:	
- GenerateDataKeyPair - GenerateDataKeyPairWithoutPlaintext	
GetKeyPolicy request rate	1000
GetKeyRotationStatus request rate	1000
GetParametersForImport request rate	0,25 (1 in ogni intervallo di 4 secondi)
GetPublicKey request rate	2000
ImportKeyMaterial request rate	15
ListAliases request rate	500
ListGrants request rate	100
ListKeyPolicies request rate	100
ListKeys request rate	500
ListKeyRotations request rate	100
ListResourceTags request rate	2000
ListRetirableGrants request rate	100
PutKeyPolicy request rate	15

Nome quota	Valore predefinito (richieste al secondo)
ReplicateKey request rate Un'operazione ReplicateKey conta come una richiesta ReplicateKey nella Regione della chiave primaria e due richieste CreateKey nella Regione della replica. Una delle richieste CreateKey serve per rilevare potenziali problemi prima di creare la chiave.	5
RetireGrant request rate	50
RevokeGrant request rate	50
RotateKeyOnDemand request rate	5
ScheduleKeyDeletion request rate	15
TagResource request rate	10
UntagResource request rate	5
UpdateAlias request rate	5
UpdateCustomKeyStore request rate	5
UpdateKeyDescription request rate	5
UpdatePrimaryRegion request rate Un'operazione UpdatePrimaryRegion conta come due richieste UpdatePrimaryRegion; una richiesta in ciascuna delle due Regioni interessate.	5

Applicazione delle quote di richieste

Durante la revisione delle quote di richieste, tieni presente le seguenti informazioni.

- Le quote di richiesta si applicano a entrambe le [chiavi gestite dal cliente](#) e le [Chiavi gestite da AWS](#). L'utilizzo di [Chiavi di proprietà di AWS](#) non rientra nelle quote richieste per l'utente Account AWS, anche quando vengono utilizzate per proteggere le risorse dell'account.
- Le quote di richieste si applicano alle richieste inviate agli endpoint FIPS e agli endpoint non FIPS. Per un elenco degli endpoint del AWS KMS servizio, consulta [AWS Key Management Service endpoint e quote](#) nel. Riferimenti generali di AWS
- Il throttling si basa su tutte le richieste per chiavi KMS di tutti i tipi della Regione. Questo totale include le richieste provenienti da tutti i principali attori del Account AWS, comprese le richieste dei AWS servizi per conto dell'utente.
- Ogni quota di richieste è calcolata in modo indipendente. Ad esempio, le richieste per l'[CreateKey](#) operazione non hanno alcun effetto sulla quota di richieste per l'[CreateAlias](#) operazione. Se alle richieste [CreateAlias](#) è applicato il throttling, è comunque possibile completare le richieste [CreateKey](#).
- Sebbene le operazioni di crittografia condividano una quota, la quota condivisa viene calcolata indipendentemente dalle quote per altre operazioni. Ad esempio, le chiamate alle operazioni [Encrypt](#) e [Decrypt](#) condividono una quota di richieste, ma tale quota è indipendente dalla quota per le operazioni di gestione, ad esempio. [EnableKey](#) Ad esempio, nella Regione Europa (Londra) è possibile eseguire 10.000 operazioni di crittografia su chiavi KMS simmetriche + 5 operazioni [EnableKey](#) al secondo senza che venga applicata alcuna limitazione.

Quote condivise per le operazioni di crittografia

AWS KMS le [operazioni crittografiche condividono le quote](#) di richiesta. Puoi richiedere qualsiasi combinazione di operazioni di crittografia supportate dalla chiave KMS per fare in modo che il numero totale di operazioni di crittografia non superi la quota di richieste per quel tipo di chiave KMS. Le eccezioni sono [GenerateDataKeyPair](#) [GenerateDataKeyPairWithoutPlaintext](#), che condividono una quota separata.

Le quote per i diversi tipi di chiavi KMS vengono calcolate in modo indipendente. Ogni quota si applica a tutte le richieste di queste operazioni nella regione Account AWS and con il tipo di chiave specificato in ogni intervallo di un secondo.

- Il tasso di richieste di operazioni di crittografia (simmetriche) è la quota di richiesta condivisa per le operazioni di crittografia che utilizzano le chiavi KMS simmetriche in un account e in una Regione. Questa quota si applica alle operazioni crittografiche con chiavi crittografiche simmetrica e chiavi HMAC, anch'esse simmetriche.

Ad esempio, potresti utilizzare [chiavi KMS simmetriche](#) in un file Regione AWS con una quota condivisa di 10.000 richieste al secondo. Quando effettui 7.000 [GenerateDataKey](#) richieste al secondo e 2.000 richieste di [decriptazione](#) al secondo, AWS KMS non limita le tue richieste. Se invece si effettuano 9.500 richieste [GenerateDataKey](#) e 1.000 richieste [Encrypt](#) al secondo, AWS KMS applica il throttling alle richieste perché queste superano la quota condivisa.

Le operazioni di crittografia sulle [chiavi KMS di crittografia simmetrica](#) in un [archivio di chiavi personalizzate](#) contano sia per la frequenza di richiesta (simmetrica) di operazioni crittografiche per l'account sia per la [quota di richiesta dell'archivio di chiavi personalizzate](#) per l'archivio di chiavi personalizzate.

- Il tasso di richieste di operazioni di crittografia (RSA) è la quota di richieste condivisa per le operazioni di crittografia che utilizzano le [chiavi KMS RSA asimmetriche](#).

Ad esempio, con una quota di richieste di 1.000 operazioni al secondo, puoi effettuare 400 richieste di crittografia e 200 richieste di crittografia con chiavi RSA KMS in grado di crittografare e decrittografare, oltre a 250 richieste di firma e 150 richieste di verifica con chiavi RSA KMS in grado di firmare e verificare.

- La frequenza di richiesta delle operazioni crittografiche (ECC) è la quota di richieste condivisa per le operazioni crittografiche che utilizzano chiavi [KMS asimmetriche a curva ellittica \(ECC\) e chiavi KMS asimmetriche SM](#).

Ad esempio, con una quota di richieste di 1.000 operazioni al secondo, puoi effettuare 400 richieste Sign e 200 richieste Verify con chiavi KMS ECC in grado di firmare e verificare, oltre a 250 richieste Sign e 150 richieste Verify con chiavi KMS in grado di firmare e verificare. SM2

- La quota di richiesta per l'archivio delle chiavi personalizzate è la quota di richiesta condivisa per le operazioni di crittografia sulle chiavi KMS in un archivio delle chiavi personalizzate. Questa quota viene calcolata separatamente per ogni archivio delle chiavi personalizzate.

Le operazioni di crittografia sulle [chiavi KMS di crittografia simmetrica](#) in un [archivio di chiavi personalizzate](#) contano sia per la frequenza di richiesta (simmetrica) di operazioni crittografiche per l'account sia per la [quota di richiesta dell'archivio di chiavi personalizzate](#) per l'archivio di chiavi personalizzate.

Anche le quote per i diversi tipi di chiave sono calcolate in modo indipendente. Ad esempio, nella regione Asia Pacifico (Singapore), se utilizzi chiavi KMS simmetriche e asimmetriche, puoi effettuare fino a 10.000 chiamate al secondo con le chiavi KMS simmetriche (incluse le chiavi HMAC), più fino

a 500 chiamate aggiuntive al secondo con le chiavi KMS asimmetriche RSA, più fino a 300 richieste aggiuntive al secondo con le chiavi KMS basate su ECC.

Richieste API eseguite per tuo conto

Puoi effettuare richieste API direttamente o utilizzando un AWS servizio integrato che effettua richieste API per tuo AWS KMS conto. La quota si applica a entrambi i tipi di richieste.

Ad esempio, è possibile archiviare i dati in Amazon S3 utilizzando la crittografia lato server con una chiave KMS (SSE-KMS). Ogni volta che carichi o scarichi un oggetto S3 crittografato con SSE-KMS, Amazon S3 invia una richiesta (per i caricamenti) o `GenerateDataKey Decrypt` (per i download) a tuo nome. AWS KMS Queste richieste vengono conteggiate ai fini della tua quota, quindi AWS KMS limitano le richieste se superi un totale combinato di 5.500 (o 10.000 o 50.000 a seconda della tua Regione AWS) caricamenti o download al secondo di oggetti S3 crittografati con SSE-KMS.

Richieste tra account

Quando un'applicazione in un'unica applicazione Account AWS utilizza una chiave KMS di proprietà di un altro account, si parla di richiesta tra account. Per le richieste tra account, AWS KMS limita l'account che effettua le richieste, non l'account proprietario della chiave KMS. Ad esempio, se un'applicazione nell'account A utilizza una chiave KMS nell'account B, l'uso della chiave KMS viene applicato solo alle quote dell'account A.

Quote di richiesta per l'archivio delle chiavi personalizzate

AWS KMS [mantiene le quote di richiesta per le operazioni crittografiche sulle chiavi KMS in un archivio di chiavi personalizzato](#). Tali quote di richiesta vengono calcolate separatamente per ogni archivio delle chiavi personalizzate.

Quote di richiesta per l'archivio delle chiavi personalizzate	Valore predefinito (richieste al secondo) per ogni archivio delle chiavi personalizzate	Regolabile
AWS CloudHSM quota di richiesta dell'archivio di chiavi	1800	No
Quota di richiesta per l' archivio delle chiavi esterne	1800	Sì

 Note

AWS KMS le [quote di richieste di archiviazione chiavi personalizzate non vengono visualizzate nella console Service Quotas](#). Non puoi visualizzare o gestire tali quote utilizzando le operazioni API Service Quotas. Per richiedere una modifica delle quote di richiesta per l'archivio delle chiavi esterne, visita il [Centro Supporto AWS](#) e crea un caso. Se il AWS CloudHSM cluster associato a un AWS CloudHSM key store sta elaborando numerosi comandi, inclusi quelli non correlati all'archivio chiavi personalizzato, potresti riceverne uno AWS KMS ThrottlingException a pagamento. lower-than-expected In tal caso, riduci la frequenza delle richieste a AWS KMS, riduci il carico non correlato o utilizza un AWS CloudHSM cluster dedicato per l'archivio delle AWS CloudHSM chiavi. AWS KMS segnala la limitazione delle richieste di archiviazione di chiavi esterne nella metrica. [ExternalKeyStoreThrottle](#) CloudWatch Puoi utilizzare questo parametro per visualizzare gli schemi di limitazione, creare allarmi e modificare la quota di richiesta dell'archivio delle chiavi esterne.

Una richiesta di un'[operazione di crittografia](#) su una chiave KMS in un archivio delle chiavi personalizzate viene conteggiata ai fini di due quote:

- Quota di frequenza della richiesta (per account) di operazioni di crittografia (simmetriche)

Le richieste di operazioni di crittografia sulle chiavi KMS in un archivio delle chiavi personalizzate vengono conteggiate ai fini della quota `Cryptographic operations (symmetric) request rate` per ciascuna regione e Account AWS . Ad esempio, negli Stati Uniti orientali (Virginia settentrionale) (us-east-1), Account AWS ognuno può ricevere fino a 100.000 richieste al secondo su chiavi KMS con crittografia simmetrica, incluse le richieste che utilizzano una chiave KMS in un archivio di chiavi personalizzato.

- Quota di richiesta dell'archivio di chiavi personalizzate (per archivio delle chiavi personalizzate)

Le richieste di operazioni di crittografia sulle chiavi KMS in un archivio delle chiavi personalizzate contano anche per una `Custom key store request` quota di 1.800 operazioni al secondo. Tali quote vengono calcolate separatamente per ogni archivio delle chiavi personalizzate. Potrebbero includere richieste provenienti da più utenti Account AWS che utilizzano chiavi KMS nell'archivio chiavi personalizzato.

Ad esempio, un'operazione di [crittografia](#) su una chiave KMS in un archivio chiavi personalizzato (di entrambi i tipi) nella regione Stati Uniti orientali (Virginia settentrionale) (us-east-1) viene conteggiata ai fini della quota a Cryptographic operations (symmetric) request rate livello di account (100.000 richieste al secondo) per l'account e la regione e per una Custom key store request quota (1.800 richieste al secondo) per l'archivio chiavi personalizzato. Tuttavia, una richiesta per un'operazione di gestione [PutKeyPolicy](#), ad esempio su una chiave KMS in un archivio chiavi personalizzato, si applica solo alla quota a livello di account (15 richieste al secondo).

Richieste di limitazione AWS KMS

Per garantire che AWS KMS possa fornire risposte rapide e affidabili alle richieste API di tutti i clienti, limita le richieste API che superano determinati limiti.

La limitazione si verifica quando AWS KMS rifiuta una richiesta che altrimenti potrebbe essere valida e restituisce un `ThrottlingException` errore come il seguente.

```
You have exceeded the rate at which you may call KMS. Reduce the frequency of your calls.
(Service: AWSKMS; Status Code: 400; Error Code: ThrottlingException; Request ID: <ID>
```

AWS KMS limita le richieste per le seguenti condizioni.

- La frequenza di richieste al secondo supera la [quota di AWS KMS richieste](#) per un account e una regione.

Ad esempio, se gli utenti del tuo account inviano 1000 `DescribeKey` richieste in un secondo, limita AWS KMS tutte le `DescribeKey` richieste successive in quel secondo.

Per rispondere alla limitazione, utilizza una [strategia di backoff e riprova](#). In alcuni casi, questa strategia viene implementata automaticamente per gli errori HTTP 400. AWS SDKs

- Una frammentazione o una frequenza elevata sostenuta di richieste per modificare lo stato della stessa chiave KMS. Questa condizione è spesso nota come "tasto di scelta rapida".

Ad esempio, se un'applicazione del tuo account invia una raffica persistente di `DisableKey` richieste `EnableKey` e per la stessa chiave KMS, limita le AWS KMS richieste. Questa limitazione si verifica anche se le richieste non superano il limite di richieste per le operazioni request-per-second and. `EnableKey` `DisableKey`

Per rispondere alla limitazione, modifica la logica dell'applicazione in modo che faccia solo richieste richieste o consolidi le richieste di più funzioni.

- Le richieste di operazioni sulle chiavi KMS in un [AWS CloudHSM key store](#) potrebbero essere limitate a una certa lower-than-expected velocità quando il AWS CloudHSM cluster associato all'archivio AWS CloudHSM chiavi elabora numerosi comandi, inclusi quelli non correlati all'archivio chiavi. AWS CloudHSM

(AWS KMS non limita più le richieste di operazioni sulle chiavi KMS in un archivio di chiavi quando non ci sono AWS CloudHSM sessioni PKCS #11 disponibili per il cluster. AWS CloudHSM Al contrario, genera un messaggio `KMSInternalException` e consiglia di riprovare la richiesta.)

Per visualizzare le tendenze nei tassi di richiesta, utilizzare la [Console Service Quotas](#). Puoi anche creare un CloudWatch allarme [Amazon](#) che ti avvisi quando la frequenza delle richieste raggiunge una determinata percentuale del valore di quota. Per i dettagli, consulta [Gestisci le tariffe di richiesta AWS KMS API utilizzando Service Quotas e Amazon CloudWatch](#) nel blog sulla AWS sicurezza.

Tutte le AWS KMS quote sono regolabili, ad eccezione della quota di [risorse di rotazione su richiesta e della quota](#) di richieste del [AWS CloudHSM key store](#). Per richiedere un aumento delle quote, consultare [Richiesta di aumento delle quote](#) nella Guida dell'utente di Service Quotas. Per richiedere una riduzione della quota, modificare una quota non elencata in Service Quotas o modificare una quota in Regione AWS cui Service Quotas AWS KMS for non è disponibile, [Supporto AWS visita il Centro](#) e crea un caso.

Note

AWS KMS le [quote di richieste di archiviazione chiavi personalizzate non vengono visualizzate nella console Service Quotas](#). Non puoi visualizzare o gestire tali quote utilizzando le operazioni API Service Quotas. Per richiedere una modifica delle quote di richiesta per l'archivio delle chiavi esterne, visita il [Centro Supporto AWS](#) e crea un caso.

Esempi di codice per AWS KMS l'utilizzo AWS SDKs

I seguenti esempi di codice mostrano come utilizzarlo AWS KMS con un kit di sviluppo AWS software (SDK).

Le nozioni di base sono esempi di codice che mostrano come eseguire le operazioni essenziali all'interno di un servizio.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati.

Gli scenari sono esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio o combinate con altri Servizi AWS.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Nozioni di base

Ciao AWS Key Management Service

L'esempio di codice seguente mostra come iniziare a utilizzare AWS Key Management Service.

Java

SDK per Java 2.x

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
import software.amazon.awssdk.services.kms.KmsAsyncClient;
import software.amazon.awssdk.services.kms.model.ListKeysRequest;
import software.amazon.awssdk.services.kms.paginators.ListKeysPublisher;
import java.util.concurrent.CompletableFuture;

/**
```

```

* Before running this Java V2 code example, set up your development
* environment, including your credentials.
*
* For more information, see the following documentation topic:
*
* https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
*/
public class HelloKMS {
    public static void main(String[] args) {
        listAllKeys();
    }

    public static void listAllKeys() {
        KmsAsyncClient kmsAsyncClient = KmsAsyncClient.builder()
            .build();
        ListKeysRequest listKeysRequest = ListKeysRequest.builder()
            .limit(15)
            .build();

        /*
         * The `subscribe` method is required when using paginator methods in the
         AWS SDK
         * because paginator methods return an instance of a `ListKeysPublisher`,
         which is
         * based on a reactive stream. This allows asynchronous retrieval of
         paginated
         * results as they become available. By subscribing to the stream, we can
         process
         * each page of results as they are emitted.
         */
        ListKeysPublisher keysPublisher =
            kmsAsyncClient.listKeysPaginator(listKeysRequest);
        CompletableFuture<Void> future = keysPublisher
            .subscribe(r -> r.keys().forEach(key ->
                System.out.println("The key ARN is: " + key.keyArn() + ". The key
Id is: " + key.keyId()))
            .whenComplete((result, exception) -> {
                if (exception != null) {
                    System.err.println("Error occurred: " +
exception.getMessage());
                } else {
                    System.out.println("Successfully listed all keys.");
                }
            })
    }
}

```

```
        });

        try {
            future.join();
        } catch (Exception e) {
            System.err.println("Failed to list keys: " + e.getMessage());
        }
    }
}
```

- Per i dettagli sull'API, [ListKeys](#) consulta AWS SDK for Java 2.x API Reference.

PHP

SDK per PHP

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
include "vendor/autoload.php";

use Aws\Kms\KmsClient;

echo "This file shows how to connect to the KmsClient, uses a paginator to get
the keys for the account, and lists the KeyIds for up to 10 keys.\n";

$client = new KmsClient([]);

$pageLength = 10; // Change this value to change the number of records shown, or
to break up the result into pages.

$keys = [];
$keysPaginator = $client->getPaginator("ListKeys", ['Limit' => $pageLength]);
foreach($keysPaginator as $page){
    foreach($page['Keys'] as $index => $key){
        echo "The $index index Key's ID is: {$key['KeyId']}\n";
    }
}
```

```
    echo "End of page one of results. Alter the \${pageLength} variable to see more
    results.\n";
    break;
}
```

- Per i dettagli sull'API, [ListKeys](#) consulta AWS SDK per PHP API Reference.

Esempi di codice

- [Esempi di base per AWS KMS l'utilizzo AWS SDKs](#)
 - [Ciao AWS Key Management Service](#)
 - [Scopri le nozioni di base di AWS KMS con un SDK AWS](#)
 - [Azioni per l'utilizzo AWS KMSAWS SDKs](#)
 - [Utilizzo CreateAlias con un AWS SDK o una CLI](#)
 - [Utilizzo CreateGrant con un AWS SDK o una CLI](#)
 - [Utilizzo CreateKey con un AWS SDK o una CLI](#)
 - [Utilizzo Decrypt con un AWS SDK o una CLI](#)
 - [Utilizzo DeleteAlias con un AWS SDK o una CLI](#)
 - [Utilizzo DescribeKey con un AWS SDK o una CLI](#)
 - [Utilizzo DisableKey con un AWS SDK o una CLI](#)
 - [Utilizzo EnableKey con un AWS SDK o una CLI](#)
 - [Utilizzo EnableKeyRotation con un AWS SDK o una CLI](#)
 - [Utilizzo Encrypt con un AWS SDK o una CLI](#)
 - [Utilizzo GenerateDataKey con un AWS SDK o una CLI](#)
 - [Utilizzo GenerateDataKeyWithoutPlaintext con un AWS SDK o una CLI](#)
 - [Utilizzo GenerateRandom con un AWS SDK o una CLI](#)
 - [Utilizzo GetKeyPolicy con un AWS SDK o una CLI](#)
 - [Utilizzo ListAliases con un AWS SDK o una CLI](#)
 - [Utilizzo ListGrants con un AWS SDK o una CLI](#)
 - [Utilizzo ListKeyPolicies con un AWS SDK o una CLI](#)
 - [Utilizzo ListKeys con un AWS SDK o una CLI](#)
 - [Utilizzo PutKeyPolicy con un AWS SDK o una CLI](#)

- [Utilizzo ReEncrypt con un AWS SDK o una CLI](#)
- [Utilizzo RetireGrant con un AWS SDK o una CLI](#)
- [Utilizzo RevokeGrant con un AWS SDK o una CLI](#)
- [Utilizzo ScheduleKeyDeletion con un AWS SDK o una CLI](#)
- [Utilizzo Sign con un AWS SDK o una CLI](#)
- [Utilizzo TagResource con un AWS SDK o una CLI](#)
- [Utilizzo UpdateAlias con un AWS SDK o una CLI](#)
- [Utilizzo Verify con un AWS SDK o una CLI](#)
- [Scenari di utilizzo AWS KMSAWS SDKs](#)
 - [Lavora con la crittografia delle tabelle DynamoDB utilizzando v2 AWS Command Line Interface](#)

Esempi di base per AWS KMS l'utilizzo AWS SDKs

I seguenti esempi di codice mostrano come utilizzare le nozioni di base di AWS Key Management Service with. AWS SDKs

Esempi

- [Ciao AWS Key Management Service](#)
- [Scopri le nozioni di base di AWS KMS con un SDK AWS](#)
- [Azioni per l'utilizzo AWS KMSAWS SDKs](#)
 - [Utilizzo CreateAlias con un AWS SDK o una CLI](#)
 - [Utilizzo CreateGrant con un AWS SDK o una CLI](#)
 - [Utilizzo CreateKey con un AWS SDK o una CLI](#)
 - [Utilizzo Decrypt con un AWS SDK o una CLI](#)
 - [Utilizzo DeleteAlias con un AWS SDK o una CLI](#)
 - [Utilizzo DescribeKey con un AWS SDK o una CLI](#)
 - [Utilizzo DisableKey con un AWS SDK o una CLI](#)
 - [Utilizzo EnableKey con un AWS SDK o una CLI](#)
 - [Utilizzo EnableKeyRotation con un AWS SDK o una CLI](#)
 - [Utilizzo Encrypt con un AWS SDK o una CLI](#)
 - [Utilizzo GenerateDataKey con un AWS SDK o una CLI](#)

- [Utilizzo GenerateDataKeyWithoutPlaintext con un AWS SDK o una CLI](#)
- [Utilizzo GenerateRandom con un AWS SDK o una CLI](#)
- [Utilizzo GetKeyPolicy con un AWS SDK o una CLI](#)
- [Utilizzo ListAliases con un AWS SDK o una CLI](#)
- [Utilizzo ListGrants con un AWS SDK o una CLI](#)
- [Utilizzo ListKeyPolicies con un AWS SDK o una CLI](#)
- [Utilizzo ListKeys con un AWS SDK o una CLI](#)
- [Utilizzo PutKeyPolicy con un AWS SDK o una CLI](#)
- [Utilizzo ReEncrypt con un AWS SDK o una CLI](#)
- [Utilizzo RetireGrant con un AWS SDK o una CLI](#)
- [Utilizzo RevokeGrant con un AWS SDK o una CLI](#)
- [Utilizzo ScheduleKeyDeletion con un AWS SDK o una CLI](#)
- [Utilizzo Sign con un AWS SDK o una CLI](#)
- [Utilizzo TagResource con un AWS SDK o una CLI](#)
- [Utilizzo UpdateAlias con un AWS SDK o una CLI](#)
- [Utilizzo Verify con un AWS SDK o una CLI](#)

Ciao AWS Key Management Service

L'esempio di codice seguente mostra come iniziare a utilizzare AWS Key Management Service.

Java

SDK per Java 2.x

Note

C'è altro su [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
import software.amazon.awssdk.services.kms.KmsAsyncClient;
import software.amazon.awssdk.services.kms.model.ListKeysRequest;
import software.amazon.awssdk.services.kms.paginators.ListKeysPublisher;
import java.util.concurrent.CompletableFuture;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class HelloKMS {
    public static void main(String[] args) {
        listAllKeys();
    }

    public static void listAllKeys() {
        KmsAsyncClient kmsAsyncClient = KmsAsyncClient.builder()
            .build();
        ListKeysRequest listKeysRequest = ListKeysRequest.builder()
            .limit(15)
            .build();

        /*
         * The `subscribe` method is required when using paginator methods in the
         AWS SDK
         * because paginator methods return an instance of a `ListKeysPublisher`,
         which is
         * based on a reactive stream. This allows asynchronous retrieval of
         paginated
         * results as they become available. By subscribing to the stream, we can
         process
         * each page of results as they are emitted.
         */
        ListKeysPublisher keysPublisher =
            kmsAsyncClient.listKeysPaginator(listKeysRequest);
        CompletableFuture<Void> future = keysPublisher
            .subscribe(r -> r.keys().forEach(key ->
                System.out.println("The key ARN is: " + key.keyArn() + ". The key
                Id is: " + key.keyId()))
            .whenComplete((result, exception) -> {
                if (exception != null) {
                    System.err.println("Error occurred: " +
                        exception.getMessage());
                } else {

```

```

        System.out.println("Successfully listed all keys.");
    }
    });

    try {
        future.join();
    } catch (Exception e) {
        System.err.println("Failed to list keys: " + e.getMessage());
    }
}
}

```

- Per i dettagli sull'API, [ListKeys](#) consulta AWS SDK for Java 2.x API Reference.

PHP

SDK per PHP

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

include "vendor/autoload.php";

use Aws\Kms\KmsClient;

echo "This file shows how to connect to the KmsClient, uses a paginator to get
the keys for the account, and lists the KeyIds for up to 10 keys.\n";

$client = new KmsClient([]);

$pageLength = 10; // Change this value to change the number of records shown, or
to break up the result into pages.

$keys = [];
$keysPaginator = $client->getPaginator("ListKeys", ['Limit' => $pageLength]);
foreach($keysPaginator as $page){
    foreach($page['Keys'] as $index => $key){
        echo "The $index index Key's ID is: {$key['KeyId']}\n";
    }
}

```

```
}  
    echo "End of page one of results. Alter the \${pageLength} variable to see more  
results.\n";  
    break;  
}
```

- Per i dettagli sull'API, [ListKeys](#) consulta AWS SDK per PHP API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un AWS SDK](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scopri le nozioni di base di AWS KMS con un SDK AWS

Gli esempi di codice seguenti mostrano come:

- Creare una chiave KMS.
- Elenca le chiavi KMS del tuo account e ottieni dettagli su di esse.
- Abilitare e disabilitare le chiavi KMS.
- Genera una chiave dati simmetrica che può essere utilizzata per la crittografia lato client.
- Genera una chiave asimmetrica utilizzata per firmare digitalmente i dati.
- Chiavi per tag.
- Eliminare le chiavi KMS.

Java

SDK per Java 2.x

Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Esegui uno scenario al prompt dei comandi.

```
import software.amazon.awssdk.core.SdkBytes;
import software.amazon.awssdk.regions.Region;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.kms.model.AlreadyExistsException;
import software.amazon.awssdk.services.kms.model.DisabledException;
import software.amazon.awssdk.services.kms.model.EnableKeyRotationResponse;
import software.amazon.awssdk.services.kms.model.KmsException;
import software.amazon.awssdk.services.kms.model.NotFoundException;
import software.amazon.awssdk.services.kms.model.RevokeGrantResponse;
import java.util.List;
import java.util.Scanner;
import java.util.concurrent.CompletableFuture;
import java.util.concurrent.CompletionException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */

public class KMSScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
    "-");
    private static String accountId = "";

    private static final Logger logger =
    LoggerFactory.getLogger(KMSScenario.class);

    static KMSActions kmsActions = new KMSActions();

    static Scanner scanner = new Scanner(System.in);

    static String aliasName = "alias/dev-encryption-key";

    public static void main(String[] args) {
        final String usage = ""
            Usage: <granteePrincipal>
```

Where:

granteePrincipal - The principal (user, service account, or group) to whom the grant or permission is being given.

```
"";
```

```
if (args.length != 1) {  
    logger.info(usage);  
    return;  
}
```

```
String granteePrincipal = args[0];  
String policyName = "default";
```

```
accountId = kmsActions.getAccountId();  
String keyDesc = "Created by the AWS KMS API";
```

```
logger.info(DASHES);  
logger.info("""
```

```
    Welcome to the AWS Key Management SDK Basics scenario.
```

This program demonstrates how to interact with AWS Key Management using the AWS SDK for Java (v2).

The AWS Key Management Service (KMS) is a secure and highly available service that allows you to create

and manage AWS KMS keys and control their use across a wide range of AWS services and applications.

KMS provides a centralized and unified approach to managing encryption keys, making it easier to meet your data protection and regulatory compliance requirements.

This Basics scenario creates two key types:

- A symmetric encryption key is used to encrypt and decrypt data.
- An asymmetric key used to digitally sign data.

Let's get started...

```
""");
```

```
waitForInputToContinue(scanner);
```

```
try {  
    // Run the methods that belong to this scenario.  
    String targetKeyId = runScenario(granteePrincipal, keyDesc, policyName);  
    requestDeleteResources(aliasName, targetKeyId);
```

```
} catch (Throwable rt) {
```

```

        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
    }
}

private static String runScenario(String granteePrincipal, String keyDesc,
String policyName) throws Throwable {
    logger.info(DASHES);
    logger.info("1. Create a symmetric KMS key\n");
    logger.info("First, the program will creates a symmetric KMS key that you
can used to encrypt and decrypt data.");
    waitForInputToContinue(scanner);
    String targetKeyId;
    try {
        CompletableFuture<String> futureKeyId =
kmsActions.createKeyAsync(keyDesc);
        targetKeyId = futureKeyId.join();
        logger.info("A symmetric key was successfully created " +
targetKeyId);

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        throw cause;
    }
    waitForInputToContinue(scanner);

    logger.info(DASHES);
    logger.info("""
        2. Enable a KMS key

        By default, when the SDK creates an AWS key, it is enabled. The next
bit of code checks to
        determine if the key is enabled.

```

```

        "");
        waitForInputToContinue(scanner);
        boolean isEnabled;
        try {
            CompletableFuture<Boolean> futureIsKeyEnabled =
kmsActions.isKeyEnabledAsync(targetKeyId);
            isEnabled = futureIsKeyEnabled.join();
            logger.info("Is the key enabled? {}", isEnabled);

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            throw cause;
        }

        if (!isEnabled)
            try {
                CompletableFuture<Void> future =
kmsActions.enableKeyAsync(targetKeyId);
                future.join();

            } catch (RuntimeException rt) {
                Throwable cause = rt.getCause();
                if (cause instanceof KmsException kmsEx) {
                    logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
                } else {
                    logger.info("An unexpected error occurred: " +
rt.getMessage());
                }
                throw cause;
            }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("3. Encrypt data using the symmetric KMS key");
        String plaintext = "Hello, AWS KMS!";
        logger.info("

```


One of the main uses of symmetric keys is to encrypt and decrypt data.

Next, the code encrypts the string {} with the SYMMETRIC_DEFAULT encryption algorithm.

```

        "", plaintext);
    waitForInputToContinue(scanner);
    SdkBytes encryptedData;
    try {
        CompletableFuture<SdkBytes> future =
kmsActions.encryptDataAsync(targetKeyId, plaintext);
        encryptedData = future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof DisabledException kmsDisabledEx) {
            logger.info("KMS error occurred due to a disabled
key: Error message: {}, Error code {}", kmsDisabledEx.getMessage(),
kmsDisabledEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        deleteKey(targetKeyId);
        throw cause;
    }
    waitForInputToContinue(scanner);

    logger.info(DASHES);
    logger.info("4. Create an alias");
    logger.info("

    The alias name should be prefixed with 'alias/'.
    The default, 'alias/dev-encryption-key'.
    ");
    waitForInputToContinue(scanner);

    try {
        CompletableFuture<Void> future =
kmsActions.createCustomAliasAsync(targetKeyId, aliasName);
        future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof AlreadyExistsException kmsExistsEx) {
            if (kmsExistsEx.getMessage().contains("already exists")) {

```

```

        logger.info("The alias '" + aliasName + "' already exists.
Moving on...");
    }
    } else {
        logger.error("An unexpected error occurred: " + rt.getMessage(),
rt);
        deleteKey(targetKeyId);
        throw cause;
    }
}
waitForInputToContinue(scanner);

logger.info(DASHES);
logger.info("5. List all of your aliases");
waitForInputToContinue(scanner);
try {
    CompletableFuture<Object> future = kmsActions.listAllAliasesAsync();
    future.join();

} catch (RuntimeException rt) {
    Throwable cause = rt.getCause();
    if (cause instanceof KmsException kmsEx) {
        logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
    } else {
        logger.info("An unexpected error occurred: " + rt.getMessage());
    }
    deleteAliasName(aliasName);
    deleteKey(targetKeyId);
    throw cause;
}
waitForInputToContinue(scanner);

logger.info(DASHES);
logger.info("6. Enable automatic rotation of the KMS key");
logger.info("

    By default, when the SDK enables automatic rotation of a KMS key,
    KMS rotates the key material of the KMS key one year (approximately
365 days) from the enable date and every year
    thereafter.
    ");
waitForInputToContinue(scanner);
try {

```

```

        CompletableFuture<EnableKeyRotationResponse> future =
kmsActions.enableKeyRotationAsync(targetKeyId);
        future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        deleteAliasName(aliasName);
        deleteKey(targetKeyId);
        throw cause;
    }
    waitForInputToContinue(scanner);

    logger.info(DASHES);
    logger.info("
7. Create a grant

A grant is a policy instrument that allows Amazon Web Services
principals to use KMS keys.
It also can allow them to view a KMS key (DescribeKey) and create and
manage grants.
When authorizing access to a KMS key, grants are considered along
with key policies and IAM policies.
");

    waitForInputToContinue(scanner);
    String grantId = null;
    try {
        CompletableFuture<String> futureGrantId =
kmsActions.grantKeyAsync(targetKeyId, granteePrincipal);
        grantId = futureGrantId.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());

```

```

        }
        deleteKey(targetKeyId);
        throw cause;
    }
    waitForInputToContinue(scanner);
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("8. List grants for the KMS key");
    waitForInputToContinue(scanner);
    try {
        CompletableFuture<Object> future =
kmsActions.displayGrantIdsAsync(targetKeyId);
        future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        deleteAliasName(aliasName);
        deleteKey(targetKeyId);
        throw cause;
    }
    waitForInputToContinue(scanner);

    logger.info(DASHES);
    logger.info("9. Revoke the grant");
    logger.info("""
        The revocation of a grant immediately removes the permissions and
        access that the grant had provided.
        This means that any principal (user, role, or service) that was
        granted access to perform specific
        KMS operations on a KMS key will no longer be able to perform those
        operations.
        """);
    waitForInputToContinue(scanner);
    try {
        CompletableFuture<RevokeGrantResponse> future =
kmsActions.revokeKeyGrantAsync(targetKeyId, grantId);
        future.join();
    }

```

```

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            if (kmsEx.getMessage().contains("Grant does not exist")) {
                logger.info("The grant ID '" + grantId + "' does not exist.
Moving on...");
            } else {
                logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
                throw cause;
            }
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
    }
}
waitForInputToContinue(scanner);

logger.info(DASHES);
logger.info("10. Decrypt the data\n");
logger.info("""
    Lets decrypt the data that was encrypted in an early step.
    The code uses the same key to decrypt the string that we encrypted
earlier in the program.
    """);
waitForInputToContinue(scanner);
String decryptedData = "";
try {
    CompletableFuture<String> future =
kmsActions.decryptDataAsync(encryptedData, targetKeyId);
    decryptedData = future.join();
    logger.info("Decrypted data: " + decryptedData);

} catch (RuntimeException rt) {
    Throwable cause = rt.getCause();
    if (cause instanceof KmsException kmsEx) {
        logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
    } else {
        logger.info("An unexpected error occurred: " + rt.getMessage());
    }
}

```

```

        deleteAliasName(aliasName);
        deleteKey(targetKeyId);
        throw cause;
    }
    logger.info("Decrypted text is: " + decryptedData);
    waitForInputToContinue(scanner);

    logger.info(DASHES);
    logger.info("11. Replace a key policy\n");
    logger.info("""
        A key policy is a resource policy for a KMS key. Key policies are the
primary way to control
        access to KMS keys. Every KMS key must have exactly one key policy.
The statements in the key policy
        determine who has permission to use the KMS key and how they can use
it.

        You can also use IAM policies and grants to control access to the KMS
key, but every KMS key
        must have a key policy.

        By default, when you create a key by using the SDK, a policy is
created that
        gives the AWS account that owns the KMS key full access to the KMS
key.

        Let's try to replace the automatically created policy with the
following policy.

        "Version":"2012-10-17",
        "Statement": [{
        "Effect": "Allow",
        "Principal": {"AWS": "arn:aws:iam::000000000000:root"},
        "Action": "kms:*",
        "Resource": "*"
        }]
        """);

    waitForInputToContinue(scanner);
    try {
        CompletableFuture<Boolean> future =
kmsActions.replacePolicyAsync(targetKeyId, policyName, accountId);
        boolean success = future.join();
        if (success) {
            logger.info("Key policy replacement succeeded.");

```

```

        } else {
            logger.error("Key policy replacement failed.");
        }

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        deleteAliasName(aliasName);
        deleteKey(targetKeyId);
        throw cause;
    }
    waitForInputToContinue(scanner);

    logger.info(DASHES);
    logger.info("12. Get the key policy\n");
    logger.info("The next bit of code that runs gets the key policy to make
sure it exists.");
    waitForInputToContinue(scanner);
    try {
        CompletableFuture<String> future =
kmsActions.getKeyPolicyAsync(targetKeyId, policyName);
        String policy = future.join();
        if (!policy.isEmpty()) {
            logger.info("Retrieved policy: " + policy);
        }
    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        deleteAliasName(aliasName);
        deleteKey(targetKeyId);
        throw cause;
    }
    waitForInputToContinue(scanner);

```

```

        logger.info(DASHES);
        logger.info("13. Create an asymmetric KMS key and sign your data\n");
        logger.info("""
            Signing your data with an AWS key can provide several benefits that
            make it an attractive option
            for your data signing needs. By using an AWS KMS key, you can
            leverage the
            security controls and compliance features provided by AWS,
            which can help you meet various regulatory requirements and enhance
            the overall security posture
            of your organization.
            """);
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Boolean> future = kmsActions.signVerifyDataAsync();
            boolean success = future.join();
            if (success) {
                logger.info("Sign and verify data operation succeeded.");
            } else {
                logger.error("Sign and verify data operation failed.");
            }
        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("14. Tag your symmetric KMS Key\n");
        logger.info("""
            By using tags, you can improve the overall management, security, and
            governance of your
            KMS keys, making it easier to organize, track, and control access to
            your encrypted data within

```



```

        your AWS environment
        "");
    waitForInputToContinue(scanner);
    try {
        CompletableFuture<Void> future =
kmsActions.tagKMSKeyAsync(targetKeyId);
        future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
        deleteAliasName(aliasName);
        deleteKey(targetKeyId);
        throw cause;
    }
    waitForInputToContinue(scanner);
    return targetKeyId;
}

// Deletes KMS resources with user input.
private static void requestDeleteResources(String aliasName, String
targetKeyId) {
    logger.info(DASHES);
    logger.info("15. Schedule the deletion of the KMS key\n");
    logger.info("
        By default, KMS applies a waiting period of 30 days,
        but you can specify a waiting period of 7-30 days. When this
operation is successful,
        the key state of the KMS key changes to PendingDeletion and the key
can't be used in any
        cryptographic operations. It remains in this state for the duration
of the waiting period.

        Deleting a KMS key is a destructive and potentially dangerous
operation. When a KMS key is deleted,
        all data that was encrypted under the KMS key is unrecoverable.
        ");
    logger.info("Would you like to delete the Key Management resources? (y/
n)");

```

```
String delAns = scanner.nextLine().trim();
if (delAns.equalsIgnoreCase("y")) {
    logger.info("You selected to delete the AWS KMS resources.");
    waitForInputToContinue(scanner);
    try {
        CompletableFuture<Void> future =
kmsActions.deleteSpecificAliasAsync(aliasName);
        future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " +
rt.getMessage());
        }
    }
    waitForInputToContinue(scanner);
    try {
        CompletableFuture<Void> future =
kmsActions.disableKeyAsync(targetKeyId);
        future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " +
rt.getMessage());
        }
    }

    try {
        CompletableFuture<Void> future =
kmsActions.deleteKeyAsync(targetKeyId);
        future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
```

```

        logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
    } else {
        logger.info("An unexpected error occurred: " +
rt.getMessage());
    }
}

} else {
    logger.info("The Key Management resources will not be deleted");
}

logger.info(DASHES);
logger.info("This concludes the AWS Key Management SDK scenario");
logger.info(DASHES);
}

// This method is invoked from Exceptions to clean up the resources.
private static void deleteKey(String targetKeyId) {
    try {
        CompletableFuture<Void> future =
kmsActions.disableKeyAsync(targetKeyId);
        future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
    }

    try {
        CompletableFuture<Void> future =
kmsActions.deleteKeyAsync(targetKeyId);
        future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());

```

```
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
    }
}

// This method is invoked from Exceptions to clean up the resources.
private static void deleteAliasName(String aliasName) {
    try {
        CompletableFuture<Void> future =
kmsActions.deleteSpecificAliasAsync(aliasName);
        future.join();

    } catch (RuntimeException rt) {
        Throwable cause = rt.getCause();
        if (cause instanceof KmsException kmsEx) {
            logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
        } else {
            logger.info("An unexpected error occurred: " + rt.getMessage());
        }
    }
}

private static void waitForInputToContinue(Scanner scanner) {
    while (true) {
        logger.info("");
        logger.info("Enter 'c' followed by <ENTER> to continue:");
        String input = scanner.nextLine();

        if (input.trim().equalsIgnoreCase("c")) {
            logger.info("Continuing with the program...");
            logger.info("");
            break;
        } else {
            // Handle invalid input.
            logger.info("Invalid input. Please try again.");
        }
    }
}
}
```

Definisci una classe che racchiuda le azioni KMS.

```
public class KMSActions {
    private static final Logger logger =
    LoggerFactory.getLogger(KMSActions.class);
    private static KmsAsyncClient kmsAsyncClient;

    /**
     * Retrieves an asynchronous AWS Key Management Service (KMS) client.
     * <p>
     * This method creates and returns a singleton instance of the KMS async
     client, with the following configurations:
     * <ul>
     * <li>Max concurrency: 100</li>
     * <li>Connection timeout: 60 seconds</li>
     * <li>Read timeout: 60 seconds</li>
     * <li>Write timeout: 60 seconds</li>
     * <li>API call timeout: 2 minutes</li>
     * <li>API call attempt timeout: 90 seconds</li>
     * <li>Retry policy: up to 3 retries</li>
     * <li>Credentials provider: environment variable credentials provider</li>
     * </ul>
     * <p>
     * If the client instance has already been created, it is returned instead of
     creating a new one.
     *
     * @return the KMS async client instance
     */
    private static KmsAsyncClient getAsyncClient() {
        if (kmsAsyncClient == null) {
            SdkAsyncHttpClient httpClient = NettyNioAsyncHttpClient.builder()
                .maxConcurrency(100)
                .connectionTimeout(Duration.ofSeconds(60))
                .readTimeout(Duration.ofSeconds(60))
                .writeTimeout(Duration.ofSeconds(60))
                .build();

            ClientOverrideConfiguration overrideConfig =
            ClientOverrideConfiguration.builder()
                .apiCallTimeout(Duration.ofMinutes(2))
                .apiCallAttemptTimeout(Duration.ofSeconds(90))
                .retryPolicy(RetryPolicy.builder()
                    .numRetries(3)
                    .build())
        }
    }
}
```

```

        .build();

        kmsAsyncClient = KmsAsyncClient.builder()
            .httpClient(httpClient)
            .overrideConfiguration(overrideConfig)
            .build();
    }
    return kmsAsyncClient;
}

/**
 * Creates a new symmetric encryption key asynchronously.
 *
 * @param keyDesc the description of the key to be created
 * @return a {@link CompletableFuture} that completes with the ID of the
newly created key
 * @throws RuntimeException if an error occurs while creating the key
 */
public CompletableFuture<String> createKeyAsync(String keyDesc) {
    CreateKeyRequest keyRequest = CreateKeyRequest.builder()
        .description(keyDesc)
        .keySpec(KeySpec.SYMMETRIC_DEFAULT)
        .keyUsage(KeyUsageType.ENCRYPT_DECRYPT)
        .build();

    return getAsyncClient().createKey(keyRequest)
        .thenApply(resp -> resp.keyMetadata().keyId())
        .exceptionally(ex -> {
            throw new RuntimeException("An error occurred while creating the
key: " + ex.getMessage(), ex);
        });
}

/**
 * Asynchronously checks if a specified key is enabled.
 *
 * @param keyId the ID of the key to check
 * @return a {@link CompletableFuture} that, when completed, indicates
whether the key is enabled or not
 *
 * @throws RuntimeException if an exception occurs while checking the key
state
 */
public CompletableFuture<Boolean> isKeyEnabledAsync(String keyId) {

```

```

        DescribeKeyRequest keyRequest = DescribeKeyRequest.builder()
            .keyId(keyId)
            .build();

        CompletableFuture<DescribeKeyResponse> responseFuture =
getAsyncClient().describeKey(keyRequest);
        return responseFuture.whenComplete((resp, ex) -> {
            if (resp != null) {
                KeyState keyState = resp.keyMetadata().keyState();
                if (keyState == KeyState.ENABLED) {
                    logger.info("The key is enabled.");
                } else {
                    logger.info("The key is not enabled. Key state: {}",
keyState);
                }
            } else {
                throw new RuntimeException(ex);
            }
        }).thenApply(resp -> resp.keyMetadata().keyState() == KeyState.ENABLED);
    }

    /**
     * Asynchronously enables the specified key.
     *
     * @param keyId the ID of the key to enable
     * @return a {@link CompletableFuture} that completes when the key has been
     * enabled
     */
    public CompletableFuture<Void> enableKeyAsync(String keyId) {
        EnableKeyRequest enableKeyRequest = EnableKeyRequest.builder()
            .keyId(keyId)
            .build();

        CompletableFuture<EnableKeyResponse> responseFuture =
getAsyncClient().enableKey(enableKeyRequest);
        responseFuture.whenComplete((response, exception) -> {
            if (exception == null) {
                logger.info("Key with ID [{}] has been enabled.", keyId);
            } else {
                if (exception instanceof KmsException kmsEx) {
                    throw new RuntimeException("KMS error occurred while enabling
key: " + kmsEx.getMessage(), kmsEx);
                } else {

```

```

        throw new RuntimeException("An unexpected error occurred
while enabling key: " + exception.getMessage(), exception);
    }
}

});

return responseFuture.thenApply(response -> null);
}

/**
 * Encrypts the given text asynchronously using the specified KMS client and
key ID.
 *
 * @param keyId the ID of the KMS key to use for encryption
 * @param text the text to encrypt
 * @return a CompletableFuture that completes with the encrypted data as an
SdkBytes object
 */
public CompletableFuture<SdkBytes> encryptDataAsync(String keyId, String
text) {
    SdkBytes myBytes = SdkBytes.fromUtf8String(text);
    EncryptRequest encryptRequest = EncryptRequest.builder()
        .keyId(keyId)
        .plaintext(myBytes)
        .build();

    CompletableFuture<EncryptResponse> responseFuture =
getAsyncClient().encrypt(encryptRequest).toCompletableFuture();
    return responseFuture.whenComplete((response, ex) -> {
        if (response != null) {
            String algorithm = response.encryptionAlgorithm().toString();
            logger.info("The string was encrypted with algorithm {}.\"",
algorithm);
        } else {
            throw new RuntimeException(ex);
        }
    }).thenApply(EncryptResponse::ciphertextBlob);
}

/**
 * Creates a custom alias for the specified target key asynchronously.
 *
 * @param targetKeyId the ID of the target key for the alias
 * @param aliasName the name of the alias to create

```



```

    * @return a {@link CompletableFuture} that completes when the alias creation
    operation is finished
    */
    public CompletableFuture<Void> createCustomAliasAsync(String targetKeyId,
String aliasName) {
        CreateAliasRequest aliasRequest = CreateAliasRequest.builder()
            .aliasName(aliasName)
            .targetKeyId(targetKeyId)
            .build();

        CompletableFuture<CreateAliasResponse> responseFuture =
getAsyncClient().createAlias(aliasRequest);
        responseFuture.whenComplete((response, exception) -> {
            if (exception == null) {
                logger.info("{} was successfully created.", aliasName);
            } else {
                if (exception instanceof ResourceExistsException) {
                    logger.info("Alias [{}] already exists. Moving on...",
aliasName);
                } else if (exception instanceof KmsException kmsEx) {
                    throw new RuntimeException("KMS error occurred while creating
alias: " + kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred
while creating alias: " + exception.getMessage(), exception);
                }
            }
        });

        return responseFuture.thenApply(response -> null);
    }

    /**
    * Asynchronously lists all the aliases in the current AWS account.
    *
    * @return a {@link CompletableFuture} that completes when the list of
    aliases has been processed
    */
    public CompletableFuture<Object> listAllAliasesAsync() {
        ListAliasesRequest aliasesRequest = ListAliasesRequest.builder()
            .limit(15)
            .build();

```

```

        ListAliasesPublisher paginator =
getAsyncClient().listAliasesPaginator(aliasesRequest);
        return paginator.subscribe(response -> {
            response.aliases().forEach(alias ->
                logger.info("The alias name is: " + alias.aliasName())
            );
        })
        .thenApply(v -> null)
        .exceptionally(ex -> {
            if (ex.getCause() instanceof KmsException) {
                KmsException e = (KmsException) ex.getCause();
                throw new RuntimeException("A KMS exception occurred: " +
e.getMessage());
            } else {
                throw new RuntimeException("An unexpected error occurred: " +
ex.getMessage());
            }
        });
    }

    /**
     * Enables key rotation asynchronously for the specified key ID.
     *
     * @param keyId the ID of the key for which to enable key rotation
     * @return a CompletableFuture that represents the asynchronous operation of
     enabling key rotation
     * @throws RuntimeException if there was an error enabling key rotation,
     either due to a KMS exception or an unexpected error
     */
    public CompletableFuture<EnableKeyRotationResponse>
enableKeyRotationAsync(String keyId) {
        EnableKeyRotationRequest enableKeyRotationRequest =
EnableKeyRotationRequest.builder()
            .keyId(keyId)
            .build();

        CompletableFuture<EnableKeyRotationResponse> responseFuture =
getAsyncClient().enableKeyRotation(enableKeyRotationRequest);
        responseFuture.whenComplete((response, exception) -> {
            if (exception == null) {
                logger.info("Key rotation has been enabled for key with id [{ }]",
keyId);
            } else {
                if (exception instanceof KmsException kmsEx) {

```

```

        throw new RuntimeException("Failed to enable key rotation: "
+ kmsEx.getMessage(), kmsEx);
    } else {
        throw new RuntimeException("An unexpected error occurred: " +
exception.getMessage(), exception);
    }
}

});

return responseFuture;
}

/**
 * Grants permissions to a specified principal on a customer master key (CMK)
asynchronously.
 *
 * @param keyId          The unique identifier for the customer master key
(CMK) that the grant applies to.
 * @param granteePrincipal The principal that is given permission to perform
the operations that the grant permits on the CMK.
 * @return A {@link CompletableFuture} that, when completed, contains the ID
of the created grant.
 * @throws RuntimeException If an error occurs during the grant creation
process.
 */
public CompletableFuture<String> grantKeyAsync(String keyId, String
granteePrincipal) {
    List<GrantOperation> grantPermissions = List.of(
        GrantOperation.ENCRYPT,
        GrantOperation.DECRYPT,
        GrantOperation.DESCRIBE_KEY
    );

    CreateGrantRequest grantRequest = CreateGrantRequest.builder()
        .keyId(keyId)
        .name("grant1")
        .granteePrincipal(granteePrincipal)
        .operations(grantPermissions)
        .build();

    CompletableFuture<CreateGrantResponse> responseFuture =
getAsyncClient().createGrant(grantRequest);
    responseFuture.whenComplete((response, ex) -> {
        if (ex == null) {

```

```

        logger.info("Grant created successfully with ID: " +
response.grantId());
    } else {
        if (ex instanceof KmsException kmsEx) {
            throw new RuntimeException("Failed to create grant: " +
kmsEx.getMessage(), kmsEx);
        } else {
            throw new RuntimeException("An unexpected error occurred: " +
ex.getMessage(), ex);
        }
    }
    });

    return responseFuture.thenApply(CreateGrantResponse::grantId);
}

/**
 * Asynchronously displays the grant IDs for the specified key ID.
 *
 * @param keyId the ID of the AWS KMS key for which to list the grants
 * @return a {@link CompletableFuture} that, when completed, will be null
if the operation succeeded, or will throw a {@link RuntimeException} if the
operation failed
 * @throws RuntimeException if there was an error listing the grants, either
due to an {@link KmsException} or an unexpected error
 */
public CompletableFuture<Object> displayGrantIdsAsync(String keyId) {
    ListGrantsRequest grantsRequest = ListGrantsRequest.builder()
        .keyId(keyId)
        .limit(15)
        .build();

    ListGrantsPublisher paginator =
getAsyncClient().listGrantsPaginator(grantsRequest);
    return paginator.subscribe(response -> {
        response.grants().forEach(grant -> {
            logger.info("The grant Id is: " + grant.grantId());
        });
    })
        .thenApply(v -> null)
        .exceptionally(ex -> {
            Throwable cause = ex.getCause();
            if (cause instanceof KmsException) {

```

```

        throw new RuntimeException("Failed to list grants: " +
cause.getMessage(), cause);
    } else {
        throw new RuntimeException("An unexpected error occurred: " +
cause.getMessage(), cause);
    }
});
}

/**
 * Revokes a grant for the specified AWS KMS key asynchronously.
 *
 * @param keyId    The ID or key ARN of the AWS KMS key.
 * @param grantId  The identifier of the grant to be revoked.
 * @return A {@link CompletableFuture} representing the asynchronous
operation of revoking the grant.
 *
 * The {@link CompletableFuture} will complete with a {@link
RevokeGrantResponse} object
 *
 * if the operation is successful, or with a {@code null} value if an
error occurs.
 */
public CompletableFuture<RevokeGrantResponse> revokeKeyGrantAsync(String
keyId, String grantId) {
    RevokeGrantRequest grantRequest = RevokeGrantRequest.builder()
        .keyId(keyId)
        .grantId(grantId)
        .build();

    CompletableFuture<RevokeGrantResponse> responseFuture =
getAsyncClient().revokeGrant(grantRequest);
    responseFuture.whenComplete((response, exception) -> {
        if (exception == null) {
            logger.info("Grant ID: [" + grantId + "] was successfully
revoked!");
        } else {
            if (exception instanceof KmsException kmsEx) {
                if (kmsEx.getMessage().contains("Grant does not exist")) {
                    logger.info("The grant ID '" + grantId + "' does not
exist. Moving on...");
                } else {
                    throw new RuntimeException("KMS error occurred: " +
kmsEx.getMessage(), kmsEx);
                }
            } else {

```

```

        throw new RuntimeException("An unexpected error occurred: " +
exception.getMessage(), exception);
    }
}
});

return responseFuture;
}

/**
 * Asynchronously decrypts the given encrypted data using the specified key
ID.
 *
 * @param encryptedData The encrypted data to be decrypted.
 * @param keyId The ID of the key to be used for decryption.
 * @return A CompletableFuture that, when completed, will contain the
decrypted data as a String.
 *
 * If an error occurs during the decryption process, the
CompletableFuture will complete
 *
 * exceptionally with the error, and the method will return an empty
String.
 */
public CompletableFuture<String> decryptDataAsync(SdkBytes encryptedData,
String keyId) {
    DecryptRequest decryptRequest = DecryptRequest.builder()
        .ciphertextBlob(encryptedData)
        .keyId(keyId)
        .build();

    CompletableFuture<DecryptResponse> responseFuture =
getAsyncClient().decrypt(decryptRequest);
    responseFuture.whenComplete((decryptResponse, exception) -> {
        if (exception == null) {
            logger.info("Data decrypted successfully for key ID: " + keyId);
        } else {
            if (exception instanceof KmsException kmsEx) {
                throw new RuntimeException("KMS error occurred while
decrypting data: " + kmsEx.getMessage(), kmsEx);
            } else {
                throw new RuntimeException("An unexpected error occurred
while decrypting data: " + exception.getMessage(), exception);
            }
        }
    })
}

```

```

    });

    return responseFuture.thenApply(decryptResponse ->
decryptResponse.plaintext().asString(StandardCharsets.UTF_8));
}

/**
 * Asynchronously replaces the policy for the specified KMS key.
 *
 * @param keyId      the ID of the KMS key to replace the policy for
 * @param policyName the name of the policy to be replaced
 * @param accountId  the AWS account ID to be used in the policy
 * @return a {@link CompletableFuture} that completes with a boolean
indicating
 *         whether the policy replacement was successful or not
 */
public CompletableFuture<Boolean> replacePolicyAsync(String keyId, String
policyName, String accountId) {
    String policy = ""
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {"AWS": "arn:aws:iam::%s:root"},
        "Action": "kms:*",
        "Resource": "*"
    }]
}
"".formatted(accountId);

    PutKeyPolicyRequest keyPolicyRequest = PutKeyPolicyRequest.builder()
        .keyId(keyId)
        .policyName(policyName)
        .policy(policy)
        .build();

    // First, get the current policy to check if it exists
    return getAsyncClient().getKeyPolicy(r ->
r.keyId(keyId).policyName(policyName))
        .thenCompose(response -> {
            logger.info("Current policy exists. Replacing it...");
            return getAsyncClient().putKeyPolicy(keyPolicyRequest);
        })
        .thenApply(putPolicyResponse -> {

```

```

        logger.info("The key policy has been replaced.");
        return true;
    })
    .exceptionally(throwable -> {
        if (throwable.getCause() instanceof LimitExceededException) {
            logger.error("Cannot replace policy, as only one policy is
allowed per key.");
            return false;
        }
        throw new RuntimeException("Error replacing policy", throwable);
    });
}

/**
 * Asynchronously retrieves the key policy for the specified key ID and
policy name.
 *
 * @param keyId      the ID of the AWS KMS key for which to retrieve the
policy
 * @param policyName the name of the key policy to retrieve
 * @return a {@link CompletableFuture} that, when completed, contains the key
policy as a {@link String}
 */
public CompletableFuture<String> getKeyPolicyAsync(String keyId, String
policyName) {
    GetKeyPolicyRequest policyRequest = GetKeyPolicyRequest.builder()
        .keyId(keyId)
        .policyName(policyName)
        .build();

    return getAsyncClient().getKeyPolicy(policyRequest)
        .thenApply(response -> {
            String policy = response.policy();
            logger.info("The response is: " + policy);
            return policy;
        })
        .exceptionally(ex -> {
            throw new RuntimeException("Failed to get key policy", ex);
        });
}

/**
 * Asynchronously signs and verifies data using AWS KMS.

```



```

*
* <p>The method performs the following steps:
* <ol>
*     <li>Creates an AWS KMS key with the specified key spec, key usage, and
origin.</li>
*     <li>Signs the provided message using the created KMS key and the
RSASSA-PSS-SHA-256 algorithm.</li>
*     <li>Verifies the signature of the message using the created KMS key
and the RSASSA-PSS-SHA-256 algorithm.</li>
* </ol>
*
* @return a {@link CompletableFuture} that completes with the result of the
signature verification,
*     {@code true} if the signature is valid, {@code false} otherwise.
* @throws KmsException if any error occurs during the KMS operations.
* @throws RuntimeException if an unexpected error occurs.
*/
public CompletableFuture<Boolean> signVerifyDataAsync() {
    String signMessage = "Here is the message that will be digitally signed";

    // Create an AWS KMS key used to digitally sign data.
    CreateKeyRequest createKeyRequest = CreateKeyRequest.builder()
        .keySpec(KeySpec.RSA_2048)
        .keyUsage(KeyUsageType.SIGN_VERIFY)
        .origin(OriginType.AWS_KMS)
        .build();

    return getAsyncClient().createKey(createKeyRequest)
        .thenCompose(createKeyResponse -> {
            String keyId = createKeyResponse.keyMetadata().keyId();

            SdkBytes messageBytes = SdkBytes.fromString(signMessage,
Charset.defaultCharset());
            SignRequest signRequest = SignRequest.builder()
                .keyId(keyId)
                .message(messageBytes)
                .signingAlgorithm(SigningAlgorithmSpec.RSASSA_PSS_SHA_256)
                .build();

            return getAsyncClient().sign(signRequest)
                .thenCompose(signResponse -> {
                    byte[] signedBytes =
signResponse.signature().asByteArray();

```

```

        VerifyRequest verifyRequest = VerifyRequest.builder()
            .keyId(keyId)

            .message(SdkBytes.fromByteArray(signMessage.getBytes(Charset.defaultCharset()))))

            .signature(SdkBytes.fromByteBuffer(ByteBuffer.wrap(signedBytes)))

            .signingAlgorithm(SigningAlgorithmSpec.RSASSA_PSS_SHA_256)
                .build();

        return getAsyncClient().verify(verifyRequest)
            .thenApply(verifyResponse -> {
                return (boolean) verifyResponse.signatureValid();
            });
    });
}

.throwable);
}

/**
 * Asynchronously tags a KMS key with a specific tag.
 *
 * @param keyId the ID of the KMS key to be tagged
 * @return a {@link CompletableFuture} that completes when the tagging
operation is finished
 */
public CompletableFuture<Void> tagKMSKeyAsync(String keyId) {
    Tag tag = Tag.builder()
        .tagKey("Environment")
        .tagValue("Production")
        .build();

    TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
        .keyId(keyId)
        .tags(tag)
        .build();

    return getAsyncClient().tagResource(tagResourceRequest)
        .thenRun(() -> {
            logger.info("{} key was tagged", keyId);
        })
}

```

```

        .exceptionally(throwable -> {
            throw new RuntimeException("Failed to tag the KMS key",
throwable);
        });
    }

    /**
     * Deletes a specific KMS alias asynchronously.
     *
     * @param aliasName the name of the alias to be deleted
     * @return a {@link CompletableFuture} representing the asynchronous
operation of deleting the specified alias
     */
    public CompletableFuture<Void> deleteSpecificAliasAsync(String aliasName) {
        DeleteAliasRequest deleteAliasRequest = DeleteAliasRequest.builder()
            .aliasName(aliasName)
            .build();

        return getAsyncClient().deleteAlias(deleteAliasRequest)
            .thenRun(() -> {
                logger.info("Alias {} has been deleted successfully", aliasName);
            })
            .exceptionally(throwable -> {
                throw new RuntimeException("Failed to delete alias: " +
aliasName, throwable);
            });
    }

    /**
     * Asynchronously disables the specified AWS Key Management Service (KMS)
key.
     *
     * @param keyId the ID or Amazon Resource Name (ARN) of the KMS key to be
disabled
     * @return a CompletableFuture that, when completed, indicates that the key
has been disabled successfully
     */
    public CompletableFuture<Void> disableKeyAsync(String keyId) {
        DisableKeyRequest keyRequest = DisableKeyRequest.builder()
            .keyId(keyId)
            .build();

        return getAsyncClient().disableKey(keyRequest)
            .thenRun(() -> {

```

```

        logger.info("Key {} has been disabled successfully",keyId);
    })
    .exceptionally(throwable -> {
        throw new RuntimeException("Failed to disable key: " + keyId,
throwable);
    });
}

/**
 * Deletes a KMS key asynchronously.
 *
 * <p><strong>Warning:</strong> Deleting a KMS key is a destructive and
potentially dangerous operation.
 * When a KMS key is deleted, all data that was encrypted under the KMS key
becomes unrecoverable.
 * This means that any files, databases, or other data that were encrypted
using the deleted KMS key
 * will become permanently inaccessible. Exercise extreme caution when
deleting KMS keys.</p>
 *
 * @param keyId the ID of the KMS key to delete
 * @return a {@link CompletableFuture} that completes when the key deletion
is scheduled
 */
public CompletableFuture<Void> deleteKeyAsync(String keyId) {
    ScheduleKeyDeletionRequest deletionRequest =
ScheduleKeyDeletionRequest.builder()
        .keyId(keyId)
        .pendingWindowInDays(7)
        .build();

    return getAsyncClient().scheduleKeyDeletion(deletionRequest)
        .thenRun(() -> {
            logger.info("Key {} will be deleted in 7 days", keyId);
        })
        .exceptionally(throwable -> {
            throw new RuntimeException("Failed to schedule key deletion for
key ID: " + keyId, throwable);
        });
}

public String getAccountId(){
    try (StsClient stsClient = StsClient.create()){

```

```
        GetCallerIdentityResponse callerIdentity =
            stsClient.getCallerIdentity();
        return callerIdentity.account();
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Java 2.x .
 - [CreateAlias](#)
 - [CreateGrant](#)
 - [CreateKey](#)
 - [Decrypt](#)
 - [DescribeKey](#)
 - [DisableKey](#)
 - [EnableKey](#)
 - [Encrypt](#)
 - [GetKeyPolicy](#)
 - [ListAliases](#)
 - [ListGrants](#)
 - [ListKeys](#)
 - [RevokeGrant](#)
 - [ScheduleKeyDeletion](#)
 - [Sign](#)
 - [TagResource](#)

PHP

SDK per PHP

 Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
echo "\n";
echo "-----\n";
echo <<<WELCOME
Welcome to the AWS Key Management Service SDK Basics scenario.

This program demonstrates how to interact with AWS Key Management Service using
the AWS SDK for PHP (v3).
The AWS Key Management Service (KMS) is a secure and highly available service
that allows you to create
and manage AWS KMS keys and control their use across a wide range of AWS services
and applications.
KMS provides a centralized and unified approach to managing encryption keys,
making it easier to meet your
data protection and regulatory compliance requirements.

This KMS Basics scenario creates two key types:
- A symmetric encryption key is used to encrypt and decrypt data.
- An asymmetric key used to digitally sign data.

Let's get started...\n
WELCOME;
echo "-----\n";
$this->pressEnter();

$this->kmsClient = new KmsClient([]);
// Initialize the KmsService class with the client. This allows you to
override any defaults in the client before giving it to the service class.
$this->kmsService = new KmsService($this->kmsClient);

// 1. Create a symmetric KMS key.
echo "\n";
```

```
    echo "1. Create a symmetric KMS key.\n";
    echo "First, we will create a symmetric KMS key that is used to encrypt
and decrypt data by invoking createKey().\n";
    $this->pressEnter();

    $key = $this->kmsService->createKey();
    $this->resources['symmetricKey'] = $key['KeyId'];
    echo "Created a customer key with ARN {$key['Arn']}.\n";
    $this->pressEnter();

    // 2. Enable a KMS key.
    echo "\n";
    echo "2. Enable a KMS key.\n";
    echo "By default when you create an AWS key, it is enabled. The code
checks to
determine if the key is enabled. If it is not enabled, the code enables it.\n";
    $this->pressEnter();

    $keyInfo = $this->kmsService->describeKey($key['KeyId']);
    if(!$keyInfo['Enabled']){
        echo "The key was not enabled, so we will enable it.\n";
        $this->pressEnter();
        $this->kmsService->enableKey($key['KeyId']);
        echo "The key was successfully enabled.\n";
    }else{
        echo "The key was already enabled, so there was no need to enable it.
\n";
    }
    $this->pressEnter();

    // 3. Encrypt data using the symmetric KMS key.
    echo "\n";
    echo "3. Encrypt data using the symmetric KMS key.\n";
    echo "One of the main uses of symmetric keys is to encrypt and decrypt
data.\n";
    echo "Next, we'll encrypt the string 'Hello, AWS KMS!' with the
SYMMETRIC_DEFAULT encryption algorithm.\n";
    $this->pressEnter();
    $text = "Hello, AWS KMS!";
    $encryption = $this->kmsService->encrypt($key['KeyId'], $text);
    echo "The plaintext data was successfully encrypted with the algorithm:
{$encryption['EncryptionAlgorithm']}.\n";
    $this->pressEnter();
```

```
// 4. Create an alias.
echo "\n";
echo "4. Create an alias.\n";
$aliasInput = testable_readline("Please enter an alias prefixed with
\"alias/\" or press enter to use a default value: ");
if($aliasInput == ""){
    $aliasInput = "alias/dev-encryption-key";
}
$this->kmsService->createAlias($key['KeyId'], $aliasInput);
$this->resources['alias'] = $aliasInput;
echo "The alias \"$aliasInput\" was successfully created.\n";
$this->pressEnter();

// 5. List all of your aliases.
$aliasPageSize = 10;
echo "\n";
echo "5. List all of your aliases, up to $aliasPageSize.\n";
$this->pressEnter();
$aliasPaginator = $this->kmsService->listAliases();
foreach($aliasPaginator as $pages){
    foreach($pages['Aliases'] as $alias){
        echo $alias['AliasName'] . "\n";
    }
    break;
}
$this->pressEnter();

// 6. Enable automatic rotation of the KMS key.
echo "\n";
echo "6. Enable automatic rotation of the KMS key.\n";
echo "By default, when the SDK enables automatic rotation of a KMS key,
KMS rotates the key material of the KMS key one year (approximately 365 days)
from the enable date and every year
thereafter.";
$this->pressEnter();
$this->kmsService->enableKeyRotation($key['KeyId']);
echo "The key's rotation was successfully set for key:
{$key['KeyId']}\n";
$this->pressEnter();

// 7. Create a grant.
echo "7. Create a grant.\n";
echo "\n";
```


echo "A grant is a policy instrument that allows Amazon Web Services principals to use KMS keys.

It also can allow them to view a KMS key (DescribeKey) and create and manage grants.

When authorizing access to a KMS key, grants are considered along with key policies and IAM policies.\n";

```
$granteeARN = testable_readline("Please enter the Amazon Resource Name (ARN) of an Amazon Web Services principal. Valid principals include Amazon Web Services accounts, IAM users, IAM roles, federated users, and assumed role users. For help with the ARN syntax for a principal, see IAM ARNs in the Identity and Access Management User Guide. \nTo skip this step, press enter without any other values: ");
```

```
if($granteeARN){
    $operations = [
        "ENCRYPT",
        "DECRYPT",
        "DESCRIBE_KEY",
    ];
    $grant = $this->kmsService->createGrant($key['KeyId'], $granteeARN,
$operations);
```

```
    echo "The grant Id is: {$grant['GrantId']}\n";
```

```
}else{
```

```
    echo "Steps 7, 8, and 9 will be skipped.\n";
```

```
}
```

```
$this->pressEnter();
```

```
// 8. List grants for the KMS key.
```

```
if($granteeARN){
```

```
    echo "8. List grants for the KMS key.\n\n";
```

```
    $grantsPaginator = $this->kmsService->listGrants($key['KeyId']);
```

```
    foreach($grantsPaginator as $page){
```

```
        foreach($page['Grants'] as $grant){
```

```
            echo $grant['GrantId'] . "\n";
```

```
        }
```

```
    }
```

```
}else{
```

```
    echo "Skipping step 8...\n";
```

```
}
```

```
$this->pressEnter();
```

```
// 9. Revoke the grant.
```

```
if($granteeARN) {
```

```
    echo "\n";
```

```
    echo "9. Revoke the grant.\n";
```

```

        $this->pressEnter();
        $this->kmsService->revokeGrant($grant['GrantId'], $keyInfo['KeyId']);
        echo "{$grant['GrantId']} was successfully revoked!\n";
    }else{
        echo "Skipping step 9...\n";
    }
    $this->pressEnter();

    // 10. Decrypt the data.
    echo "\n";
    echo "10. Decrypt the data.\n";
    echo "Let's decrypt the data that was encrypted before.\n";
    echo "We'll use the same key to decrypt the string that we encrypted
earlier in the program.\n";
    $this->pressEnter();
    $decryption = $this->kmsService->decrypt($keyInfo['KeyId'],
$encryption['CiphertextBlob'], $encryption['EncryptionAlgorithm']);
    echo "The decrypted text is: {$decryption['Plaintext']}\n";
    $this->pressEnter();

    // 11. Replace a Key Policy.
    echo "\n";
    echo "11. Replace a Key Policy.\n";
    echo "A key policy is a resource policy for a KMS key. Key policies are
the primary way to control access to KMS keys.\n";
    echo "Every KMS key must have exactly one key policy. The statements in
the key policy determine who has permission to use the KMS key and how they can
use it.\n";
    echo " You can also use IAM policies and grants to control access to the
KMS key, but every KMS key must have a key policy.\n";
    echo "We will replace the key's policy with a new one:\n";
    $stsClient = new StsClient([]);
    $result = $stsClient->getCallerIdentity();
    $accountId = $result['Account'];
    $keyPolicy = <<< KEYPOLICY
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {"AWS": "arn:aws:iam::$accountId:root"},
        "Action": "kms:*",
        "Resource": "*"
    }]
}

```

```
KEYPOLICY;
    echo $keyPolicy;
    $this->pressEnter();
    $this->kmsService->putKeyPolicy($keyInfo['KeyId'], $keyPolicy);
    echo "The Key Policy was successfully replaced!\n";
    $this->pressEnter();

    // 12. Retrieve the key policy.
    echo "\n";
    echo "12. Retrieve the key policy.\n";
    echo "Let's get some information about the new policy and print it to the
screen.\n";
    $this->pressEnter();
    $policyInfo = $this->kmsService->getKeyPolicy($keyInfo['KeyId']);
    echo "We got the info! Here is the policy: \n";
    echo $policyInfo['Policy'] . "\n";
    $this->pressEnter();

    // 13. Create an asymmetric KMS key and sign data.
    echo "\n";
    echo "13. Create an asymmetric KMS key and sign data.\n";
    echo "Signing your data with an AWS key can provide several benefits that
make it an attractive option for your data signing needs.\n";
    echo "By using an AWS KMS key, you can leverage the security controls and
compliance features provided by AWS, which can help you meet various regulatory
requirements and enhance the overall security posture of your organization.\n";
    echo "First we'll create the asymmetric key.\n";
    $this->pressEnter();
    $keySpec = "RSA_2048";
    $keyUsage = "SIGN_VERIFY";
    $asymmetricKey = $this->kmsService->createKey($keySpec, $keyUsage);
    $this->resources['asymmetricKey'] = $asymmetricKey['KeyId'];
    echo "Created the key with ID: {$asymmetricKey['KeyId']}\n";
    echo "Next, we'll sign the data.\n";
    $this->pressEnter();
    $algorithm = "RSASSA_PSS_SHA_256";
    $sign = $this->kmsService->sign($asymmetricKey['KeyId'], $text,
$algorithm);
    $verify = $this->kmsService->verify($asymmetricKey['KeyId'], $text,
$sign['Signature'], $algorithm);
    echo "Signature verification result: {$sign['signature']}\n";
    $this->pressEnter();

    // 14. Tag the symmetric KMS key.
```

```

    echo "\n";
    echo "14. Tag the symmetric KMS key.\n";
    echo "By using tags, you can improve the overall management, security,
and governance of your KMS keys, making it easier to organize, track, and
control access to your encrypted data within your AWS environment.\n";
    echo "Let's tag our symmetric key as Environment->Production\n";
    $this->pressEnter();
    $this->kmsService->tagResource($key['KeyId'], [
        [
            'TagKey' => "Environment",
            'TagValue' => "Production",
        ],
    ]);
    echo "The key was successfully tagged!\n";
    $this->pressEnter();

    // 15. Schedule the deletion of the KMS key
    echo "\n";
    echo "15. Schedule the deletion of the KMS key.\n";
    echo "By default, KMS applies a waiting period of 30 days, but you can
specify a waiting period of 7-30 days.\n";
    echo "When this operation is successful, the key state of the KMS key
changes to PendingDeletion and the key can't be used in any cryptographic
operations.\n";
    echo "It remains in this state for the duration of the waiting period.\n
\n";

    echo "Deleting a KMS key is a destructive and potentially dangerous
operation. When a KMS key is deleted, all data that was encrypted under the KMS
key is unrecoverable.\n\n";

    $cleanUp = testable_readline("Would you like to delete the resources
created during this scenario, including the keys? (y/n): ");
    if($cleanUp == "Y" || $cleanUp == "y"){
        $this->cleanUp();
    }

    echo
    "-----
\n";
    echo "This concludes the AWS Key Management SDK Basics scenario\n";
    echo
    "-----
\n";

```

```

namespace Kms;

use Aws\Kms\Exception\KmsException;
use Aws\Kms\KmsClient;
use Aws\Result;
use Aws\ResultPaginator;
use AwsUtilities\AWSServiceClass;

class KmsService extends AWSServiceClass
{
    protected KmsClient $client;
    protected bool $verbose;

    /**
     * @param KmsClient|null $client
     * @param bool $verbose
     */
    public function __construct(KmsClient $client = null, bool $verbose = false)
    {
        $this->verbose = $verbose;
        if($client){
            $this->client = $client;
            return;
        }
        $this->client = new KmsClient([]);
    }

    /**
     * @param string $keySpec
     * @param string $keyUsage
     * @param string $description
     * @return array
     */
    public function createKey(string $keySpec = "", string $keyUsage = "", string
    $description = "Created by the SDK for PHP")
    {
        $parameters = ['Description' => $description];
        if($keySpec && $keyUsage){
            $parameters['KeySpec'] = $keySpec;

```

```

        $parameters['KeyUsage'] = $keyUsage;
    }
    try {
        $result = $this->client->createKey($parameters);
        return $result['KeyMetadata'];
    } catch (KmsException $caught){
        // Check for error specific to createKey operations
        if ($caught->getAwsErrorMessage() == "LimitExceededException"){
            echo "The request was rejected because a quota was exceeded. For
more information, see Quotas in the Key Management Service Developer Guide.";
        }
        throw $caught;
    }
}

/**
 * @param string $keyId
 * @param string $ciphertext
 * @param string $algorithm
 * @return Result
 */
public function decrypt(string $keyId, string $ciphertext, string $algorithm
= "SYMMETRIC_DEFAULT")
{
    try{
        return $this->client->decrypt([
            'CiphertextBlob' => $ciphertext,
            'EncryptionAlgorithm' => $algorithm,
            'KeyId' => $keyId,
        ]);
    } catch (KmsException $caught){
        echo "There was a problem decrypting the data: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}

/**
 * @param string $keyId
 * @param string $text

```

```

    * @return Result
    */
    public function encrypt(string $keyId, string $text)
    {
        try {
            return $this->client->encrypt([
                'KeyId' => $keyId,
                'Plaintext' => $text,
            ]);
        } catch (KmsException $caught){
            if($caught->getAwsErrorMessage() == "DisabledException"){
                echo "The request was rejected because the specified KMS key is
not enabled.\n";
            }
            throw $caught;
        }
    }

    /**
     * @param string $keyId
     * @param int $limit
     * @return ResultPaginator
     */
    public function listAliases(string $keyId = "", int $limit = 0)
    {
        $args = [];
        if($keyId){
            $args['KeyId'] = $keyId;
        }
        if($limit){
            $args['Limit'] = $limit;
        }
        try{
            return $this->client->getPaginator("ListAliases", $args);
        } catch (KmsException $caught){
            if($caught->getAwsErrorMessage() == "InvalidMarkerException"){
                echo "The request was rejected because the marker that specifies
where pagination should next begin is not valid.\n";
            }
            throw $caught;
        }
    }
}

```

```

    /**
     * @param string $keyId
     * @param string $alias
     * @return void
     */
    public function createAlias(string $keyId, string $alias)
    {
        try{
            $this->client->createAlias([
                'TargetKeyId' => $keyId,
                'AliasName' => $alias,
            ]);
        }catch (KmsException $caught){
            if($caught->getAwsErrorMessage() == "InvalidAliasNameException"){
                echo "The request was rejected because the specified alias name
is not valid.";
            }
            throw $caught;
        }
    }
}

    /**
     * @param string $keyId
     * @param string $granteePrincipal
     * @param array $operations
     * @param array $grantTokens
     * @return Result
     */
    public function createGrant(string $keyId, string $granteePrincipal, array
$operations, array $grantTokens = [])
    {
        $args = [
            'KeyId' => $keyId,
            'GranteePrincipal' => $granteePrincipal,
            'Operations' => $operations,
        ];
        if($grantTokens){
            $args['GrantTokens'] = $grantTokens;
        }
    }
}

```



```

        try{
            return $this->client->createGrant($args);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "InvalidGrantTokenException"){
                echo "The request was rejected because the specified grant token
is not valid.\n";
            }
            throw $caught;
        }
    }

    /**
     * @param string $keyId
     * @return array
     */
    public function describeKey(string $keyId)
    {
        try {
            $result = $this->client->describeKey([
                "KeyId" => $keyId,
            ]);
            return $result['KeyMetadata'];
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "The request was rejected because the specified entity or
resource could not be found.\n";
            }
            throw $caught;
        }
    }

    /**
     * @param string $keyId
     * @return void
     */
    public function disableKey(string $keyId)
    {
        try {
            $this->client->disableKey([
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "The request was rejected because the specified entity or
resource could not be found.\n";
            }
            throw $caught;
        }
    }

```

```

    });
    }catch(KmsException $caught){
        echo "There was a problem disabling the key: {"$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}

/**
 * @param string $keyId
 * @return void
 */
public function enableKey(string $keyId)
{
    try {
        $this->client->enableKey([
            'KeyId' => $keyId,
        ]);
    }catch(KmsException $caught){
        if($caught->getAwsErrorMessage() == "NotFoundException"){
            echo "The request was rejected because the specified entity or
resource could not be found.\n";
        }
        throw $caught;
    }
}

/**
 * @return array
 */
public function listKeys()
{
    try {
        $contents = [];
        $paginator = $this->client->getPaginator("ListKeys");
        foreach($paginator as $result){
            foreach ($result['Content'] as $object) {
                $contents[] = $object;
            }
        }
    }
}

```

```

        return $contents;
    }catch(KmsException $caught){
        echo "There was a problem listing the keys: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}

/**
 * @param string $keyId
 * @return Result
 */
public function listGrants(string $keyId)
{
    try{
        return $this->client->listGrants([
            'KeyId' => $keyId,
        ]);
    }catch(KmsException $caught){
        if($caught->getAwsErrorMessage() == "NotFoundException"){
            echo "    The request was rejected because the specified entity
or resource could not be found.\n";
        }
        throw $caught;
    }
}

/**
 * @param string $keyId
 * @return Result
 */
public function getKeyPolicy(string $keyId)
{
    try {
        return $this->client->getKeyPolicy([
            'KeyId' => $keyId,
        ]);
    }catch(KmsException $caught){
        echo "There was a problem getting the key policy: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}

```

```

    }
}

/**
 * @param string $grantId
 * @param string $keyId
 * @return void
 */
public function revokeGrant(string $grantId, string $keyId)
{
    try{
        $this->client->revokeGrant([
            'GrantId' => $grantId,
            'KeyId' => $keyId,
        ]);
    }catch(KmsException $caught){
        echo "There was a problem with revoking the grant: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}

/**
 * @param string $keyId
 * @param int $pendingWindowInDays
 * @return void
 */
public function scheduleKeyDeletion(string $keyId, int $pendingWindowInDays =
7)
{
    try {
        $this->client->scheduleKeyDeletion([
            'KeyId' => $keyId,
            'PendingWindowInDays' => $pendingWindowInDays,
        ]);
    }catch(KmsException $caught){
        echo "There was a problem scheduling the key deletion: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}

```

```
/**
 * @param string $keyId
 * @param array $tags
 * @return void
 */
public function tagResource(string $keyId, array $tags)
{
    try {
        $this->client->tagResource([
            'KeyId' => $keyId,
            'Tags' => $tags,
        ]);
    } catch (KmsException $caught){
        echo "There was a problem applying the tag(s): {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}

/**
 * @param string $keyId
 * @param string $message
 * @param string $algorithm
 * @return Result
 */
public function sign(string $keyId, string $message, string $algorithm)
{
    try {
        return $this->client->sign([
            'KeyId' => $keyId,
            'Message' => $message,
            'SigningAlgorithm' => $algorithm,
        ]);
    } catch (KmsException $caught){
        echo "There was a problem signing the data: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}
```

```
/**
 * @param string $keyId
 * @param int $rotationPeriodInDays
 * @return void
 */
public function enableKeyRotation(string $keyId, int $rotationPeriodInDays =
365)
{
    try{
        $this->client->enableKeyRotation([
            'KeyId' => $keyId,
            'RotationPeriodInDays' => $rotationPeriodInDays,
        ]);
    }catch(KmsException $caught){
        if($caught->getAwsErrorMessage() == "NotFoundException"){
            echo "The request was rejected because the specified entity or
resource could not be found.\n";
        }
        throw $caught;
    }
}

/**
 * @param string $keyId
 * @param string $policy
 * @return void
 */
public function putKeyPolicy(string $keyId, string $policy)
{
    try {
        $this->client->putKeyPolicy([
            'KeyId' => $keyId,
            'Policy' => $policy,
        ]);
    }catch(KmsException $caught){
        echo "There was a problem replacing the key policy: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}
```

```
}

/**
 * @param string $aliasName
 * @return void
 */
public function deleteAlias(string $aliasName)
{
    try {
        $this->client->deleteAlias([
            'AliasName' => $aliasName,
        ]);
    } catch (KmsException $caught){
        echo "There was a problem deleting the alias: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}

/**
 * @param string $keyId
 * @param string $message
 * @param string $signature
 * @param string $signingAlgorithm
 * @return bool
 */
public function verify(string $keyId, string $message, string $signature,
string $signingAlgorithm)
{
    try {
        $result = $this->client->verify([
            'KeyId' => $keyId,
            'Message' => $message,
            'Signature' => $signature,
            'SigningAlgorithm' => $signingAlgorithm,
        ]);
        return $result['SignatureValid'];
    } catch (KmsException $caught){
        echo "There was a problem verifying the signature: {$caught-
>getAwsErrorMessage()}\n";
    }
}
```

```
        throw $caught;
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK per PHP .

- [CreateAlias](#)
- [CreateGrant](#)
- [CreateKey](#)
- [Decrypt](#)
- [DescribeKey](#)
- [DisableKey](#)
- [EnableKey](#)
- [Encrypt](#)
- [GetKeyPolicy](#)
- [ListAliases](#)
- [ListGrants](#)
- [ListKeys](#)
- [RevokeGrant](#)
- [ScheduleKeyDeletion](#)
- [Sign](#)
- [TagResource](#)

Python

SDK per Python (Boto3)

Note

C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
class KMSScenario:
    """Runs an interactive scenario that shows how to get started with KMS."""

    def __init__(
        self,
        key_manager: KeyManager,
        key_encryption: KeyEncrypt,
        alias_manager: AliasManager,
        grant_manager: GrantManager,
        key_policy: KeyPolicy,
    ):
        self.key_manager = key_manager
        self.key_encryption = key_encryption
        self.alias_manager = alias_manager
        self.grant_manager = grant_manager
        self.key_policy = key_policy
        self.key_id = ""
        self.alias_name = ""
        self.asymmetric_key_id = ""

    def kms_scenario(self):
        key_description = "Created by the AWS KMS API"

        print(DASHES)
        print(
            """
Welcome to the AWS Key Management SDK Basics scenario.

This program demonstrates how to interact with AWS Key Management using the AWS
SDK for Python (Boto3).
The AWS Key Management Service (KMS) is a secure and highly available service
that allows you to create
```

and manage AWS KMS keys and control their use across a wide range of AWS services and applications.

KMS provides a centralized and unified approach to managing encryption keys, making it easier to meet your data protection and regulatory compliance requirements.

This Basics scenario creates two key types:

- A symmetric encryption key is used to encrypt and decrypt data.
- An asymmetric key used to digitally sign data.

Let's get started...

```

        """
    )
    q.ask("Press Enter to continue...")

    print(DASHES)
    print(f"1. Create a symmetric KMS key\n")
    print(
        f"First, the program will creates a symmetric KMS key that you can
        used to encrypt and decrypt data."
    )
    q.ask("Press Enter to continue...")
    self.key_id = self.key_manager.create_key(key_description)["KeyId"]
    print(f"A symmetric key was successfully created {self.key_id}.")
    q.ask("Press Enter to continue...")
    print(DASHES)
    print(
        """

```

2. Enable a KMS key

By default, when the SDK creates an AWS key, it is enabled. The next bit of code checks to

determine if the key is enabled.

```

        """
    )
    q.ask("Press Enter to continue...")
    is_enabled = self.is_key_enabled(self.key_id)
    print(f"Is the key enabled? {is_enabled}")
    if not is_enabled:
        self.key_manager.enable_key(self.key_id)
    q.ask("Press Enter to continue...")
    print(DASHES)
    print(f"3. Encrypt data using the symmetric KMS key")

```

```

    plain_text = "Hello, AWS KMS!"
    print(
        f"""

```

One of the main uses of symmetric keys is to encrypt and decrypt data.

Next, the code encrypts the string "{plain_text}" with the SYMMETRIC_DEFAULT encryption algorithm.

```

        """
    )
    q.ask("Press Enter to continue...")
    encrypted_text = self.key_encryption.encrypt(self.key_id, plain_text)
    print(DASHES)
    print(f"4. Create an alias")
    print(
        """

```

Now, the program will create an alias for the KMS key. An alias is a friendly name that you

can associate with a KMS key. The alias name should be prefixed with 'alias/'.

```

        """
    )
    alias_name = q.ask("Enter an alias name: ", q.non_empty)
    self.alias_manager.create_alias(self.key_id, alias_name)
    print(f"{alias_name} was successfully created.")
    self.alias_name = alias_name
    print(DASHES)
    print(f"5. List all of your aliases")
    q.ask("Press Enter to continue...")
    self.alias_manager.list_aliases(10)
    q.ask("Press Enter to continue...")
    print(DASHES)
    print(f"6. Enable automatic rotation of the KMS key")
    print(
        """

```

By default, when the SDK enables automatic rotation of a KMS key,

KMS rotates the key material of the KMS key one year (approximately 365 days) from the enable date and every year

thereafter.

```

        """
    )
    q.ask("Press Enter to continue...")
    self.key_manager.enable_key_rotation(self.key_id)
    print(DASHES)
    print(f"Key rotation has been enabled for key with id {self.key_id}")
    print(

```

```

    """
7. Create a grant

```

A grant is a policy instrument that allows Amazon Web Services principals to use KMS keys.

It also can allow them to view a KMS key (DescribeKey) and create and manage grants.

When authorizing access to a KMS key, grants are considered along with key policies and IAM policies.

```

    """
    )
    print(
        """

```

To create a grant you must specify a `account_id`. To specify the grantee `account_id`, use the Amazon Resource Name (ARN)

of an AWS `account_id`. Valid principals include AWS accounts, IAM users, IAM roles, federated users, and assumed role users.

```

    """
    )
    account_id = q.ask(
        "Enter an account_id, or press enter to skip creating a grant... "
    )
    grant = None
    if account_id != "":
        grant = self.grant_manager.create_grant(
            self.key_id,
            account_id,
            [
                "Encrypt",
                "Decrypt",
                "DescribeKey",
            ],
        )
        print(f"Grant created successfully with ID: {grant['GrantId']}")

    q.ask("Press Enter to continue...")
    print(DASHES)
    print(DASHES)
    print(f"8. List grants for the KMS key")
    q.ask("Press Enter to continue...")
    self.grant_manager.list_grants(self.key_id)
    q.ask("Press Enter to continue...")
    print(DASHES)

```

```

    print(f"9. Revoke the grant")
    print(
        """

```

The revocation of a grant immediately removes the permissions and access that the grant had provided.

This means that any `account_id` (user, role, or service) that was granted access to perform specific

KMS operations on a KMS key will no longer be able to perform those operations.

```

        """
    )
    q.ask("Press Enter to continue...")

    if grant is not None:
        self.grant_manager.revoke_grant(self.key_id, grant["GrantId"])
        print(f"Grant ID: {grant['GrantId']} was successfully revoked!")

```

```

    q.ask("Press Enter to continue...")
    print(DASHES)
    print(f"10. Decrypt the data\n")
    print(
        """

```

Lets decrypt the data that was encrypted in an early step.

The code uses the same key to decrypt the string that we encrypted earlier in the program.

```

        """
    )
    q.ask("Press Enter to continue...")
    decrypted_data = self.key_encryption.decrypt(self.key_id, encrypted_text)
    print(f"Data decrypted successfully for key ID: {self.key_id}")
    print(f"Decrypted data: {decrypted_data}")

```

```

    q.ask("Press Enter to continue...")
    print(DASHES)
    print(f"11. Replace a key policy\n")
    print(
        """

```

A key policy is a resource policy for a KMS key. Key policies are the primary way to control

access to KMS keys. Every KMS key must have exactly one key policy. The statements in the key policy

determine who has permission to use the KMS key and how they can use it.

You can also use IAM policies and grants to control access to the KMS key, but every KMS key

must have a key policy.

By default, when you create a key by using the SDK, a policy is created that gives the AWS account that owns the KMS key full access to the KMS key.

Let's try to replace the automatically created policy with the following policy.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {"AWS": "arn:aws:iam::000000000000:root"},
    "Action": "kms:*",
    "Resource": "*"
  }]
}
```

```

    """
    )
    account_id = q.ask("Enter your account ID or press enter to skip: ")
    if account_id != "":
        policy = {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {"AWS": f"arn:aws:iam::{account_id}:root"},
                    "Action": "kms:*",
                    "Resource": "*",
                }
            ],
        }

        self.key_policy.set_new_policy(self.key_id, policy)
        print("Key policy replacement succeeded.")
        q.ask("Press Enter to continue...")
    else:
        print("Skipping replacing the key policy.")

    print(DASHES)
    print(f"12. Get the key policy\n")
    print(
        f"The next bit of code that runs gets the key policy to make sure it
exists."
    )
    q.ask("Press Enter to continue...")
    policy = self.key_policy.get_policy(self.key_id)
```

```

print(f"The key policy is: {policy}")

q.ask("Press Enter to continue...")
print(DASHES)
print(f"13. Create an asymmetric KMS key and sign your data\n")
print(
    """
    Signing your data with an AWS key can provide several benefits that make
it an attractive option
    for your data signing needs. By using an AWS KMS key, you can leverage
the
    security controls and compliance features provided by AWS,
    which can help you meet various regulatory requirements and enhance the
overall security posture
    of your organization.
    """
)
q.ask("Press Enter to continue...")
print(f"Sign and verify data operation succeeded.")
self.asymmetric_key_id = self.key_manager.create_asymmetric_key()
message = "Here is the message that will be digitally signed"
signature = self.key_encryption.sign(self.asymmetric_key_id, message)
if self.key_encryption.verify(self.asymmetric_key_id, message,
signature):
    print("Signature verification succeeded.")
else:
    print("Signature verification failed.")

q.ask("Press Enter to continue...")
print(DASHES)
print(f"14. Tag your symmetric KMS Key\n")
print(
    """
    By using tags, you can improve the overall management, security, and
governance of your
    KMS keys, making it easier to organize, track, and control access to your
encrypted data within
    your AWS environment
    """
)
q.ask("Press Enter to continue...")
self.key_manager.tag_resource(self.key_id, "Environment", "Production")
self.clean_up()

```

```
def is_key_enabled(self, key_id: str) -> bool:
    """
    Check if the key is enabled or not.

    :param key_id: The key to check.
    :return: True if the key is enabled, otherwise False.
    """
    response = self.key_manager.describe_key(key_id)
    return response["Enabled"] is True
```

```
def clean_up(self):
    """
    Delete resources created by this scenario.
    """
    if self.alias_name != "":
        print(f"Deleting the alias {self.alias_name}.")
        self.alias_manager.delete_alias(self.alias_name)
    window = 7 # The window in days for a scheduled deletion.
    if self.key_id != "":
        print(
            """
```

Warning:

Deleting a KMS key is a destructive and potentially dangerous operation. When a KMS key is deleted, all data that was encrypted under the KMS key is unrecoverable.

```
        """
    )
    if q.ask(
        f"Do you want to delete the key with ID {self.key_id} (y/n)?",
        q.is_yesno,
    ):
        print(
            f"The key {self.key_id} will be deleted with a window of {window} days. You can cancel the deletion before"
        )
        print("the window expires.")
        self.key_manager.delete_key(self.key_id, window)
        self.key_id = ""

    if self.asymmetric_key_id != "":
        if q.ask(
            f"Do you want to delete the asymmetric key with ID {self.asymmetric_key_id} (y/n)?",
            q.is_yesno,
```



```

        ):
            print(
                f"The key {self.asymmetric_key_id} will be deleted with a
window of {window} days. You can cancel the deletion before"
            )
            print("the window expires.")
            self.key_manager.delete_key(self.asymmetric_key_id, window)
            self.asymmetric_key_id = ""

if __name__ == "__main__":
    kms_scenario = None
    try:
        kms_client = boto3.client("kms")
        a_key_manager = KeyManager(kms_client)
        a_key_encrypt = KeyEncrypt(kms_client)
        an_alias_manager = AliasManager(kms_client)
        a_grant_manager = GrantManager(kms_client)
        a_key_policy = KeyPolicy(kms_client)
        kms_scenario = KMSScenario(
            key_manager=a_key_manager,
            key_encryption=a_key_encrypt,
            alias_manager=an_alias_manager,
            grant_manager=a_grant_manager,
            key_policy=a_key_policy,
        )
        kms_scenario.kms_scenario()
    except Exception:
        logging.exception("Something went wrong with the demo!")
        if kms_scenario is not None:
            kms_scenario.clean_up()

```

Classe e metodi Wrapper per la gestione delle chiavi KMS.

```

class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":

```

```

    """
    Creates a KeyManager instance with a default KMS client.

    :return: An instance of KeyManager initialized with the default KMS
client.
    """
    kms_client = boto3.client("kms")
    return cls(kms_client)

def create_key(self, key_description: str) -> dict[str, any]:
    """
    Creates a key with a user-provided description.

    :param key_description: A description for the key.
    :return: The key ID.
    """
    try:
        key = self.kms_client.create_key(Description=key_description)
["KeyMetadata"]
        self.created_keys.append(key)
        return key
    except ClientError as err:
        logging.error(
            "Couldn't create your key. Here's why: %s",
            err.response["Error"]["Message"],
        )
        raise

def describe_key(self, key_id: str) -> dict[str, any]:
    """
    Describes a key.

    :param key_id: The ARN or ID of the key to describe.
    :return: Information about the key.
    """

    try:
        key = self.kms_client.describe_key(KeyId=key_id)["KeyMetadata"]
        return key
    except ClientError as err:
        logging.error(
            "Couldn't get key '%s'. Here's why: %s",

```

```

        key_id,
        err.response["Error"]["Message"],
    )
    raise

def enable_key_rotation(self, key_id: str) -> None:
    """
    Enables rotation for a key.

    :param key_id: The ARN or ID of the key to enable rotation for.
    """
    try:
        self.kms_client.enable_key_rotation(KeyId=key_id)
    except ClientError as err:
        logging.error(
            "Couldn't enable rotation for key '%s'. Here's why: %s",
            key_id,
            err.response["Error"]["Message"],
        )
        raise

def create_asymmetric_key(self) -> str:
    """
    Creates an asymmetric key in AWS KMS for signing messages.

    :return: The ID of the created key.
    """
    try:
        key = self.kms_client.create_key(
            KeySpec="RSA_2048", KeyUsage="SIGN_VERIFY", Origin="AWS_KMS"
        )["KeyMetadata"]
        self.created_keys.append(key)
        return key["KeyId"]
    except ClientError as err:
        logger.error(
            "Couldn't create your key. Here's why: %s",
            err.response["Error"]["Message"],
        )
        raise

def tag_resource(self, key_id: str, tag_key: str, tag_value: str) -> None:

```

```

"""
Add or edit tags on a customer managed key.

:param key_id: The ARN or ID of the key to enable rotation for.
:param tag_key: Key for the tag.
:param tag_value: Value for the tag.
"""
try:
    self.kms_client.tag_resource(
        KeyId=key_id, Tags=[{"TagKey": tag_key, "TagValue": tag_value}]
    )
except ClientError as err:
    logging.error(
        "Couldn't add a tag for the key '%s'. Here's why: %s",
        key_id,
        err.response["Error"]["Message"],
    )
    raise

def delete_key(self, key_id: str, window: int) -> None:
    """
    Deletes a list of keys.

    Warning:
    Deleting a KMS key is a destructive and potentially dangerous operation.
    When a KMS key is deleted,
    all data that was encrypted under the KMS key is unrecoverable.

    :param key_id: The ARN or ID of the key to delete.
    :param window: The waiting period, in days, before the KMS key is
    deleted.
    """
    try:
        self.kms_client.schedule_key_deletion(
            KeyId=key_id, PendingWindowInDays=window
        )
    except ClientError as err:
        logging.error(
            "Couldn't delete key %s. Here's why: %s",
            key_id,
            err.response["Error"]["Message"],
        )

```

```
raise
```

Classe e metodi Wrapper per gli alias delle chiavi KMS.

```
class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def create_alias(self, key_id: str, alias: str) -> None:
        """
        Creates an alias for the specified key.

        :param key_id: The ARN or ID of a key to give an alias.
        :param alias: The alias to assign to the key.
        """
        try:
            self.kms_client.create_alias(AliasName=alias, TargetKeyId=key_id)
        except ClientError as err:
            if err.response["Error"]["Code"] == "AlreadyExistsException":
                logger.error(
                    "Could not create the alias %s because it already exists.",
                    key_id
                )
            else:
                logger.error(
                    "Couldn't encrypt text. Here's why: %s",
                    err.response["Error"]["Message"],
                )
```

```

        raise

def list_aliases(self, page_size: int) -> None:
    """
    Lists aliases for the current account.
    :param page_size: The number of aliases to list per page.
    """
    try:
        alias_paginator = self.kms_client.get_paginator("list_aliases")
        for alias_page in alias_paginator.paginate(
            PaginationConfig={"PageSize": page_size}
        ):
            print(f"Here are {page_size} aliases:")
            pprint(alias_page["Aliases"])
            if alias_page["Truncated"]:
                answer = input(
                    f"Do you want to see the next {page_size} aliases (y/n)?
"
                    )
                if answer.lower() != "y":
                    break
            else:
                print("That's all your aliases!")
    except ClientError as err:
        logging.error(
            "Couldn't list your aliases. Here's why: %s",
            err.response["Error"]["Message"],
        )
        raise

def delete_alias(self, alias: str) -> None:
    """
    Deletes an alias.

    :param alias: The alias to delete.
    """
    try:
        self.kms_client.delete_alias(AliasName=alias)
    except ClientError as err:
        logger.error(
            "Couldn't delete alias %s. Here's why: %s",
            alias,

```

```

        err.response["Error"]["Message"],
    )
    raise

```

Classe e metodi Wrapper per la crittografia delle chiavi KMS.

```

class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def encrypt(self, key_id: str, text: str) -> bytes:
        """
        Encrypts text by using the specified key.

        :param key_id: The ARN or ID of the key to use for encryption.
        :param text: The text to encrypt.
        :return: The encrypted version of the text.
        """
        try:
            response = self.kms_client.encrypt(KeyId=key_id,
            Plaintext=text.encode())
            print(
                f"The string was encrypted with algorithm
            {response['EncryptionAlgorithm']}"
            )
            return response["CiphertextBlob"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DisabledException":

```

```

        logger.error(
            "Could not encrypt because the key %s is disabled.", key_id
        )
    else:
        logger.error(
            "Couldn't encrypt text. Here's why: %s",
            err.response["Error"]["Message"],
        )
    raise

def decrypt(self, key_id: str, cipher_text: bytes) -> str:
    """
    Decrypts text previously encrypted with a key.

    :param key_id: The ARN or ID of the key used to decrypt the data.
    :param cipher_text: The encrypted text to decrypt.
    :return: The decrypted text.
    """
    try:
        return self.kms_client.decrypt(KeyId=key_id,
            CiphertextBlob=cipher_text)[
            "Plaintext"
        ].decode()
    except ClientError as err:
        logger.error(
            "Couldn't decrypt your ciphertext. Here's why: %s",
            err.response["Error"]["Message"],
        )
    raise

def sign(self, key_id: str, message: str) -> str:
    """
    Signs a message with a key.

    :param key_id: The ARN or ID of the key to use for signing.
    :param message: The message to sign.
    :return: The signature of the message.
    """
    try:
        return self.kms_client.sign(
            KeyId=key_id,
            Message=message.encode(),

```



```

        SigningAlgorithm="RSASSA_PSS_SHA_256",
    )["Signature"]
except ClientError as err:
    logger.error(
        "Couldn't sign your message. Here's why: %s",
        err.response["Error"]["Message"],
    )
    raise

def verify(self, key_id: str, message: str, signature: str) -> bool:
    """
    Verifies a signature against a message.

    :param key_id: The ARN or ID of the key used to sign the message.
    :param message: The message to verify.
    :param signature: The signature to verify.
    :return: True when the signature matches the message, otherwise False.
    """
    try:
        response = self.kms_client.verify(
            KeyId=key_id,
            Message=message.encode(),
            Signature=signature,
            SigningAlgorithm="RSASSA_PSS_SHA_256",
        )
        valid = response["SignatureValid"]
        print(f"The signature is {'valid' if valid else 'invalid'}.")
        return valid
    except ClientError as err:
        if err.response["Error"]["Code"] == "SignatureDoesNotMatchException":
            print("The signature is not valid.")
        else:
            logger.error(
                "Couldn't verify your signature. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise

```

Classe e metodi Wrapper per la concessione di chiavi KMS.

```

class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def create_grant(
        self, key_id: str, principal: str, operations: [str]
    ) -> dict[str, str]:
        """
        Creates a grant for a key that lets a principal generate a symmetric data
        encryption key.

        :param key_id: The ARN or ID of the key.
        :param principal: The principal to grant permission to.
        :param operations: The operations to grant permission for.
        :return: The grant that is created.
        """
        try:
            return self.kms_client.create_grant(
                KeyId=key_id,
                GranteePrincipal=principal,
                Operations=operations,
            )
        except ClientError as err:
            logger.error(
                "Couldn't create a grant on key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise

```

```
def list_grants(self, key_id):
    """
    Lists grants for a key.

    :param key_id: The ARN or ID of the key to query.
    :return: The grants for the key.
    """
    try:
        paginator = self.kms_client.get_paginator("list_grants")
        grants = []
        page_iterator = paginator.paginate(KeyId=key_id)
        for page in page_iterator:
            grants.extend(page["Grants"])

        print(f"Grants for key {key_id}:")
        pprint(grants)
        return grants
    except ClientError as err:
        logger.error(
            "Couldn't list grants for key %s. Here's why: %s",
            key_id,
            err.response["Error"]["Message"],
        )
        raise

def revoke_grant(self, key_id: str, grant_id: str) -> None:
    """
    Revokes a grant so that it can no longer be used.

    :param key_id: The ARN or ID of the key associated with the grant.
    :param grant_id: The ID of the grant to revoke.
    """
    try:
        self.kms_client.revoke_grant(KeyId=key_id, GrantId=grant_id)
    except ClientError as err:
        logger.error(
            "Couldn't revoke grant %s. Here's why: %s",
            grant_id,
            err.response["Error"]["Message"],
        )
        raise
```

Classe e metodi Wrapper per le politiche chiave KMS.

```
class KeyPolicy:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyPolicy":
        """
        Creates a KeyPolicy instance with a default KMS client.

        :return: An instance of KeyPolicy initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def set_new_policy(self, key_id: str, policy: dict[str, any]) -> None:
        """
        Sets the policy of a key. Setting a policy entirely overwrites the
existing
        policy, so care is taken to add a statement to the existing list of
statements
        rather than simply writing a new policy.

        :param key_id: The ARN or ID of the key to set the policy to.
        :param policy: A new key policy. The key policy must allow the calling
principal to make a subsequent
            PutKeyPolicy request on the KMS key. This reduces the risk
that the KMS key becomes unmanageable
        """

        try:
            self.kms_client.put_key_policy(KeyId=key_id,
Policy=json.dumps(policy))
        except ClientError as err:
            logger.error(
                "Couldn't set policy for key %s. Here's why %s",
                key_id,
                err.response["Error"]["Message"],
```

```
    )
    raise

def get_policy(self, key_id: str) -> dict[str, str]:
    """
    Gets the policy of a key.

    :param key_id: The ARN or ID of the key to query.
    :return: The key policy as a dict.
    """
    if key_id != "":
        try:
            response = self.kms_client.get_key_policy(
                KeyId=key_id,
            )
            policy = json.loads(response["Policy"])
        except ClientError as err:
            logger.error(
                "Couldn't get policy for key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
        else:
            pprint(policy)
            return policy
    else:
        print("Skipping get policy demo.")
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Python (Boto3).
 - [CreateAlias](#)
 - [CreateGrant](#)
 - [CreateKey](#)
 - [Decrypt](#)
 - [DescribeKey](#)
 - [DisableKey](#)

- [EnableKey](#)
- [Encrypt](#)
- [GetKeyPolicy](#)
- [ListAliases](#)
- [ListGrants](#)
- [ListKeys](#)
- [RevokeGrant](#)
- [ScheduleKeyDeletion](#)
- [Sign](#)
- [TagResource](#)

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un AWS SDK](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Azioni per l'utilizzo AWS KMSAWS SDKs

I seguenti esempi di codice mostrano come eseguire singole AWS KMS azioni con AWS SDKs. Ogni esempio include un collegamento a GitHub, dove sono disponibili le istruzioni per la configurazione e l'esecuzione del codice.

Questi estratti richiamano l' AWS KMS API e sono estratti di codice di programmi più grandi che devono essere eseguiti nel contesto. È possibile visualizzare le azioni nel contesto in [Scenari di utilizzo AWS KMSAWS SDKs](#).

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta la [Documentazione di riferimento delle API AWS Key Management Service](#).

Esempi

- [Utilizzo CreateAlias con un AWS SDK o una CLI](#)
- [Utilizzo CreateGrant con un AWS SDK o una CLI](#)
- [Utilizzo CreateKey con un AWS SDK o una CLI](#)
- [Utilizzo Decrypt con un AWS SDK o una CLI](#)
- [Utilizzo DeleteAlias con un AWS SDK o una CLI](#)

- [Utilizzo DescribeKey con un AWS SDK o una CLI](#)
- [Utilizzo DisableKey con un AWS SDK o una CLI](#)
- [Utilizzo EnableKey con un AWS SDK o una CLI](#)
- [Utilizzo EnableKeyRotation con un AWS SDK o una CLI](#)
- [Utilizzo Encrypt con un AWS SDK o una CLI](#)
- [Utilizzo GenerateDataKey con un AWS SDK o una CLI](#)
- [Utilizzo GenerateDataKeyWithoutPlaintext con un AWS SDK o una CLI](#)
- [Utilizzo GenerateRandom con un AWS SDK o una CLI](#)
- [Utilizzo GetKeyPolicy con un AWS SDK o una CLI](#)
- [Utilizzo ListAliases con un AWS SDK o una CLI](#)
- [Utilizzo ListGrants con un AWS SDK o una CLI](#)
- [Utilizzo ListKeyPolicies con un AWS SDK o una CLI](#)
- [Utilizzo ListKeys con un AWS SDK o una CLI](#)
- [Utilizzo PutKeyPolicy con un AWS SDK o una CLI](#)
- [Utilizzo ReEncrypt con un AWS SDK o una CLI](#)
- [Utilizzo RetireGrant con un AWS SDK o una CLI](#)
- [Utilizzo RevokeGrant con un AWS SDK o una CLI](#)
- [Utilizzo ScheduleKeyDeletion con un AWS SDK o una CLI](#)
- [Utilizzo Sign con un AWS SDK o una CLI](#)
- [Utilizzo TagResource con un AWS SDK o una CLI](#)
- [Utilizzo UpdateAlias con un AWS SDK o una CLI](#)
- [Utilizzo Verify con un AWS SDK o una CLI](#)

Utilizzo **CreateAlias** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare CreateAlias.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Informazioni di base](#)

.NET

SDK per .NET

Note

C'è altro su GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Creates an alias for an AWS Key Management Service (AWS KMS) key.
/// </summary>
public class CreateAlias
{
    public static async Task Main()
    {
        var client = new AmazonKeyManagementServiceClient();

        // The alias name must start with alias/ and can be
        // up to 256 alphanumeric characters long.
        var aliasName = "alias/ExampleAlias";

        // The value supplied as the TargetKeyId can be either
        // the key ID or key Amazon Resource Name (ARN) of the
        // AWS KMS key.
        var keyId = "1234abcd-12ab-34cd-56ef-1234567890ab";

        var request = new CreateAliasRequest
        {
            AliasName = aliasName,
            TargetKeyId = keyId,
        };

        var response = await client.CreateAliasAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
```



```
        {  
            Console.WriteLine($"Alias, {aliasName}, successfully created.");  
        }  
        else  
        {  
            Console.WriteLine($"Could not create alias.");  
        }  
    }  
}
```

- Per i dettagli sull'API, [CreateAlias](#) consulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Come creare un alias per una chiave KMS

Il comando `create-alias` seguente crea un alias denominato `example-alias` per la chiave KMS identificata dall'ID chiave `1234abcd-12ab-34cd-56ef-1234567890ab`.

Un nome di alias deve iniziare con `alias/`. Non utilizzare alias che iniziano con `alias/aws`; questi sono riservati all'uso di AWS.

```
aws kms create-alias \  
    --alias-name alias/example-alias \  
    --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non restituisce output. Per visualizzare il nuovo alias, utilizza il comando `list-aliases`.

Per ulteriori informazioni, consulta [Utilizzo di alias](#) nella Guida per gli sviluppatori di Servizio AWS di gestione delle chiavi.

- Per i dettagli sull'API, consulta [CreateAlias AWS CLI](#) Command Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * Creates a custom alias for the specified target key asynchronously.
 *
 * @param targetKeyId the ID of the target key for the alias
 * @param aliasName the name of the alias to create
 * @return a {@link CompletableFuture} that completes when the alias creation
 * operation is finished
 */
public CompletableFuture<Void> createCustomAliasAsync(String targetKeyId,
String aliasName) {
    CreateAliasRequest aliasRequest = CreateAliasRequest.builder()
        .aliasName(aliasName)
        .targetKeyId(targetKeyId)
        .build();

    CompletableFuture<CreateAliasResponse> responseFuture =
getAsyncClient().createAlias(aliasRequest);
    responseFuture.whenComplete((response, exception) -> {
        if (exception == null) {
            logger.info("{} was successfully created.", aliasName);
        } else {
            if (exception instanceof ResourceExistsException) {
                logger.info("Alias [{}] already exists. Moving on...",
aliasName);
            } else if (exception instanceof KmsException kmsEx) {
                throw new RuntimeException("KMS error occurred while creating
alias: " + kmsEx.getMessage(), kmsEx);
            } else {
                throw new RuntimeException("An unexpected error occurred
while creating alias: " + exception.getMessage(), exception);
            }
        }
    })
}
```

```
});  
  
    return responseFuture.thenApply(response -> null);  
}
```

- Per i dettagli sull'API, [CreateAlias](#) consulta AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
suspend fun createCustomAlias(  
    targetKeyIdVal: String?,  
    aliasNameVal: String?,  
) {  
    val request =  
        CreateAliasRequest {  
            aliasName = aliasNameVal  
            targetKeyId = targetKeyIdVal  
        }  
  
    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->  
        kmsClient.createAlias(request)  
        println("$aliasNameVal was successfully created")  
    }  
}
```

- Per i dettagli sull'API, [CreateAlias](#) consulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * @param string $keyId
 * @param string $alias
 * @return void
 */
public function createAlias(string $keyId, string $alias)
{
    try{
        $this->client->createAlias([
            'TargetKeyId' => $keyId,
            'AliasName' => $alias,
        ]);
    }catch (KmsException $caught){
        if($caught->getAwsErrorMessage() == "InvalidAliasNameException"){
            echo "The request was rejected because the specified alias name
is not valid.";
        }
        throw $caught;
    }
}
```

- Per i dettagli sull'API, [CreateAlias](#) consulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def create_alias(self, key_id: str, alias: str) -> None:
        """
        Creates an alias for the specified key.

        :param key_id: The ARN or ID of a key to give an alias.
        :param alias: The alias to assign to the key.
        """
        try:
            self.kms_client.create_alias(AliasName=alias, TargetKeyId=key_id)
        except ClientError as err:
            if err.response["Error"]["Code"] == "AlreadyExistsException":
                logger.error(
                    "Could not create the alias %s because it already exists.",
key_id
                )
```

```
else:
    logger.error(
        "Couldn't encrypt text. Here's why: %s",
        err.response["Error"]["Message"],
    )
    raise
```

- Per i dettagli sull'API, consulta [CreateAlias AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un AWS SDK](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **CreateGrant** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare CreateGrant.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Informazioni di base](#)

.NET

SDK per .NET

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
public static async Task Main()
{
    var client = new AmazonKeyManagementServiceClient();

    // The identity that is given permission to perform the operations
    // specified in the grant.
```

```

        var grantee = "arn:aws:iam::111122223333:role/ExampleRole";

        // The identifier of the AWS KMS key to which the grant applies. You
        // can use the key ID or the Amazon Resource Name (ARN) of the KMS
key.
        var keyId = "7c9eccc2-38cb-4c4f-9db3-766ee8dd3ad4";

        var request = new CreateGrantRequest
        {
            GranteePrincipal = grantee,
            KeyId = keyId,

            // A list of operations that the grant allows.
            Operations = new List<string>
            {
                "Encrypt",
                "Decrypt",
            },
        };

        var response = await client.CreateGrantAsync(request);

        string grantId = response.GrantId; // The unique identifier of the
grant.
        string grantToken = response.GrantToken; // The grant token.

        Console.WriteLine($"Id: {grantId}, Token: {grantToken}");
    }
}

```

- Per i dettagli sull'API, [CreateGrant](#) consulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Come creare una concessione

L'esempio `create-grant` seguente crea una concessione che consente all'utente `exampleUser` di utilizzare il comando `decrypt` sulla chiave KMS di esempio `1234abcd-12ab-34cd-56ef-1234567890ab`. Il principale in fase di ritiro è il ruolo

adminRole. La concessione utilizza il vincolo di concessione EncryptionContextSubset per consentire questa autorizzazione solo quando il contesto di crittografia nella richiesta decrypt include la coppia chiave-valore "Department": "IT".

```
aws kms create-grant \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --grantee-principal arn:aws:iam::123456789012:user/exampleUser \  
  --operations Decrypt \  
  --constraints EncryptionContextSubset={Department=IT} \  
  --retiring-principal arn:aws:iam::123456789012:role/adminRole
```

Output:

```
{  
  "GrantId":  
    "1a2b3c4d2f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",  
  "GrantToken": "<grant token here>"  
}
```

Per visualizzare informazioni dettagliate sulla concessione, utilizza il comando `list-grants`.

Per ulteriori informazioni, consulta [Grants in AWS KMS nella AWS Key Management Service Developer Guide](#).

- Per i dettagli sull'API, consulta AWS CLI Command [CreateGrant](#)Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**  
 * Grants permissions to a specified principal on a customer master key (CMK)  
 * asynchronously.  
 */
```



```

    * @param keyId          The unique identifier for the customer master key
    (CMK) that the grant applies to.
    * @param granteePrincipal The principal that is given permission to perform
    the operations that the grant permits on the CMK.
    * @return A {@link CompletableFuture} that, when completed, contains the ID
    of the created grant.
    * @throws RuntimeException If an error occurs during the grant creation
    process.
    */
    public CompletableFuture<String> grantKeyAsync(String keyId, String
    granteePrincipal) {
        List<GrantOperation> grantPermissions = List.of(
            GrantOperation.ENCRYPT,
            GrantOperation.DECRYPT,
            GrantOperation.DESCRIBE_KEY
        );

        CreateGrantRequest grantRequest = CreateGrantRequest.builder()
            .keyId(keyId)
            .name("grant1")
            .granteePrincipal(granteePrincipal)
            .operations(grantPermissions)
            .build();

        CompletableFuture<CreateGrantResponse> responseFuture =
        getAsyncClient().createGrant(grantRequest);
        responseFuture.whenComplete((response, ex) -> {
            if (ex == null) {
                logger.info("Grant created successfully with ID: " +
                response.grantId());
            } else {
                if (ex instanceof KmsException kmsEx) {
                    throw new RuntimeException("Failed to create grant: " +
                    kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred: " +
                    ex.getMessage(), ex);
                }
            }
        });

        return responseFuture.thenApply(CreateGrantResponse::grantId);
    }
}

```

- Per i dettagli sull'API, [CreateGrant](#) consulta AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
suspend fun createNewGrant(
    keyIdVal: String?,
    granteePrincipalVal: String?,
    operation: String,
): String? {
    val operation0b = GrantOperation.fromValue(operation)
    val grantOperationList = ArrayList<GrantOperation>()
    grantOperationList.add(operation0b)

    val request =
        CreateGrantRequest {
            keyId = keyIdVal
            granteePrincipal = granteePrincipalVal
            operations = grantOperationList
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.createGrant(request)
        return response.grantId
    }
}
```

- Per i dettagli sull'API, [CreateGrant](#) consulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

 Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * @param string $keyId
 * @param string $granteePrincipal
 * @param array $operations
 * @param array $grantTokens
 * @return Result
 */
public function createGrant(string $keyId, string $granteePrincipal, array
$operations, array $grantTokens = [])
{
    $args = [
        'KeyId' => $keyId,
        'GranteePrincipal' => $granteePrincipal,
        'Operations' => $operations,
    ];
    if($grantTokens){
        $args['GrantTokens'] = $grantTokens;
    }
    try{
        return $this->client->createGrant($args);
    }catch(KmsException $caught){
        if($caught->getAwsErrorMessage() == "InvalidGrantTokenException"){
            echo "The request was rejected because the specified grant token
is not valid.\n";
        }
        throw $caught;
    }
}
```

- Per i dettagli sull'API, [CreateGrant](#) consulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def create_grant(
        self, key_id: str, principal: str, operations: [str]
    ) -> dict[str, str]:
        """
        Creates a grant for a key that lets a principal generate a symmetric data
        encryption key.

        :param key_id: The ARN or ID of the key.
        :param principal: The principal to grant permission to.
        :param operations: The operations to grant permission for.
        :return: The grant that is created.
        """
        try:
            return self.kms_client.create_grant(
```

```
        KeyId=key_id,  
        GranteePrincipal=principal,  
        Operations=operations,  
    )  
except ClientError as err:  
    logger.error(  
        "Couldn't create a grant on key %s. Here's why: %s",  
        key_id,  
        err.response["Error"]["Message"],  
    )  
    raise
```

- Per i dettagli sull'API, consulta [CreateGrant AWS](#) SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **CreateKey** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare CreateKey.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- [Informazioni di base](#)
- [Utilizzo della crittografia delle tabelle](#)

.NET

SDK per .NET

Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Shows how to create a new AWS Key Management Service (AWS KMS)
/// key.
/// </summary>
public class CreateKey
{
    public static async Task Main()
    {
        // Note that if you need to create a Key in an AWS Region
        // other than the Region defined for the default user, you need to
        // pass the Region to the client constructor.
        var client = new AmazonKeyManagementServiceClient();

        // The call to CreateKeyAsync will create a symmetrical AWS KMS
        // key. For more information about symmetrical and asymmetrical
        // keys, see:
        //
        // https://docs.aws.amazon.com/kms/latest/developerguide/symm-asymm-
choose.html
        var response = await client.CreateKeyAsync(new CreateKeyRequest());

        // The KeyMetadata object contains information about the new AWS KMS
key.
        KeyMetadata keyMetadata = response.KeyMetadata;

        if (keyMetadata is not null)
        {
            Console.WriteLine($"KMS Key: {keyMetadata.KeyId} was successfully
created.");
        }
        else
        {
            Console.WriteLine("Could not create KMS Key.");
        }
    }
}

```

- Per i dettagli sull'API, [CreateKey](#) consulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Esempio 1: creare una chiave KMS gestita dal cliente in AWS KMS

L'esempio `create-key` seguente crea una chiave KMS di crittografia simmetrica.

Per creare la chiave KMS di base, una chiave di crittografia simmetrica, non è necessario specificare alcun parametro. I valori predefiniti per questi parametri creano una chiave di crittografia simmetrica.

Poiché questo comando non specifica una policy della chiave, la chiave KMS ottiene la [policy di chiave predefinita](#) per le chiavi KMS create a livello di programmazione. Per visualizzare la policy della chiave, utilizza il comando `get-key-policy`. Per modificare la policy della chiave, utilizza il comando `put-key-policy`.

```
aws kms create-key
```

Il comando `create-key` restituisce i metadati della chiave, inclusi l'ID chiave e l'ARN della nuova chiave KMS. Puoi utilizzare questi valori per identificare la chiave KMS in altre AWS operazioni KMS. L'output non include i tag. Per visualizzare i tag di una chiave KMS, utilizza `list-resource-tags` command.

Output:

```
{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": "2017-07-05T14:04:55-07:00",
    "CurrentKeyMaterialId":
      "0b7fd7ddb6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "Description": "",
    "Enabled": true,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
```

```

    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "AWS_KMS"
    "EncryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
    ]
  }
}

```

Nota: il comando `create-key` non consente di specificare un alias. Per creare un alias per la nuova chiave KMS, utilizza il comando `create-alias`.

Per ulteriori informazioni, consulta [Creazione di chiavi](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

Esempio 2: come creare una chiave KMS RSA asimmetrica per la crittografia e la decrittografia

L'esempio `create-key` seguente crea una chiave KMS contenente una coppia di chiavi RSA asimmetriche per la crittografia e la decrittografia. La specifica e l'utilizzo della chiave non possono essere modificati dopo che la chiave è stata creata.

```

aws kms create-key \
  --key-spec RSA_4096 \
  --key-usage ENCRYPT_DECRYPT

```

Output:

```

{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2021-04-05T14:04:55-07:00",
    "CustomerMasterKeySpec": "RSA_4096",
    "Description": "",
    "Enabled": true,
    "EncryptionAlgorithms": [
      "RSAES_OAEP_SHA_1",

```



```

        "RSAES_OAEP_SHA_256"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "RSA_4096",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "AWS_KMS"
}
}

```

Per ulteriori informazioni, consulta Chiavi [asimmetriche in AWS KMS nella Guida per gli sviluppatori del servizio di gestione delle AWS chiavi](#).

Esempio 3: come creare una chiave KMS asimmetrica a curva ellittica per la firma e la verifica

Per creare una chiave KMS asimmetrica contenente una coppia di chiavi asimmetriche basate su curva ellittica (ECC) per la firma e la verifica. Il parametro `--key-usage` è obbligatorio anche se `SIGN_VERIFY` è l'unico valore valido per le chiavi KMS ECC. La specifica e l'utilizzo della chiave non possono essere modificati dopo che la chiave è stata creata.

```

aws kms create-key \
  --key-spec ECC_NIST_P521 \
  --key-usage SIGN_VERIFY

```

Output:

```

{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2019-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "ECC_NIST_P521",
    "Description": "",
    "Enabled": true,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "ECC_NIST_P521",
    "KeyState": "Enabled",
    "KeyUsage": "SIGN_VERIFY",
  }
}

```

```

    "MultiRegion": false,
    "Origin": "AWS_KMS",
    "SigningAlgorithms": [
        "ECDSA_SHA_512"
    ]
}
}

```

Per ulteriori informazioni, consulta [Chiavi asimmetriche in AWS KMS nella Key Management Service Developer Guide](#).AWS

Esempio 4: come creare una chiave KMS ML-DSA asimmetrica per la firma e la verifica

Questo esempio crea una chiave basata sullo schema ML-DSA (Module-Lattice Digital Signature Algorithm) per la firma e la verifica. Il parametro `key-usage` è obbligatorio anche se `SIGN_VERIFY` è l'unico valore valido per le chiavi ML-DSA.

```

aws kms create-key \
  --key-spec ML_DSA_65 \
  --key-usage SIGN_VERIFY

```

Output:

```

{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2019-12-02T07:48:55-07:00",
    "Description": "",
    "Enabled": true,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "ML_DSA_65",
    "KeyState": "Enabled",
    "KeyUsage": "SIGN_VERIFY",
    "MultiRegion": false,
    "Origin": "AWS_KMS",
    "SigningAlgorithms": [
        "ML_DSA_SHAKE_256"
    ]
  }
}

```

```
}
```

Per ulteriori informazioni, consulta [Chiavi asimmetriche in AWS KMS nella Key Management Service Developer Guide](#).AWS

Esempio 5: come creare una chiave KMS HMAC

L'esempio `create-key` seguente crea una chiave KMS HMAC a 384 bit. Il valore `GENERATE_VERIFY_MAC` del parametro `--key-usage` è obbligatorio anche se è l'unico valore valido per le chiavi KMS HMAC.

```
aws kms create-key \  
  --key-spec HMAC_384 \  
  --key-usage GENERATE_VERIFY_MAC
```

Output:

```
{  
  "KeyMetadata": {  
    "Arn": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
    "AWSAccountId": "111122223333",  
    "CreationDate": "2022-04-05T14:04:55-07:00",  
    "CustomerMasterKeySpec": "HMAC_384",  
    "Description": "",  
    "Enabled": true,  
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
    "KeyManager": "CUSTOMER",  
    "KeySpec": "HMAC_384",  
    "KeyState": "Enabled",  
    "KeyUsage": "GENERATE_VERIFY_MAC",  
    "MacAlgorithms": [  
      "HMAC_SHA_384"  
    ],  
    "MultiRegion": false,  
    "Origin": "AWS_KMS"  
  }  
}
```

Per ulteriori informazioni, consulta le [chiavi HMAC in AWS KMS nella Key Management Service Developer Guide](#).AWS

Esempio 6: come creare una chiave KMS primaria multi-Regione

L'esempio `create-key` seguente crea una chiave di crittografia simmetrica primaria multi-Regione. Poiché i valori predefiniti per tutti i parametri creano una chiave di crittografia simmetrica, per questa chiave KMS è richiesto solo il parametro `--multi-region`. Nella AWS CLI, per indicare che un parametro booleano è vero, è sufficiente specificare il nome del parametro.

```
aws kms create-key \  
  --multi-region
```

Output:

```
{  
  "KeyMetadata": {  
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/  
mrk-1234abcd12ab34cd56ef12345678990ab",  
    "AWSAccountId": "111122223333",  
    "CreationDate": "2021-09-02T016:15:21-09:00",  
    "CurrentKeyMaterialId":  
"0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",  
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",  
    "Description": "",  
    "Enabled": true,  
    "EncryptionAlgorithms": [  
      "SYMMETRIC_DEFAULT"  
    ],  
    "KeyId": "mrk-1234abcd12ab34cd56ef12345678990ab",  
    "KeyManager": "CUSTOMER",  
    "KeySpec": "SYMMETRIC_DEFAULT",  
    "KeyState": "Enabled",  
    "KeyUsage": "ENCRYPT_DECRYPT",  
    "MultiRegion": true,  
    "MultiRegionConfiguration": {  
      "MultiRegionKeyType": "PRIMARY",  
      "PrimaryKey": {  
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/  
mrk-1234abcd12ab34cd56ef12345678990ab",  
        "Region": "us-west-2"  
      },  
      "ReplicaKeys": []  
    },  
  },  
}
```

```
    "Origin": "AWS_KMS"
  }
}
```

Per ulteriori informazioni, consulta la sezione [Chiavi asimmetriche in AWS KMS nella Key Management Service Developer Guide](#).AWS

Esempio 7: come creare una chiave KMS per il materiale della chiave importato

L'esempio `create-key` seguente crea una chiave KMS senza materiale di chiave. Quando l'operazione è stata completata, puoi importare il materiale della chiave nella chiave KMS. Per creare questa chiave KMS, imposta il parametro `--origin` su `EXTERNAL`.

```
aws kms create-key \
  --origin EXTERNAL
```

Output:

```
{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2019-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "Description": "",
    "Enabled": false,
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "PendingImport",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "EXTERNAL"
  }
}
```

Per ulteriori informazioni, consulta [Importazione di materiale chiave nelle chiavi AWS KMS nella Key Management Service Developer Guide](#).AWS

Esempio 6: creare una chiave KMS in un key store AWS CloudHSM

L'operazione `create-key` seguente crea una chiave KMS nell'archivio di chiavi AWS CloudHSM specificato. L'operazione crea la chiave KMS e i relativi metadati in AWS KMS e crea il materiale chiave nel cluster CloudHSM associato all'AWS archivio chiavi personalizzato. I parametri `--custom-key-store-id` e `--origin` sono obbligatori.

```
aws kms create-key \
  --origin AWS_CLOUDHSM \
  --custom-key-store-id cks-1234567890abcdef0
```

Output:

```
{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CloudHsmClusterId": "cluster-1a23b4cdefg",
    "CreationDate": "2019-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "CustomKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "CustomKeyId": "cks-1234567890abcdef0",
    "Description": "",
    "Enabled": true,
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "AWS_CLOUDHSM"
  }
}
```

Per ulteriori informazioni, consulta [AWS CloudHSM key stores](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

Esempio 8: come creare una chiave KMS in un archivio chiavi esterno

L'esempio `create-key` seguente crea una chiave KMS nell'archivio chiavi esterno specificato. I parametri `--custom-key-store-id`, `--origin` e `--xks-key-id` sono tutti obbligatori in questo comando.

Il parametro `--xks-key-id` specifica l'ID di una chiave di crittografia simmetrica esistente nel gestore chiavi esterno. Questa chiave funge da materiale della chiave esterna per la chiave KMS. Il valore del parametro `--origin` deve essere `EXTERNAL_KEY_STORE`. Il parametro `custom-key-store-id` deve identificare un archivio chiavi esterno collegato al proxy dell'archivio chiavi esterno.

```
aws kms create-key \
  --origin EXTERNAL_KEY_STORE \
  --custom-key-store-id cks-9876543210fedcba9 \
  --xks-key-id bb8562717f809024
```

Output:

```
{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2022-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "CustomKeyId": "cks-9876543210fedcba9",
    "Description": "",
    "Enabled": true,
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "EXTERNAL_KEY_STORE",
    "XksKeyConfiguration": {
      "Id": "bb8562717f809024"
    }
  }
}
```

```
}
```

Per ulteriori informazioni, consulta [Archivi delle chiavi esterne](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

- Per i dettagli sull'API, consulta Command Reference. [CreateKey](#) AWS CLI

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * Creates a new symmetric encryption key asynchronously.
 *
 * @param keyDesc the description of the key to be created
 * @return a {@link CompletableFuture} that completes with the ID of the
 * newly created key
 * @throws RuntimeException if an error occurs while creating the key
 */
public CompletableFuture<String> createKeyAsync(String keyDesc) {
    CreateKeyRequest keyRequest = CreateKeyRequest.builder()
        .description(keyDesc)
        .keySpec(KeySpec.SYMMETRIC_DEFAULT)
        .keyUsage(KeyUsageType.ENCRYPT_DECRYPT)
        .build();

    return getAsyncClient().createKey(keyRequest)
        .thenApply(resp -> resp.keyMetadata().keyId())
        .exceptionally(ex -> {
            throw new RuntimeException("An error occurred while creating the
key: " + ex.getMessage(), ex);
        });
}
```

- Per i dettagli sull'API, [CreateKey](#) consulta AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
suspend fun createKey(keyDesc: String?): String? {
    val request =
        CreateKeyRequest {
            description = keyDesc
            customerMasterKeySpec = CustomerMasterKeySpec.SymmetricDefault
            keyUsage = KeyUsageType.fromValue("ENCRYPT_DECRYPT")
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val result = kmsClient.createKey(request)
        println("Created a customer key with id " + result.keyMetadata?.arn)
        return result.keyMetadata?.keyId
    }
}
```

- Per i dettagli sull'API, [CreateKey](#) consulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
```

```

    * @param string $keySpec
    * @param string $keyUsage
    * @param string $description
    * @return array
    */
    public function createKey(string $keySpec = "", string $keyUsage = "", string
    $description = "Created by the SDK for PHP")
    {
        $parameters = ['Description' => $description];
        if($keySpec && $keyUsage){
            $parameters['KeySpec'] = $keySpec;
            $parameters['KeyUsage'] = $keyUsage;
        }
        try {
            $result = $this->client->createKey($parameters);
            return $result['KeyMetadata'];
        }catch(KmsException $caught){
            // Check for error specific to createKey operations
            if ($caught->getAwsErrorMessage() == "LimitExceededException"){
                echo "The request was rejected because a quota was exceeded. For
                more information, see Quotas in the Key Management Service Developer Guide.";
            }
            throw $caught;
        }
    }
}

```

- Per i dettagli sull'API, [CreateKey](#) consulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

class KeyManager:
    def __init__(self, kms_client):

```

```

        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def create_key(self, key_description: str) -> dict[str, any]:
        """
        Creates a key with a user-provided description.

        :param key_description: A description for the key.
        :return: The key ID.
        """
        try:
            key = self.kms_client.create_key(Description=key_description)
["KeyMetadata"]
            self.created_keys.append(key)
            return key
        except ClientError as err:
            logging.error(
                "Couldn't create your key. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise

```

- Per i dettagli sull'API, consulta [CreateKey AWS SDK for Python \(Boto3\) API Reference](#).

Ruby

SDK per Ruby

Note

C'è di più su. [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
require 'aws-sdk-kms' # v2: require 'aws-sdk'

# Create a AWS KMS key.
# As long we are only encrypting small amounts of data (4 KiB or less) directly,
# a KMS key is fine for our purposes.
# For larger amounts of data,
# use the KMS key to encrypt a data encryption key (DEK).

client = Aws::KMS::Client.new

resp = client.create_key({
  tags: [
    {
      tag_key: 'CreatedBy',
      tag_value: 'ExampleUser'
    }
  ]
})

puts resp.key_metadata.key_id
```

- Per i dettagli sull'API, [CreateKey](#) consulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
async fn make_key(client: &Client) -> Result<(), Error> {  
    let resp = client.create_key().send().await?;  
  
    let id = resp.key_metadata.as_ref().unwrap().key_id();  
  
    println!("Key: {}", id);  
  
    Ok(())  
}
```

- Per i dettagli sulle API, consulta la [CreateKey](#) guida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **Decrypt** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare Decrypt.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Informazioni di base](#)

CLI

AWS CLI

Esempio 1: come decrittografare un messaggio crittografato con una chiave KMS simmetrica (Linux e macOS)

Il seguente esempio di `decrypt` comando mostra il metodo consigliato per decrittografare i dati con la CLI. AWS Questa versione mostra come decrittografare i dati con una chiave KMS simmetrica.

Fornisci il testo criptato in un file. Nel valore del parametro `--ciphertext-blob`, utilizza il prefisso `fileb://` per indicare alla CLI di leggere i dati da un file binario. Se il file non è presente nella directory corrente, digita il percorso completo del file. Per ulteriori informazioni sulla lettura dei valori dei parametri AWS CLI da un file, consulta [Loading AWS CLI parameters from a file < https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-parameters-file .html>](https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-parameters-file.html) nella Command Line Interface User Guide e [Best Practices for Local File Parameters< https://aws.amazon.com/blogs/developer/best-practices-for-local-file-parameters/>](https://aws.amazon.com/blogs/developer/best-practices-for-local-file-parameters/) nel AWS Command Line Tool Blog .Specificare la chiave KMS per decrittografare il `Ciphertext`. Il AWS parametro non è necessario quando si esegue la decrittografia con una chiave KMS simmetrica. `--key-id` AWS KMS può ottenere l'ID della chiave KMS utilizzata per crittografare i dati dai metadati nel testo cifrato. Tuttavia è sempre consigliabile specificare la chiave KMS che stai utilizzando. Questa procedura consente di utilizzare la chiave KMS desiderata e impedire di decrittografare inavvertitamente un testo criptato utilizzando una chiave KMS non attendibile. Richiedi l'output in testo normale come valore di testo. Il parametro `--query` indica alla CLI di recuperare solo il valore del campo `Plaintext` dall'output. Il parametro `--output` restituisce l'output come testo. Procedi a decrittografare in formato Base64 il testo normale e a salvarlo in un file. Nell'esempio seguente il valore del parametro `Plaintext` viene inviato (`|`) all'utilità Base64, che lo decodifica. Quindi, reindirizza (`>`) l'output decodificato al file `ExamplePlaintext`.

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID chiave valido del tuo account. AWS

```
aws kms decrypt \
  --ciphertext-blob fileb://ExampleEncryptedFile \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --output text \
  --query Plaintext | base64 \
```

```
--decode > ExamplePlaintextFile
```

Questo comando non produce alcun output. L'output del comando decrypt viene decrittografato in formato Base64 e salvato in un file.

Per ulteriori informazioni, consulta [Decrypt](#) nel riferimento API del Servizio AWS di gestione delle chiavi.

Esempio 2: come decrittografare un messaggio crittografato con una chiave KMS simmetrica (prompt dei comandi di Windows)

L'esempio seguente è lo stesso del precedente, tranne per il fatto che utilizza l'utilità `certutil` per decrittografare in formato Base64 i dati del testo normale. Questa procedura richiede due comandi, come illustrato negli esempi seguenti.

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID chiave valido del tuo AWS account.

```
aws kms decrypt ^  
  --ciphertext-blob fileb://ExampleEncryptedFile ^  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^  
  --output text ^  
  --query Plaintext > ExamplePlaintextFile.base64
```

Esegui il comando `certutil`.

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Output:

```
Input Length = 18  
Output Length = 12  
CertUtil: -decode command completed successfully.
```

Per ulteriori informazioni, consulta [Decrypt](#) nel riferimento API del Servizio AWS di gestione delle chiavi.

Esempio 3: come decrittografare un messaggio crittografato con una chiave KMS asimmetrica (Linux e macOS)

Il seguente esempio di comando `decrypt` mostra come decrittografare i dati crittografati con una chiave KMS asimmetrica RSA.

Quando si utilizza una chiave KMS asimmetrica, il parametro `encryption-algorithm` obbligatorio specifica l'algoritmo utilizzato per crittografare il testo normale.

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID chiave valido del tuo AWS account.

```
aws kms decrypt \
  --ciphertext-blob fileb://ExampleEncryptedFile \
  --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
  --encryption-algorithm RSAES_OAEP_SHA_256 \
  --output text \
  --query Plaintext | base64 \
  --decode > ExamplePlaintextFile
```

Questo comando non produce alcun output. L'output del comando `decrypt` viene decrittografato in formato Base64 e salvato in un file.

Per ulteriori informazioni, consulta Chiavi [asimmetriche in AWS KMS nella Guida](#) per gli sviluppatori del servizio di gestione delle AWS chiavi.

- Per informazioni dettagliate sull'API, consulta [Decrypt](#) in AWS CLI Command Reference.

Java

SDK per Java 2.x

Note

C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * Asynchronously decrypts the given encrypted data using the specified key
ID.
 *
 * @param encryptedData The encrypted data to be decrypted.
```



```

    * @param keyId The ID of the key to be used for decryption.
    * @return A CompletableFuture that, when completed, will contain the
    decrypted data as a String.
    *         If an error occurs during the decryption process, the
    CompletableFuture will complete
    *         exceptionally with the error, and the method will return an empty
    String.
    */
    public CompletableFuture<String> decryptDataAsync(SdkBytes encryptedData,
    String keyId) {
        DecryptRequest decryptRequest = DecryptRequest.builder()
            .ciphertextBlob(encryptedData)
            .keyId(keyId)
            .build();

        CompletableFuture<DecryptResponse> responseFuture =
        getAsyncClient().decrypt(decryptRequest);
        responseFuture.whenComplete((decryptResponse, exception) -> {
            if (exception == null) {
                logger.info("Data decrypted successfully for key ID: " + keyId);
            } else {
                if (exception instanceof KmsException kmsEx) {
                    throw new RuntimeException("KMS error occurred while
decrypting data: " + kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred
while decrypting data: " + exception.getMessage(), exception);
                }
            }
        });

        return responseFuture.thenApply(decryptResponse ->
        decryptResponse.plaintext().asString(StandardCharsets.UTF_8));
    }

```

- Per i dettagli sull'API, [consulta Decrypt](#) in AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

Note

C'è altro su GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
suspend fun encryptData(keyIdValue: String): ByteArray? {
    val text = "This is the text to encrypt by using the AWS KMS Service"
    val myBytes: ByteArray = text.toByteArray()

    val encryptRequest =
        EncryptRequest {
            keyId = keyIdValue
            plaintext = myBytes
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.encrypt(encryptRequest)
        val algorithm: String = response.encryptionAlgorithm.toString()
        println("The encryption algorithm is $algorithm")

        // Return the encrypted data.
        return response.ciphertextBlob
    }
}

suspend fun decryptData(
    encryptedDataVal: ByteArray?,
    keyIdVal: String?,
) {
    val decryptRequest =
        DecryptRequest {
            ciphertextBlob = encryptedDataVal
            keyId = keyIdVal
        }

    KmsClient { region = "us-west-2" }.use { kmsClient ->
        val decryptResponse = kmsClient.decrypt(decryptRequest)
        val myVal = decryptResponse.plaintext
    }
}
```

```
        // Print the decrypted data.
        print(myVal)
    }
}
```

- Per i dettagli sull'API, [consulta Decrypt](#) in AWS SDK per il riferimento all'API Kotlin.

PHP

SDK per PHP

Note

C'è di più su. [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * @param string $keyId
 * @param string $ciphertext
 * @param string $algorithm
 * @return Result
 */
public function decrypt(string $keyId, string $ciphertext, string $algorithm
= "SYMMETRIC_DEFAULT")
{
    try{
        return $this->client->decrypt([
            'CiphertextBlob' => $ciphertext,
            'EncryptionAlgorithm' => $algorithm,
            'KeyId' => $keyId,
        ]);
    }catch(KmsException $caught){
        echo "There was a problem decrypting the data: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}
```

- Per i dettagli sull'API, [consulta Decrypt](#) in AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def decrypt(self, key_id: str, cipher_text: bytes) -> str:
        """
        Decrypts text previously encrypted with a key.

        :param key_id: The ARN or ID of the key used to decrypt the data.
        :param cipher_text: The encrypted text to decrypt.
        :return: The decrypted text.
        """
        try:
            return self.kms_client.decrypt(KeyId=key_id,
            CiphertextBlob=cipher_text)[
                "Plaintext"]
```

```

        ].decode()
    except ClientError as err:
        logger.error(
            "Couldn't decrypt your ciphertext. Here's why: %s",
            err.response["Error"]["Message"],
        )
        raise

```

- Per i dettagli sull'API, [consulta Decrypt](#) in AWS SDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

Note

C'è GitHub di più su. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

require 'aws-sdk-kms' # v2: require 'aws-sdk'

# Decrypted blob

blob =
  '01020200785d68faeec386af1057904926253051eb2919d3c16078badf65b808b26dd057c101747cadf3593'
blob_packed = [blob].pack('H*')

client = Aws::KMS::Client.new(region: 'us-west-2')

resp = client.decrypt({
  ciphertext_blob: blob_packed
})

puts 'Raw text: '
puts resp.plaintext

```

- Per i dettagli sull'API, [consulta Decrypt](#) in AWS SDK per Ruby API Reference.

Rust

SDK per Rust

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
async fn decrypt_key(client: &Client, key: &str, filename: &str) -> Result<(),
Error> {
    // Open input text file and get contents as a string
    // input is a base-64 encoded string, so decode it:
    let data = fs::read_to_string(filename)
        .map(|input| {
            base64::decode(input).expect("Input file does not contain valid base
64 characters.")
        })
        .map(Blob::new);

    let resp = client
        .decrypt()
        .key_id(key)
        .ciphertext_blob(data.unwrap())
        .send()
        .await?;

    let inner = resp.plaintext.unwrap();
    let bytes = inner.as_ref();

    let s = String::from_utf8(bytes.to_vec()).expect("Could not convert to
UTF-8");

    println!();
    println!("Decoded string:");
    println!("{}", s);

    Ok(())
}
```

- Per i dettagli sull'API, [consulta Decrypt](#) in AWS SDK for Rust API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DeleteAlias** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare DeleteAlias.

CLI

AWS CLI

Per eliminare un alias AWS KMS

L'esempio delete-alias seguente elimina l'alias alias/example-alias. Il nome di alias deve iniziare con alias/.

```
aws kms delete-alias \  
  --alias-name alias/example-alias
```

Questo comando non produce alcun output. Per cercare l'alias, utilizza il comando list-aliases.

Per ulteriori informazioni, consulta [Eliminazione di un alias](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

- Per i dettagli sull'API, consulta AWS CLI Command [DeleteAlias](#)Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

/**
 * Deletes a specific KMS alias asynchronously.
 *
 * @param aliasName the name of the alias to be deleted
 * @return a {@link CompletableFuture} representing the asynchronous
operation of deleting the specified alias
 */
public CompletableFuture<Void> deleteSpecificAliasAsync(String aliasName) {
    DeleteAliasRequest deleteAliasRequest = DeleteAliasRequest.builder()
        .aliasName(aliasName)
        .build();

    return getAsyncClient().deleteAlias(deleteAliasRequest)
        .thenRun(() -> {
            logger.info("Alias {} has been deleted successfully", aliasName);
        })
        .exceptionally(throwable -> {
            throw new RuntimeException("Failed to delete alias: " +
aliasName, throwable);
        });
}

```

- Per i dettagli sull'API, [DeleteAlias](#) consulta AWS SDK for Java 2.x API Reference.

PHP

SDK per PHP

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

/**
 * @param string $aliasName
 * @return void
 */
public function deleteAlias(string $aliasName)

```



```

{
    try {
        $this->client->deleteAlias([
            'AliasName' => $aliasName,
        ]);
    } catch (KmsException $caught){
        echo "There was a problem deleting the alias: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}

```

- Per i dettagli sull'API, [DeleteAlias](#) consulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

```

```
def delete_alias(self, alias: str) -> None:
    """
    Deletes an alias.

    :param alias: The alias to delete.
    """
    try:
        self.kms_client.delete_alias(AliasName=alias)
    except ClientError as err:
        logger.error(
            "Couldn't delete alias %s. Here's why: %s",
            alias,
            err.response["Error"]["Message"],
        )
        raise
```

- Per i dettagli sull'API, consulta [DeleteAlias AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un AWS SDK](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DescribeKey** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `DescribeKey`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Informazioni di base](#)

.NET

SDK per .NET

Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Retrieve information about an AWS Key Management Service (AWS KMS) key.
/// You can supply either the key Id or the key Amazon Resource Name (ARN)
/// to the DescribeKeyRequest KeyId property.
/// </summary>
public class DescribeKey
{
    public static async Task Main()
    {
        var keyId = "7c9eccc2-38cb-4c4f-9db3-766ee8dd3ad4";
        var request = new DescribeKeyRequest
        {
            KeyId = keyId,
        };

        var client = new AmazonKeyManagementServiceClient();

        var response = await client.DescribeKeyAsync(request);
        var metadata = response.KeyMetadata;

        Console.WriteLine($"{metadata.KeyId} created on:
{metadata.CreationDate}");
        Console.WriteLine($"State: {metadata.KeyState}");
        Console.WriteLine($"{metadata.Description}");
    }
}
```

- Per i dettagli sull'API, [DescribeKey](#) consulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Esempio 1: come trovare informazioni dettagliate su una chiave KMS

L'`describe-key` seguente ottiene informazioni dettagliate sulla chiave AWS gestita per Amazon S3 nell'account e nella regione di esempio. Puoi utilizzare questo comando per trovare dettagli sulle chiavi gestite e sulle chiavi AWS gestite dai clienti.

Per specificare la chiave KMS, utilizza il parametro `key-id`. Questo esempio utilizza un valore per il nome di alias, ma in questo comando puoi utilizzare un ID di chiave, un ARN di chiave o un ARN di alias.

```
aws kms describe-key \  
  --key-id alias/aws/s3
```

Output:

```
{  
  "KeyMetadata": {  
    "AWSAccountId": "846764612917",  
    "KeyId": "b8a9477d-836c-491f-857e-07937918959b",  
    "Arn": "arn:aws:kms:us-west-2:846764612917:key/  
b8a9477d-836c-491f-857e-07937918959b",  
    "CurrentKeyMaterialId":  
"0b7fd7ddb6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",  
    "CreationDate": 2017-06-30T21:44:32.140000+00:00,  
    "Enabled": true,  
    "Description": "Default KMS key that protects my S3 objects when no other  
key is defined",  
    "KeyUsage": "ENCRYPT_DECRYPT",  
    "KeyState": "Enabled",  
    "Origin": "AWS_KMS",  
    "KeyManager": "AWS",  
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",  
    "EncryptionAlgorithms": [  

```

```

        "SYMMETRIC_DEFAULT"
    ]
}
}

```

Per ulteriori informazioni, consulta [Visualizzazione di chiavi](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

Esempio 2: come ottenere dettagli su una chiave KMS RSA asimmetrica

L'esempio `describe-key` seguente ottiene informazioni dettagliate su una chiave KMS RSA asimmetrica utilizzata per la firma e la verifica.

```

aws kms describe-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

```

Output:

```

{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": "2019-12-02T19:47:14.861000+00:00",
    "CustomerMasterKeySpec": "RSA_2048",
    "Enabled": false,
    "Description": "",
    "KeyState": "Disabled",
    "Origin": "AWS_KMS",
    "MultiRegion": false,
    "KeyManager": "CUSTOMER",
    "KeySpec": "RSA_2048",
    "KeyUsage": "SIGN_VERIFY",
    "SigningAlgorithms": [
      "RSASSA_PKCS1_V1_5_SHA_256",
      "RSASSA_PKCS1_V1_5_SHA_384",
      "RSASSA_PKCS1_V1_5_SHA_512",
      "RSASSA_PSS_SHA_256",
      "RSASSA_PSS_SHA_384",
      "RSASSA_PSS_SHA_512"
    ]
  }
}

```

```
}
}
```

Esempio 3: come ottenere dettagli su una chiave di replica multi-Regione

L'esempio `describe-key` seguente ottiene i metadati per una chiave di replica multi-Regione. Questa chiave multi-Regione è una chiave di crittografia simmetrica. L'output di un comando `describe-key` per qualsiasi chiave multi-Regione restituisce informazioni sulla chiave primaria e su tutte le relative repliche.

```
aws kms describe-key \
  --key-id arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab
```

Output:

```
{
  "KeyMetadata": {
    "MultiRegion": true,
    "AWSAccountId": "111122223333",
    "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
    "CreationDate": "2021-06-28T21:09:16.114000+00:00",
    "CurrentKeyMaterialId":
"0b7fd7ddb6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
    "Description": "",
    "Enabled": true,
    "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "Origin": "AWS_KMS",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "MultiRegionConfiguration": {
      "MultiRegionKeyType": "PRIMARY",
      "PrimaryKey": {
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "Region": "us-west-2"
```

```

    },
    "ReplicaKeys": [
      {
        "Arn": "arn:aws:kms:eu-west-1:111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab",
        "Region": "eu-west-1"
      },
      {
        "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab",
        "Region": "ap-northeast-1"
      },
      {
        "Arn": "arn:aws:kms:sa-east-1:111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab",
        "Region": "sa-east-1"
      }
    ]
  }
}

```

Esempio 4: come ottenere dettagli su una chiave KMS HMAC

L'esempio `describe-key` seguente ottiene informazioni dettagliate su una chiave KMS HMAC.

```

aws kms describe-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

```

Output:

```

{
  "KeyMetadata": {
    "AWSAccountId": "123456789012",
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "Arn": "arn:aws:kms:us-west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": "2022-04-03T22:23:10.194000+00:00",
    "Enabled": true,
    "Description": "Test key",
    "KeyUsage": "GENERATE_VERIFY_MAC",

```

```

    "KeyState": "Enabled",
    "Origin": "AWS_KMS",
    "KeyManager": "CUSTOMER",
    "CustomerMasterKeySpec": "HMAC_256",
    "MacAlgorithms": [
        "HMAC_SHA_256"
    ],
    "MultiRegion": false
}

```

- Per i dettagli sull'API, consulta [DescribeKey AWS CLI Command Reference](#).

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

/**
 * Asynchronously checks if a specified key is enabled.
 *
 * @param keyId the ID of the key to check
 * @return a {@link CompletableFuture} that, when completed, indicates
 * whether the key is enabled or not
 *
 * @throws RuntimeException if an exception occurs while checking the key
 * state
 */
public CompletableFuture<Boolean> isKeyEnabledAsync(String keyId) {
    DescribeKeyRequest keyRequest = DescribeKeyRequest.builder()
        .keyId(keyId)
        .build();

    CompletableFuture<DescribeKeyResponse> responseFuture =
getAsyncClient().describeKey(keyRequest);
    return responseFuture.whenComplete((resp, ex) -> {
        if (resp != null) {

```



```

        KeyState keyState = resp.keyMetadata().keyState();
        if (keyState == KeyState.ENABLED) {
            logger.info("The key is enabled.");
        } else {
            logger.info("The key is not enabled. Key state: {}",
keyState);
        }
    } else {
        throw new RuntimeException(ex);
    }
}).thenApply(resp -> resp.keyMetadata().keyState() == KeyState.ENABLED);
}

```

- Per i dettagli sull'API, [DescribeKey](#) consulta AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

suspend fun describeSpecifcKey(keyIdVal: String?) {
    val request =
        DescribeKeyRequest {
            keyId = keyIdVal
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.describeKey(request)
        println("The key description is ${response.keyMetadata?.description}")
        println("The key ARN is ${response.keyMetadata?.arn}")
    }
}

```

- Per i dettagli sull'API, [DescribeKey](#) consulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

Note

C'è di più su. [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * @param string $keyId
 * @return array
 */
public function describeKey(string $keyId)
{
    try {
        $result = $this->client->describeKey([
            "KeyId" => $keyId,
        ]);
        return $result['KeyMetadata'];
    } catch (KmsException $caught) {
        if ($caught->getAwsErrorMessage() == "NotFoundException") {
            echo "The request was rejected because the specified entity or
resource could not be found.\n";
        }
        throw $caught;
    }
}
```

- Per i dettagli sull'API, [DescribeKey](#) consulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def describe_key(self, key_id: str) -> dict[str, any]:
        """
        Describes a key.

        :param key_id: The ARN or ID of the key to describe.
        :return: Information about the key.
        """

        try:
            key = self.kms_client.describe_key(KeyId=key_id)["KeyMetadata"]
            return key
        except ClientError as err:
            logging.error(
                "Couldn't get key '%s'. Here's why: %s",
                key_id,
```

```
        err.response["Error"]["Message"],
    )
    raise
```

- Per i dettagli sull'API, consulta [DescribeKey AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DisableKey** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `DisableKey`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Informazioni di base](#)

.NET

SDK per .NET

Note

C'è altro su. [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Disable an AWS Key Management Service (AWS KMS) key and then retrieve
/// the key's status to show that it has been disabled.
/// </summary>
```

```
public class DisableKey
{
    public static async Task Main()
    {
        var client = new AmazonKeyManagementServiceClient();

        // The identifier of the AWS KMS key to disable. You can use the
        // key Id or the Amazon Resource Name (ARN) of the AWS KMS key.
        var keyId = "1234abcd-12ab-34cd-56ef-1234567890ab";

        var request = new DisableKeyRequest
        {
            KeyId = keyId,
        };

        var response = await client.DisableKeyAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            // Retrieve information about the key to show that it has now
            // been disabled.
            var describeResponse = await client.DescribeKeyAsync(new
DescribeKeyRequest
            {
                KeyId = keyId,
            });
            Console.WriteLine($"{describeResponse.KeyMetadata.KeyId} - state:
{describeResponse.KeyMetadata.KeyState}");
        }
    }
}
```

- Per i dettagli sull'API, [DisableKey](#) consulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Come disabilitare temporaneamente una chiave KMS

Il seguente comando `disable-key` disabilita una chiave KMS gestita dal cliente. Per riabilitare la chiave KMS, utilizza il comando `enable-key`.

```
aws kms disable-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Abilitazione e disabilitazione delle chiavi](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

- Per i dettagli sull'API, consulta [DisableKey AWS CLI Command Reference](#).

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**  
 * Asynchronously disables the specified AWS Key Management Service (KMS)  
key.  
 *  
 * @param keyId the ID or Amazon Resource Name (ARN) of the KMS key to be  
disabled  
 * @return a CompletableFuture that, when completed, indicates that the key  
has been disabled successfully  
 */  
public CompletableFuture<Void> disableKeyAsync(String keyId) {  
    DisableKeyRequest keyRequest = DisableKeyRequest.builder()  
        .keyId(keyId)  
        .build();  
  
    return getAsyncClient().disableKey(keyRequest)  
        .thenRun(() -> {  
            logger.info("Key {} has been disabled successfully",keyId);  
        })  
}
```

```
        .exceptionally(throwable -> {
            throw new RuntimeException("Failed to disable key: " + keyId,
            throwable);
        });
    }
```

- Per i dettagli sull'API, [DisableKey](#) consulta AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
suspend fun disableKey(keyIdVal: String?) {
    val request =
        DisableKeyRequest {
            keyId = keyIdVal
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        kmsClient.disableKey(request)
        println("$keyIdVal was successfully disabled")
    }
}
```

- Per i dettagli sull'API, [DisableKey](#) consulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

Note

C'è di più su [GitHub](#). Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * @param string $keyId
 * @return void
 */
public function disableKey(string $keyId)
{
    try {
        $this->client->disableKey([
            'KeyId' => $keyId,
        ]);
    } catch (KmsException $caught){
        echo "There was a problem disabling the key: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}
```

- Per i dettagli sull'API, [DisableKey](#) consulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su [GitHub](#). Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).


```

class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def disable_key(self, key_id: str) -> None:
        try:
            self.kms_client.disable_key(KeyId=key_id)
        except ClientError as err:
            logging.error(
                "Couldn't disable key '%s'. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise

```

- Per i dettagli sull'API, consulta [DisableKey AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un AWS SDK](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **EnableKey** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare EnableKey.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Informazioni di base](#)

.NET

SDK per .NET

Note

C'è altro su [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Enable an AWS Key Management Service (AWS KMS) key.
/// </summary>
public class EnableKey
{
    public static async Task Main()
    {
        var client = new AmazonKeyManagementServiceClient();

        // The identifier of the AWS KMS key to enable. You can use the
        // key Id or the Amazon Resource Name (ARN) of the AWS KMS key.
        var keyId = "1234abcd-12ab-34cd-56ef-1234567890ab";

        var request = new EnableKeyRequest
        {
            KeyId = keyId,
        };

        var response = await client.EnableKeyAsync(request);
        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            // Retrieve information about the key to show that it has now
            // been enabled.
            var describeResponse = await client.DescribeKeyAsync(new
DescribeKeyRequest
```

```
        {
            KeyId = keyId,
        });
        Console.WriteLine($"{describeResponse.KeyMetadata.KeyId} - state:
{describeResponse.KeyMetadata.KeyState}");
    }
}
```

- Per i dettagli sull'API, [EnableKey](#) consulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Come abilitare una chiave KMS

L'esempio `enable-key` seguente abilita una chiave gestita dal cliente. Puoi utilizzare un comando come il seguente per abilitare una chiave KMS temporaneamente disabilitata utilizzando il comando `disable-key`. Puoi anche usarlo per abilitare una chiave KMS disabilitata perché era stata pianificata per l'eliminazione e l'eliminazione è stata annullata.

Per specificare la chiave KMS, utilizza il parametro `key-id`. Questo esempio utilizza un valore per l'ID chiave, ma in questo comando puoi utilizzare un ID o un ARN di chiave.

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID valido.

```
aws kms enable-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non produce alcun output. Per verificare se la chiave KMS è abilitata, utilizza il comando `describe-key`. Verifica i valori dei campi `KeyState` e `Enabled` nell'output di `describe-key`.

Per ulteriori informazioni, consulta [Abilitazione e disabilitazione delle chiavi](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

- Per i dettagli sull'API, consulta [EnableKey AWS CLI](#) Command Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * Asynchronously enables the specified key.
 *
 * @param keyId the ID of the key to enable
 * @return a {@link CompletableFuture} that completes when the key has been
 * enabled
 */
public CompletableFuture<Void> enableKeyAsync(String keyId) {
    EnableKeyRequest enableKeyRequest = EnableKeyRequest.builder()
        .keyId(keyId)
        .build();

    CompletableFuture<EnableKeyResponse> responseFuture =
        getAsyncClient().enableKey(enableKeyRequest);
    responseFuture.whenComplete((response, exception) -> {
        if (exception == null) {
            logger.info("Key with ID [{}] has been enabled.", keyId);
        } else {
            if (exception instanceof KmsException kmsEx) {
                throw new RuntimeException("KMS error occurred while enabling
key: " + kmsEx.getMessage(), kmsEx);
            } else {
                throw new RuntimeException("An unexpected error occurred
while enabling key: " + exception.getMessage(), exception);
            }
        }
    });

    return responseFuture.thenApply(response -> null);
}
```

- Per i dettagli sull'API, [EnableKey](#) consulta AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
suspend fun enableKey(keyIdVal: String?) {  
    val request =  
        EnableKeyRequest {  
            keyId = keyIdVal  
        }  
  
    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->  
        kmsClient.enableKey(request)  
        println("$keyIdVal was successfully enabled.")  
    }  
}
```

- Per i dettagli sull'API, [EnableKey](#) consulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
```

```

    * @param string $keyId
    * @return void
    */
    public function enableKey(string $keyId)
    {
        try {
            $this->client->enableKey([
                'KeyId' => $keyId,
            ]);
        } catch (KmsException $caught) {
            if ($caught->getAwsErrorMessage() == "NotFoundException") {
                echo "The request was rejected because the specified entity or
resource could not be found.\n";
            }
            throw $caught;
        }
    }
}

```

- Per i dettagli sull'API, [EnableKey](#) consulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

```

```

        :return: An instance of KeyManager initialized with the default KMS
client.
    """
    kms_client = boto3.client("kms")
    return cls(kms_client)

def enable_key(self, key_id: str) -> None:
    """
    Enables a key. Gets the key state after each state change.

    :param key_id: The ARN or ID of the key to enable.
    """
    try:
        self.kms_client.enable_key(KeyId=key_id)
    except ClientError as err:
        logging.error(
            "Couldn't enable key '%s'. Here's why: %s",
            key_id,
            err.response["Error"]["Message"],
        )
        raise

```

- Per i dettagli sull'API, consulta [EnableKey AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **EnableKeyRotation** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare EnableKeyRotation.

CLI

AWS CLI

Come abilitare la rotazione automatica di una chiave KMS

L'esempio `enable-key-rotation` seguente abilita la rotazione automatica di una chiave KMS gestita dal cliente con un periodo di rotazione di 180 giorni. La rotazione della chiave

KMS avviene dopo un anno (circa 365 giorni) dalla data di completamento di questo comando e successivamente ogni anno.

Il parametro `--key-id` identifica la chiave KMS. Questo esempio utilizza un valore ARN della chiave, ma è possibile utilizzare l'ID chiave o l'ARN della chiave KMS. Il parametro `--rotation-period-in-days` specifica il numero di giorni tra ogni data di rotazione. Specifica un valore compreso tra 90 e 2.560 giorni. Se non viene specificato alcun valore, il valore predefinito è 365 giorni.

```
aws kms enable-key-rotation \
  --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
  --rotation-period-in-days 180
```

Questo comando non produce alcun output. Per verificare se la chiave KMS è abilitata, utilizza il comando `get-key-rotation-status`.

Per ulteriori informazioni, consulta [Rotazione delle chiavi](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

- Per i dettagli sull'API, consulta [EnableKeyRotation AWS CLI](#) Command Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
```



```

    Creates a KeyManager instance with a default KMS client.

    :return: An instance of KeyManager initialized with the default KMS
client.
    """
    kms_client = boto3.client("kms")
    return cls(kms_client)

def enable_key_rotation(self, key_id: str) -> None:
    """
    Enables rotation for a key.

    :param key_id: The ARN or ID of the key to enable rotation for.
    """
    try:
        self.kms_client.enable_key_rotation(KeyId=key_id)
    except ClientError as err:
        logging.error(
            "Couldn't enable rotation for key '%s'. Here's why: %s",
            key_id,
            err.response["Error"]["Message"],
        )
        raise

```

- Per i dettagli sull'API, consulta [EnableKeyRotation AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un AWS SDK](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **Encrypt** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare Encrypt.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Informazioni di base](#)

CLI

AWS CLI

Esempio 1: come crittografare il contenuto di un file in Linux o macOS

Il `encrypt` comando seguente illustra il metodo consigliato per crittografare i dati con la AWS CLI.

```
aws kms encrypt \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --plaintext fileb://ExamplePlaintextFile \
  --output text \
  --query CiphertextBlob | base64 \
  --decode > ExampleEncryptedFile
```

Il comando esegue diverse operazioni:

Utilizza il parametro `--plaintext` per indicare i dati da crittografare. Il valore di questo parametro deve essere codificato in base 64. Il valore del `plaintext` parametro deve essere codificato in base 64 oppure è necessario utilizzare il prefisso `fileb://`, che indica alla CLI di leggere i dati binari dal file. Se il AWS file non si trova nella directory corrente, digitare il percorso completo del file. Ad esempio: `fileb:///var/tmp/ExamplePlaintextFile` o `fileb://C:\Temp\ExamplePlaintextFile`. [Per ulteriori informazioni sulla lettura dei valori dei parametri AWS CLI da un file, vedere Loading Parameters from a File nella AWS Command Line Interface User Guide e Best Practices for Local File Parameters nel blog AWS Command Line Tool. Utilizza `--query` i parametri `--output` and per controllare l'output del comando. Questi parametri estraggono i dati crittografati, chiamati `ciphertext`, dall'output del comando. Per ulteriori informazioni sul controllo dell'output, vedere \[Controllo del comando Output\]\(#\) nella Guida per l'utente dell'interfaccia a riga di AWS comando. Utilizza l'base64 utilità per decodificare l'output estratto in dati binari. Il testo cifrato restituito da un `encrypt` comando riuscito è testo con codifica in base64. È necessario decodificare questo testo prima di poter utilizzare la AWS CLI per decrittografarlo. Salva il testo cifrato binario in un file. La parte finale del comando `> ExampleEncryptedFile \(\)` salva il testo cifrato binario in un file per semplificare la decrittografia. Per un comando di esempio che utilizza la AWS CLI per decrittografare i dati, consulta gli esempi di decrittografia.](#)

Esempio 2: utilizzo della AWS CLI per crittografare i dati su Windows

Questo esempio è uguale a quello precedente, tranne per il fatto che utilizza lo strumento `certutil` al posto di `base64`. Questa procedura richiede due comandi, come illustrato nell'esempio seguente.

```
aws kms encrypt \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --plaintext fileb://ExamplePlaintextFile \  
  --output text \  
  --query CiphertextBlob > C:\Temp\ExampleEncryptedFile.base64  
  
certutil -decode C:\Temp\ExampleEncryptedFile.base64 C:\Temp\ExampleEncryptedFile
```

Esempio 3: crittografia con una chiave KMS asimmetrica

Il seguente comando `encrypt` mostra come crittografare il testo normale con una chiave KMS asimmetrica. Il parametro `--encryption-algorithm` è obbligatorio. Come in tutti i comandi `encrypt` CLI, il `plaintext` parametro deve essere codificato in base 64 oppure è necessario utilizzare il prefisso `fileb://`, che indica alla CLI di leggere i dati binari dal file AWS.

```
aws kms encrypt \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --encryption-algorithm RSAES_OAEP_SHA_256 \  
  --plaintext fileb://ExamplePlaintextFile \  
  --output text \  
  --query CiphertextBlob | base64 \  
  --decode > ExampleEncryptedFile
```

Questo comando non produce alcun output.

- Per informazioni dettagliate sull'API, consulta [Encrypt](#) in AWS CLI Command Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

/**
 * Encrypts the given text asynchronously using the specified KMS client and
 * key ID.
 *
 * @param keyId the ID of the KMS key to use for encryption
 * @param text the text to encrypt
 * @return a CompletableFuture that completes with the encrypted data as an
 * SdkBytes object
 */
public CompletableFuture<SdkBytes> encryptDataAsync(String keyId, String
text) {
    SdkBytes myBytes = SdkBytes.fromUtf8String(text);
    EncryptRequest encryptRequest = EncryptRequest.builder()
        .keyId(keyId)
        .plaintext(myBytes)
        .build();

    CompletableFuture<EncryptResponse> responseFuture =
getAsyncClient().encrypt(encryptRequest).toCompletableFuture();
    return responseFuture.whenComplete((response, ex) -> {
        if (response != null) {
            String algorithm = response.encryptionAlgorithm().toString();
            logger.info("The string was encrypted with algorithm {}.\"",
algorithm);
        } else {
            throw new RuntimeException(ex);
        }
    }).thenApply(EncryptResponse::ciphertextBlob);
}

```

- Per i dettagli sull'API, consulta [Encrypt](#) in AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
suspend fun encryptData(keyIdValue: String): ByteArray? {
    val text = "This is the text to encrypt by using the AWS KMS Service"
    val myBytes: ByteArray = text.toByteArray()

    val encryptRequest =
        EncryptRequest {
            keyId = keyIdValue
            plaintext = myBytes
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.encrypt(encryptRequest)
        val algorithm: String = response.encryptionAlgorithm.toString()
        println("The encryption algorithm is $algorithm")

        // Return the encrypted data.
        return response.ciphertextBlob
    }
}

suspend fun decryptData(
    encryptedDataVal: ByteArray?,
    keyIdVal: String?,
) {
    val decryptRequest =
        DecryptRequest {
            ciphertextBlob = encryptedDataVal
            keyId = keyIdVal
        }

    KmsClient { region = "us-west-2" }.use { kmsClient ->
        val decryptResponse = kmsClient.decrypt(decryptRequest)
        val myVal = decryptResponse.plaintext

        // Print the decrypted data.
        print(myVal)
    }
}
```

- Per i dettagli sull'API, [consulta Encrypt](#) in AWS SDK for Kotlin API reference.

PHP

SDK per PHP

Note

C'è di più su. [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * @param string $keyId
 * @param string $text
 * @return Result
 */
public function encrypt(string $keyId, string $text)
{
    try {
        return $this->client->encrypt([
            'KeyId' => $keyId,
            'Plaintext' => $text,
        ]);
    } catch (KmsException $caught){
        if($caught->getAwsErrorMessage() == "DisabledException"){
            echo "The request was rejected because the specified KMS key is
not enabled.\n";
        }
        throw $caught;
    }
}
```

- Per i dettagli sull'API, consulta [Encrypt](#) in AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def encrypt(self, key_id: str, text: str) -> bytes:
        """
        Encrypts text by using the specified key.

        :param key_id: The ARN or ID of the key to use for encryption.
        :param text: The text to encrypt.
        :return: The encrypted version of the text.
        """
        try:
            response = self.kms_client.encrypt(KeyId=key_id,
Plaintext=text.encode())
            print(
                f"The string was encrypted with algorithm
{response['EncryptionAlgorithm']}"
            )
            return response["CiphertextBlob"]
```

```
except ClientError as err:
    if err.response["Error"]["Code"] == "DisabledException":
        logger.error(
            "Could not encrypt because the key %s is disabled.", key_id
        )
    else:
        logger.error(
            "Couldn't encrypt text. Here's why: %s",
            err.response["Error"]["Message"],
        )
    raise
```

- Per i dettagli sull'API, [consulta Encrypt](#) in AWS SDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

Note

C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
require 'aws-sdk-kms' # v2: require 'aws-sdk'

# ARN of the AWS KMS key.
#
# Replace the fictitious key ARN with a valid key ID

keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

text = '1234567890'

client = Aws::KMS::Client.new(region: 'us-west-2')

resp = client.encrypt({
    key_id: keyId,
    plaintext: text
```



```

    })

    # Display a readable version of the resulting encrypted blob.
    puts 'Blob:'
    puts resp.ciphertext_blob.unpack('H*')

```

- Per i dettagli sull'API, consulta [Encrypt](#) in AWS SDK per Ruby API Reference.

Rust

SDK per Rust

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

async fn encrypt_string(
    verbose: bool,
    client: &Client,
    text: &str,
    key: &str,
    out_file: &str,
) -> Result<(), Error> {
    let blob = Blob::new(text.as_bytes());

    let resp = client.encrypt().key_id(key).plaintext(blob).send().await?;

    // Did we get an encrypted blob?
    let blob = resp.ciphertext_blob.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);

    let mut ofile = File::create(out_file).expect("unable to create file");
    ofile.write_all(s.as_bytes()).expect("unable to write");

    if verbose {
        println!("Wrote the following to {:?}", out_file);
        println!("{}", s);
    }
}

```

```
}  
  
    Ok(())  
}
```

- Per i dettagli sull'API, [consulta Encrypt](#) in AWS SDK for Rust API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un AWS SDK](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **GenerateDataKey** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `GenerateDataKey`.

CLI

AWS CLI

Esempio 1: come generare una chiave dati simmetrica a 256 bit

L'`generate-data-key` seguente richiede una chiave dati simmetrica a 256 bit da utilizzare all'esterno di AWS. Il comando restituisce una chiave dati in testo normale per l'utilizzo e l'eliminazione immediati e una copia di tale chiave dati crittografata sotto la chiave KMS specificata. Puoi archiviare la chiave di dati crittografata in modo sicuro con i dati crittografati.

Per richiedere una chiave dati a 256 bit, utilizza il parametro `key-spec` con il valore `AES_256`. Per richiedere una chiave dati a 128 bit, utilizza il parametro `key-spec` con il valore `AES_128`. Per tutte le altre lunghezze di chiavi dati, utilizza il parametro `number-of-bytes`.

La chiave KMS specificata deve essere una chiave KMS di crittografia simmetrica, ovvero una chiave KMS con un valore di `KeySpec` pari a `SYMMETRIC_DEFAULT`.

```
aws kms generate-data-key \  
    --key-id alias/ExampleAlias \  
    --key-spec AES_256
```

Output:

```
{
  "Plaintext": "VdzKNHGzUAzJeRBVY+uUmofUGGiDzyB3+i9fVkh3piw=",
  "KeyId": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyMaterialId":
    "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
  "CiphertextBlob":
    "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgk
    +YdhV8MrkBQPeac0ReRVNDt9qlAt+SHgIRF8P0H+7U="
}
```

I valori `Plaintext` (chiave dati in testo normale) `CiphertextBlob` e (chiave dati crittografata) vengono restituiti crittografati in formato Base64.

Per ulteriori informazioni, consulta [Chiavi di dati](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi. Esempio 2: come generare una chiave dati simmetrica a 512 bit

L'esempio `generate-data-key` seguente richiede una chiave dati simmetrica a 512 bit per la crittografia e la decrittografia. Il comando restituisce una chiave dati in testo normale per l'utilizzo e l'eliminazione immediati e una copia di tale chiave dati crittografata sotto la chiave KMS specificata. Puoi archiviare la chiave di dati crittografata in modo sicuro con i dati crittografati.

Per richiedere una lunghezza di chiave diversa da 128 o 256 bit, utilizza il parametro `number-of-bytes`. Per richiedere una chiave dati a 512 bit, l'esempio seguente utilizza il parametro `number-of-bytes` con il valore 64 (byte).

La chiave KMS specificata deve essere una chiave KMS di crittografia simmetrica, ovvero una chiave KMS con un valore di `KeySpec` pari a `SYMMETRIC_DEFAULT`.

NOTA: i valori nell'output di questo esempio vengono troncati ai fini della visualizzazione.

```
aws kms generate-data-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --number-of-bytes 64
```

Output:

```
{
  "CiphertextBlob": "AQIBAHi6LtupRpdKl2aJTzkkK6Fbh0tQkMlQJJH3PdtHvS/y+hAEnX/
  QQNmMwDfg2koirNMEc8AAACaDCCAmQGCSqGSib3DQEHbqCCA1UwggJRAgEAMIICSgYJKoZ...",
```

```

    "Plaintext": "ty8Lr0Bk60F07M2Bwt6qbFdNB
+G00ZLtf5MSEb4a13R2UKWG0p06njAwy2n72VRm2m7z/
Pm9Wpbvttz6a4lSo9hgPvKhZ5y6RTm40ovEXiVfBveyX3DQxDzRSwbKDPk/...",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
    "0b7fd7ddb6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}

```

I valori `Plaintext` (chiave di dati in testo normale) e `CiphertextBlob` (chiave di dati crittografati) vengono restituiti crittografati in formato Base64.

Per ulteriori informazioni, consulta [Chiavi di dati](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

- Per i dettagli sull'API, consulta [GenerateDataKey](#) Command Reference.AWS CLI

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

```

```

def generate_data_key(self, key_id):
    """
    Generates a symmetric data key that can be used for client-side
    encryption.
    """
    answer = input(
        f"Do you want to generate a symmetric data key from key {key_id} (y/n)? "
    )
    if answer.lower() == "y":
        try:
            data_key = self.kms_client.generate_data_key(
                KeyId=key_id, KeySpec="AES_256"
            )
        except ClientError as err:
            logger.error(
                "Couldn't generate a data key for key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
        else:
            pprint(data_key)

```

- Per i dettagli sull'API, consulta [GenerateDataKey AWS SDK for Python \(Boto3\) API Reference](#).

Rust

SDK per Rust

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

async fn make_key(client: &Client, key: &str) -> Result<(), Error> {

```

```
let resp = client
    .generate_data_key()
    .key_id(key)
    .key_spec(DataKeySpec::Aes256)
    .send()
    .await?;

// Did we get an encrypted blob?
let blob = resp.ciphertext_blob.expect("Could not get encrypted text");
let bytes = blob.as_ref();

let s = base64::encode(bytes);

println!();
println!("Data key:");
println!("{}", s);

Ok(())
}
```

- Per i dettagli sulle API, consulta la [GenerateDataKey](#) guida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **GenerateDataKeyWithoutPlaintext** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `GenerateDataKeyWithoutPlaintext`.

CLI

AWS CLI

Come generare una chiave dati simmetrica a 256 bit senza una chiave in testo normale

L'esempio `generate-data-key-without-plaintext` seguente richiede una copia crittografata di una chiave dati simmetrica a 256 bit da utilizzare all'esterno di AWS. Puoi chiamare AWS KMS per decrittografare la chiave dati quando sei pronto per usarla.

Per richiedere una chiave dati a 256 bit, utilizza il parametro `key-spec` con il valore `AES_256`.
Per richiedere una chiave dati a 128 bit, utilizza il parametro `key-spec` con il valore `AES_128`.
Per tutte le altre lunghezze di chiavi dati, utilizza il parametro `number-of-bytes`.

La chiave KMS specificata deve essere una chiave KMS di crittografia simmetrica, ovvero una chiave KMS con un valore di `KeySpec` pari a `SYMMETRIC_DEFAULT`.

```
aws kms generate-data-key-without-plaintext \
  --key-id "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab" \
  --key-spec AES_256
```

Output:

```
{
  "CiphertextBlob":
"AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgk
  "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyMaterialId":
"0b7fd7ddb6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

Il valore `CiphertextBlob` (chiave dati crittografati) viene restituito crittografato in formato Base64.

Per ulteriori informazioni, consulta [Chiavi di dati](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

- Per i dettagli sull'API, consulta Command [GenerateDataKeyWithoutPlaintext](#) Reference AWS CLI .

Rust

SDK per Rust

Note

C'è altro da fare. [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

async fn make_key(client: &Client, key: &str) -> Result<(), Error> {
    let resp = client
        .generate_data_key_without_plaintext()
        .key_id(key)
        .key_spec(DataKeySpec::Aes256)
        .send()
        .await?;

    // Did we get an encrypted blob?
    let blob = resp.ciphertext_blob.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);

    println!();
    println!("Data key:");
    println!("{}", s);

    Ok(())
}

```

- Per i dettagli sulle API, consulta la [GenerateDataKeyWithoutPlaintext](#) guida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **GenerateRandom** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare GenerateRandom.

CLI

AWS CLI

Esempio 1: come generare una stringa di byte casuale a 256 bit (Linux o macOS)

L'esempio `generate-random` seguente genera una stringa di byte casuali a 256 bit (32 byte) crittografata in formato Base64. L'esempio decodifica la stringa di byte e la salva in un file casuale.

Quando esegui questo comando, devi utilizzare il parametro `number-of-bytes` per specificare la lunghezza del valore casuale in byte.

Non viene specificata una chiave KMS quando si esegue questo comando. La stringa di byte casuali non è correlata ad alcuna chiave KMS.

Per impostazione predefinita, AWS KMS genera il numero casuale. Tuttavia, se si specifica un [key store personalizzato](#), la stringa di byte casuale viene generata nel cluster AWS CloudHSM associato all'archivio chiavi personalizzato.

Questo esempio utilizza i seguenti parametri e valori:

Utilizza il `--number-of-bytes` parametro richiesto con un valore di 32 per richiedere una stringa da 32 byte (256 bit). Utilizza il `--output` parametro con un valore di `text` per indirizzare la AWS CLI a restituire l'output come testo, anziché JSON. Utilizza il `--query Plaintext` per estrarre il valore della `Plaintext` proprietà da `Response`. it invia (`|`) l'output del comando `--query` parameter all'`base64` utilità, che decodifica l'output estratto. Utilizza l'operatore di reindirizzamento (`>`) per salvare la decodifica stringa di byte aggiunta al file. Utilizza l'operatore di reindirizzamento (`>`) `ExampleRandom` per salvare il testo cifrato binario in un file.

```
aws kms generate-random \
  --number-of-bytes 32 \
  --output text \
  --query Plaintext | base64 --decode > ExampleRandom
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [GenerateRandom](#) nel AWS Key Management Service API Reference.

Esempio 2: come generare un numero casuale a 256 bit (prompt dei comandi di Windows)

L'esempio seguente utilizza il comando `generate-random` per generare una stringa di byte casuali a 256 bit (32 byte) crittografata in formato Base64. L'esempio decodifica la stringa di byte e la salva in un file casuale. Questo esempio è uguale all'esempio precedente tranne per il fatto che utilizza l'utilità `certutil` di Windows per decrittografare in formato Base64 la stringa di byte casuali prima di salvarla in un file.

Genera innanzitutto una stringa di byte casuali crittografata in formato Base64 e la salva nel file temporaneo `ExampleRandom.base64`.

```
aws kms generate-random \  
  --number-of-bytes 32 \  
  --output text \  
  --query Plaintext > ExampleRandom.base64
```

Poiché l'output del comando `generate-random` viene salvato in un file, questo esempio non produce alcun output.

Ora utilizza il comando `certutil -decode` per decrittografare la stringa di byte crittografata in formato Base64 nel file `ExampleRandom.base64`. Quindi, salva la stringa di byte decodificata nel file `ExampleRandom`.

```
certutil -decode ExampleRandom.base64 ExampleRandom
```

Output:

```
Input Length = 18  
Output Length = 12  
CertUtil: -decode command completed successfully.
```

Per ulteriori informazioni, vedere [GenerateRandom](#) nel AWS Key Management Service API Reference.

- Per i dettagli sull'API, consulta [GenerateRandom AWS CLI Command Reference](#).

Rust

SDK per Rust

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
async fn make_string(client: &Client, length: i32) -> Result<(), Error> {  
    let resp = client  
        .generate_random()  
        .number_of_bytes(length)  
        .send()
```

```

        .await?;

    // Did we get an encrypted blob?
    let blob = resp.plaintext.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);

    println!();
    println!("Data key:");
    println!("{}", s);

    Ok(())
}

```

- Per i dettagli sulle API, consulta la [GenerateRandom](#) guida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **GetKeyPolicy** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare GetKeyPolicy.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Informazioni di base](#)

CLI

AWS CLI

Come copiare una policy della chiave da una chiave KMS a un'altra

L'esempio `get-key-policy` seguente ottiene la policy della chiave da una chiave KMS e la salva in un file di testo. Sostituisce quindi la policy di una chiave KMS diversa utilizzando il file di testo come input della policy.

Poiché il parametro `--policy` di `put-key-policy` richiede una stringa, devi utilizzare l'opzione `--output text` per restituire l'output come stringa di testo anziché JSON.

```
aws kms get-key-policy \
  --policy-name default \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --query Policy \
  --output text > policy.txt

aws kms put-key-policy \
  --policy-name default \
  --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
  --policy file://policy.txt
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [PutKeyPolicy](#) la sezione AWS KMS API Reference.

- Per i dettagli sull'API, consulta [GetKeyPolicy AWS CLI Command Reference](#).

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
class KeyPolicy:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyPolicy":
        """
        Creates a KeyPolicy instance with a default KMS client.

        :return: An instance of KeyPolicy initialized with the default KMS
        client.
```

```
"""
kms_client = boto3.client("kms")
return cls(kms_client)

def get_policy(self, key_id: str) -> dict[str, str]:
    """
    Gets the policy of a key.

    :param key_id: The ARN or ID of the key to query.
    :return: The key policy as a dict.
    """
    if key_id != "":
        try:
            response = self.kms_client.get_key_policy(
                KeyId=key_id,
            )
            policy = json.loads(response["Policy"])
        except ClientError as err:
            logger.error(
                "Couldn't get policy for key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
        else:
            pprint(policy)
            return policy
    else:
        print("Skipping get policy demo.")
```

- Per i dettagli sull'API, consulta [GetKeyPolicy AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **ListAliases** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `ListAliases`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Informazioni di base](#)

.NET

SDK per .NET

 Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// List the AWS Key Management Service (AWS KMS) aliases that have been
defined for
/// the keys in the same AWS Region as the default user. If you want to list
/// the aliases in a different Region, pass the Region to the client
/// constructor.
/// </summary>
public class ListAliases
{
    public static async Task Main()
    {
        var client = new AmazonKeyManagementServiceClient();
        var request = new ListAliasesRequest();
        var response = new ListAliasesResponse();

        do
        {
            response = await client.ListAliasesAsync(request);

            response.Aliases.ForEach(alias =>
            {
```

```
        Console.WriteLine($"Created: {alias.CreationDate} Last  
Update: {alias.LastUpdatedDate} Name: {alias.AliasName}");  
    });  
  
    request.Marker = response.NextMarker;  
    }  
    while (response.Truncated);  
    }  
}
```

- Per i dettagli sull'API, [ListAliases](#) consulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Esempio 1: per elencare tutti gli alias in un AWS account e in una regione

L'esempio seguente utilizza il `list-aliases` comando per elencare tutti gli alias nella regione predefinita dell' AWS account. L'output include alias associati alle chiavi KMS AWS gestite e alle chiavi KMS gestite dal cliente.

```
aws kms list-aliases
```

Output:

```
{  
  "Aliases": [  
    {  
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/testKey",  
      "AliasName": "alias/testKey",  
      "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"  
    },  
    {  
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/FinanceDept",  
      "AliasName": "alias/FinanceDept",  
      "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321"  
    },  
    {  
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/dynamodb",  
      "AliasName": "alias/aws/dynamodb",  
      "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"  
    }  
  ]  
}
```

```

        "AliasName": "alias/aws/dynamodb",
        "TargetKeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
    },
    {
        "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/ebs",
        "AliasName": "alias/aws/ebs",
        "TargetKeyId": "0987ab65-43cd-21ef-09ab-87654321cdef"
    },
    ...
]
}

```

Esempio 2: come elencare tutti gli alias per una particolare chiave KMS

L'esempio seguente utilizza il comando `list-aliases` e il relativo parametro `key-id` per elencare tutti gli alias associati a una particolare chiave KMS.

Ogni alias è associato a una sola chiave KMS, ma una chiave KMS può avere più alias. Questo comando è molto utile perché la console AWS KMS elenca solo un alias per ogni chiave KMS. Per trovare tutti gli alias di una chiave KMS, devi utilizzare il comando `list-aliases`.

Questo esempio utilizza l'ID della chiave KMS per il parametro `--key-id`, ma in questo comando puoi utilizzare un ID chiave, un ARN di chiave, un nome di alias o un ARN di alias.

```
aws kms list-aliases --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Output:

```

{
  "Aliases": [
    {
      "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/oregon-test-key",
      "AliasName": "alias/oregon-test-key"
    },
    {
      "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/project121-test",
      "AliasName": "alias/project121-test"
    }
  ]
}

```



```
    }
  ]
}
```

Per ulteriori informazioni, consulta [Utilizzo degli alias](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

- Per i dettagli sull'API, consulta Command [ListAliases](#)Reference AWS CLI .

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * Asynchronously lists all the aliases in the current AWS account.
 *
 * @return a {@link CompletableFuture} that completes when the list of
 * aliases has been processed
 */
public CompletableFuture<Object> listAllAliasesAsync() {
    ListAliasesRequest aliasesRequest = ListAliasesRequest.builder()
        .limit(15)
        .build();

    ListAliasesPublisher paginator =
getAsyncClient().listAliasesPaginator(aliasesRequest);
    return paginator.subscribe(response -> {
        response.aliases().forEach(alias ->
            logger.info("The alias name is: " + alias.aliasName())
        );
    })
        .thenApply(v -> null)
        .exceptionally(ex -> {
            if (ex.getCause() instanceof KmsException) {
                KmsException e = (KmsException) ex.getCause();
```

```
        throw new RuntimeException("A KMS exception occurred: " +
e.getMessage());
    } else {
        throw new RuntimeException("An unexpected error occurred: " +
ex.getMessage());
    }
    });
}
```

- Per i dettagli sull'API, [ListAliases](#) consulta AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
suspend fun listAllAliases() {
    val request =
        ListAliasesRequest {
            limit = 15
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.listAliases(request)
        response.aliases?.forEach { alias ->
            println("The alias name is ${alias.aliasName}")
        }
    }
}
```

- Per i dettagli sull'API, [ListAliases](#) consulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

 Note

C'è di più su. [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * @param string $keyId
 * @param int $limit
 * @return ResultPaginator
 */
public function listAliases(string $keyId = "", int $limit = 0)
{
    $args = [];
    if($keyId){
        $args['KeyId'] = $keyId;
    }
    if($limit){
        $args['Limit'] = $limit;
    }
    try{
        return $this->client->getPaginator("ListAliases", $args);
    }catch(KmsException $caught){
        if($caught->getAwsErrorMessage() == "InvalidMarkerException"){
            echo "The request was rejected because the marker that specifies
where pagination should next begin is not valid.\n";
        }
        throw $caught;
    }
}
```

- Per i dettagli sull'API, [ListAliases](#) consulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def list_aliases(self, page_size: int) -> None:
        """
        Lists aliases for the current account.
        :param page_size: The number of aliases to list per page.
        """
        try:
            alias_paginator = self.kms_client.get_paginator("list_aliases")
            for alias_page in alias_paginator.paginate(
                PaginationConfig={"PageSize": page_size}
            ):
                print(f"Here are {page_size} aliases:")
                pprint(alias_page["Aliases"])
                if alias_page["Truncated"]:
                    answer = input(
```

```
        f"Do you want to see the next {page_size} aliases (y/n)?\n\n        )\n        if answer.lower() != "y":\n            break\n        else:\n            print("That's all your aliases!")\nexcept ClientError as err:\n    logging.error(\n        "Couldn't list your aliases. Here's why: %s",\n        err.response["Error"]["Message"],\n    )\n    raise
```

- Per i dettagli sull'API, consulta [ListAliases AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **ListGrants** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `ListGrants`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Informazioni di base](#)

.NET

SDK per .NET

Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// List the AWS Key Management Service (AWS KMS) grants that are associated
with
/// a specific key.
/// </summary>
public class ListGrants
{
    public static async Task Main()
    {
        // The identifier of the AWS KMS key to disable. You can use the
        // key Id or the Amazon Resource Name (ARN) of the AWS KMS key.
        var keyId = "1234abcd-12ab-34cd-56ef-1234567890ab";
        var client = new AmazonKeyManagementServiceClient();
        var request = new ListGrantsRequest
        {
            KeyId = keyId,
        };

        var response = new ListGrantsResponse();

        do
        {
            response = await client.ListGrantsAsync(request);

            response.Grants.ForEach(grant =>
            {
                Console.WriteLine($"{grant.GrantId}");
            });

            request.Marker = response.NextMarker;
        }
        while (response.Truncated);
    }
}
```

- Per i dettagli sull'API, [ListGrants](#) consulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Per visualizzare le sovvenzioni su una chiave AWS KMS

L'`list-grants` esempio seguente mostra tutte le concessioni sulla chiave KMS AWS gestita specificata per Amazon DynamoDB nel tuo account. Questa concessione consente a DynamoDB di utilizzare la chiave KMS per tuo conto per crittografare una tabella DynamoDB prima di scriverla su disco. Puoi utilizzare un comando come questo per visualizzare le concessioni relative alle chiavi KMS gestite e alle chiavi KMS AWS gestite dal cliente nell'account e nella regione. AWS

Questo comando utilizza il parametro `key-id` con un ID chiave per identificare la chiave KMS. Puoi utilizzare un ID o un ARN di chiave per identificare la chiave KMS. Per ottenere l'ID o l'ARN della chiave di una chiave KMS AWS gestita, usa il `list-keys` comando o `list-aliases`

```
aws kms list-grants \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

L'output mostra che la concessione assegna l'autorizzazione Amazon DynamoDB per utilizzare la chiave KMS per le operazioni di crittografia e visualizzare i dettagli sulla chiave KMS (`DescribeKey`) e di ritirare le concessioni (`RetireGrant`). Il vincolo `EncryptionContextSubset` limita queste autorizzazioni alle richieste che includono le coppie di contesti di crittografia specificate. Di conseguenza, le autorizzazioni incluse nella concessione sono valide solo per l'account e la tabella DynamoDB specificati.

```
{
  "Grants": [
    {
      "Constraints": {
        "EncryptionContextSubset": {
          "aws:dynamodb:subscriberId": "123456789012",
          "aws:dynamodb:tableName": "Services"
        }
      },
      "IssuingAccount": "arn:aws:iam::123456789012:root",
      "Name": "8276b9a6-6cf0-46f1-b2f0-7993a7f8c89a",
      "Operations": [
        "Decrypt",
```

```

        "Encrypt",
        "GenerateDataKey",
        "ReEncryptFrom",
        "ReEncryptTo",
        "RetireGrant",
        "DescribeKey"
    ],
    "GrantId":
    "1667b97d27cf748cf05b487217dd4179526c949d14fb3903858e25193253fe59",
    "KeyId": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "RetiringPrincipal": "dynamodb.us-west-2.amazonaws.com",
    "GranteePrincipal": "dynamodb.us-west-2.amazonaws.com",
    "CreationDate": "2021-05-13T18:32:45.144000+00:00"
    }
}

```

Per ulteriori informazioni, consulta [Grants in AWS KMS nella AWS Key Management Service Developer Guide](#).

- Per i dettagli sull'API, consulta AWS CLI Command [ListGrants](#)Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

/**
 * Asynchronously displays the grant IDs for the specified key ID.
 *
 * @param keyId the ID of the AWS KMS key for which to list the grants
 * @return a {@link CompletableFuture} that, when completed, will be null
 * if the operation succeeded, or will throw a {@link RuntimeException} if the
 * operation failed
 * @throws RuntimeException if there was an error listing the grants, either
 * due to an {@link KmsException} or an unexpected error

```



```

    */
    public CompletableFuture<Object> displayGrantIdsAsync(String keyId) {
        ListGrantsRequest grantsRequest = ListGrantsRequest.builder()
            .keyId(keyId)
            .limit(15)
            .build();

        ListGrantsPublisher paginator =
            getAsyncClient().listGrantsPaginator(grantsRequest);
        return paginator.subscribe(response -> {
            response.grants().forEach(grant -> {
                logger.info("The grant Id is: " + grant.grantId());
            });
        })
        .thenApply(v -> null)
        .exceptionally(ex -> {
            Throwable cause = ex.getCause();
            if (cause instanceof KmsException) {
                throw new RuntimeException("Failed to list grants: " +
                    cause.getMessage(), cause);
            } else {
                throw new RuntimeException("An unexpected error occurred: " +
                    cause.getMessage(), cause);
            }
        });
    }
}

```

- Per i dettagli sull'API, [ListGrants](#) consulta AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

suspend fun displayGrantIds(keyIdVal: String?) {
    val request =

```

```

        ListGrantsRequest {
            keyId = keyIdVal
            limit = 15
        }

        KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
            val response = kmsClient.listGrants(request)
            response.grants?.forEach { grant ->
                println("The grant Id is ${grant.grantId}")
            }
        }
    }
}

```

- Per i dettagli sull'API, [ListGrants](#) consulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

Note

C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

/**
 * @param string $keyId
 * @return Result
 */
public function listGrants(string $keyId)
{
    try{
        return $this->client->listGrants([
            'KeyId' => $keyId,
        ]);
    }catch(KmsException $caught){
        if($caught->getAwsErrorMessage() == "NotFoundException"){
            echo "    The request was rejected because the specified entity
or resource could not be found.\n";
        }
    }
}

```

```
        throw $caught;
    }
}
```

- Per i dettagli sull'API, [ListGrants](#) consulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def list_grants(self, key_id):
        """
        Lists grants for a key.

        :param key_id: The ARN or ID of the key to query.
        :return: The grants for the key.
        """
        try:
```

```
paginator = self.kms_client.get_paginator("list_grants")
grants = []
page_iterator = paginator.paginate(KeyId=key_id)
for page in page_iterator:
    grants.extend(page["Grants"])

print(f"Grants for key {key_id}:")
pprint(grants)
return grants
except ClientError as err:
    logger.error(
        "Couldn't list grants for key %s. Here's why: %s",
        key_id,
        err.response["Error"]["Message"],
    )
    raise
```

- Per i dettagli sull'API, consulta [ListGrants AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **ListKeyPolicies** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `ListKeyPolicies`.

CLI

AWS CLI

Come ottenere i nomi delle policy della chiave per una chiave KMS

L'esempio `list-key-policies` seguente ottiene i nomi delle policy della chiave per una chiave gestita dal cliente nell'account e nella Regione di esempio. È possibile utilizzare questo comando per trovare i nomi delle politiche chiave per le chiavi AWS gestite e le chiavi gestite dai clienti.

Poiché l'unico nome di policy della chiave valido è `default`, questo comando non è utile.

Per specificare la chiave KMS, utilizza il parametro `key-id`. Questo esempio utilizza un valore per l'ID chiave, ma in questo comando è possibile utilizzare un ID o un ARN di chiave.

```
aws kms list-key-policies \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Output:

```
{
  "PolicyNames": [
    "default"
  ]
}
```

Per ulteriori informazioni sulle politiche chiave AWS KMS, consulta [Using Key Policies in AWS KMS nella AWS Key Management Service Developer Guide](#).

- Per i dettagli sulle API, consulta AWS CLI Command [ListKeyPolicies](#) Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * Asynchronously retrieves the key policy for the specified key ID and
 * policy name.
 *
 * @param keyId      the ID of the AWS KMS key for which to retrieve the
 * policy
 * @param policyName the name of the key policy to retrieve
 * @return a {@link CompletableFuture} that, when completed, contains the key
 * policy as a {@link String}
 */
public CompletableFuture<String> getKeyPolicyAsync(String keyId, String
policyName) {
```

```

    GetKeyPolicyRequest policyRequest = GetKeyPolicyRequest.builder()
        .keyId(keyId)
        .policyName(policyName)
        .build();

    return getAsyncClient().getKeyPolicy(policyRequest)
        .thenApply(response -> {
            String policy = response.policy();
            logger.info("The response is: " + policy);
            return policy;
        })
        .exceptionally(ex -> {
            throw new RuntimeException("Failed to get key policy", ex);
        });
}

```

- Per i dettagli sull'API, [ListKeyPolicies](#) consulta AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

class KeyPolicy:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyPolicy":
        """
        Creates a KeyPolicy instance with a default KMS client.

        :return: An instance of KeyPolicy initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")

```

```
        return cls(kms_client)

    def list_policies(self, key_id):
        """
        Lists the names of the policies for a key.

        :param key_id: The ARN or ID of the key to query.
        """
        try:
            policy_names = self.kms_client.list_key_policies(KeyId=key_id)[
                "PolicyNames"
            ]
        except ClientError as err:
            logging.error(
                "Couldn't list your policies. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise
        else:
            print(f"The policies for key {key_id} are:")
            pprint(policy_names)
```

- Per i dettagli sull'API, consulta [ListKeyPolicies AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un AWS SDK](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **ListKeys** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `ListKeys`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Informazioni di base](#)

.NET

SDK per .NET

Note

C'è altro su GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// List the AWS Key Managements Service (AWS KMS) keys for the AWS Region
/// of the default user. To list keys in another AWS Region, supply the
Region
/// as a parameter to the client constructor.
/// </summary>
public class ListKeys
{
    public static async Task Main()
    {
        var client = new AmazonKeyManagementServiceClient();
        var request = new ListKeysRequest();
        var response = new ListKeysResponse();

        do
        {
            response = await client.ListKeysAsync(request);

            response.Keys.ForEach(key =>
            {
                Console.WriteLine($"ID: {key.KeyId}, {key.KeyArn}");
            });

            // Set the Marker property when response.Truncated is true
            // in order to get the next keys.
            request.Marker = response.NextMarker;
        }
    }
}
```



```
        while (response.Truncated);  
    }  
}
```

- Per i dettagli sull'API, [ListKeys](#) consulta AWS SDK per .NET API Reference.

CLI

AWS CLI

Come ottenere le chiavi KMS in un account e in una Regione

L'esempio `list-keys` seguente ottiene le chiavi KMS in un account e in una Regione. Questo comando restituisce sia le chiavi AWS gestite che le chiavi gestite dal cliente.

```
aws kms list-keys
```

Output:

```
{  
  "Keys": [  
    {  
      "KeyArn": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
      "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"  
    },  
    {  
      "KeyArn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",  
      "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321"  
    },  
    {  
      "KeyArn": "arn:aws:kms:us-east-2:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",  
      "KeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Visualizzazione di chiavi](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

- Per i dettagli sull'API, consulta [ListKeys AWS CLI](#) Command Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
import software.amazon.awssdk.services.kms.KmsAsyncClient;
import software.amazon.awssdk.services.kms.model.ListKeysRequest;
import software.amazon.awssdk.services.kms.paginators.ListKeysPublisher;
import java.util.concurrent.CompletableFuture;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class HelloKMS {
    public static void main(String[] args) {
        listAllKeys();
    }

    public static void listAllKeys() {
        KmsAsyncClient kmsAsyncClient = KmsAsyncClient.builder()
            .build();
        ListKeysRequest listKeysRequest = ListKeysRequest.builder()
            .limit(15)
            .build();

        /*
```

```

    * The `subscribe` method is required when using paginator methods in the
    AWS SDK
    * because paginator methods return an instance of a `ListKeysPublisher`,
    which is
    * based on a reactive stream. This allows asynchronous retrieval of
    paginated
    * results as they become available. By subscribing to the stream, we can
    process
    * each page of results as they are emitted.
    */
    ListKeysPublisher keysPublisher =
kmsAsyncClient.listKeysPaginator(listKeysRequest);
    CompletableFuture<Void> future = keysPublisher
        .subscribe(r -> r.keys().forEach(key ->
            System.out.println("The key ARN is: " + key.keyArn() + ". The key
Id is: " + key.keyId()))
        .whenComplete((result, exception) -> {
            if (exception != null) {
                System.err.println("Error occurred: " +
exception.getMessage());
            } else {
                System.out.println("Successfully listed all keys.");
            }
        });

    try {
        future.join();
    } catch (Exception e) {
        System.err.println("Failed to list keys: " + e.getMessage());
    }
}
}

```

- Per i dettagli sull'API, [ListKeys](#) consulta AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
suspend fun listAllKeys() {
    val request =
        ListKeysRequest {
            limit = 15
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.listKeys(request)
        response.keys?.forEach { key ->
            println("The key ARN is ${key.keyArn}")
            println("The key Id is ${key.keyId}")
        }
    }
}
```

- Per i dettagli sull'API, [ListKeys](#) consulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
```

```

    * @return array
    */
    public function listKeys()
    {
        try {
            $contents = [];
            $paginator = $this->client->getPaginator("ListKeys");
            foreach($paginator as $result){
                foreach ($result['Content'] as $object) {
                    $contents[] = $object;
                }
            }
            return $contents;
        }catch(KmsException $caught){
            echo "There was a problem listing the keys: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
}

```

- Per i dettagli sull'API, [ListKeys](#) consulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """

```

```

    Creates a KeyManager instance with a default KMS client.

    :return: An instance of KeyManager initialized with the default KMS
client.
    """
    kms_client = boto3.client("kms")
    return cls(kms_client)

def list_keys(self):
    """
    Lists the keys for the current account by using a paginator.
    """
    try:
        page_size = 10
        print("\nLet's list your keys.")
        key_paginator = self.kms_client.get_paginator("list_keys")
        for key_page in key_paginator.paginate(PaginationConfig={"PageSize":
10}):
            print(f"Here are {len(key_page['Keys'])} keys:")
            pprint(key_page["Keys"])
            if key_page["Truncated"]:
                answer = input(
                    f"Do you want to see the next {page_size} keys (y/n)? "
                )
                if answer.lower() != "y":
                    break
            else:
                print("That's all your keys!")
    except ClientError as err:
        logging.error(
            "Couldn't list your keys. Here's why: %s",
            err.response["Error"]["Message"],
        )

```

- Per i dettagli sull'API, consulta [ListKeys AWS SDK for Python \(Boto3\) API Reference](#).

Rust

SDK per Rust

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
async fn show_keys(client: &Client) -> Result<(), Error> {
    let resp = client.list_keys().send().await?;

    let keys = resp.keys.unwrap_or_default();

    let len = keys.len();

    for key in keys {
        println!("Key ARN: {}", key.key_arn.as_deref().unwrap_or_default());
    }

    println!();
    println!("Found {} keys", len);

    Ok(())
}
```

- Per i dettagli sulle API, consulta la [ListKeys](#) guida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **PutKeyPolicy** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare PutKeyPolicy.

CLI

AWS CLI

Come modificare la policy della chiave per una chiave KMS

L'esempio `put-key-policy` seguente modifica la policy della chiave per una chiave gestita dal cliente.

Per iniziare, crea una policy della chiave e salvala in un file JSON locale. In questo esempio, il file è `key_policy.json`. Puoi anche specificare la policy della chiave come valore di stringa del parametro `policy`.

La prima dichiarazione di questa policy chiave fornisce all' AWS account il permesso di utilizzare le policy IAM per controllare l'accesso alla chiave KMS. La seconda istruzione concede all'utente `test-user` l'autorizzazione a eseguire i comandi `describe-key` e `list-keys` sulla chiave KMS.

Contenuto di `key_policy.json`:

```
{
  "Version": "2012-10-17",
  "Id" : "key-default-1",
  "Statement" : [
    {
      "Sid" : "Enable IAM User Permissions",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "arn:aws:iam::111122223333:root"
      },
      "Action" : "kms:*",
      "Resource" : "*"
    },
    {
      "Sid" : "Allow Use of Key",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "arn:aws:iam::111122223333:user/test-user"
      },
      "Action" : [
        "kms:DescribeKey",
        "kms:ListKeys"
      ]
    }
  ]
}
```



```

        "Resource" : "*"
    }
]
}

```

Per identificare la chiave KMS, questo esempio utilizza l'ID chiave, ma puoi utilizzare anche un ARN di chiave. Per specificare la policy della chiave, il comando utilizza il parametro `policy`. Per indicare che la policy è contenuta in un file, utilizza il prefisso obbligatorio `file://`. Questo prefisso è necessario per identificare i file in tutti i sistemi operativi supportati. Infine, il comando utilizza il parametro `policy-name` con il valore `default`. Se nessun nome di policy è specificato, il valore predefinito è `default`. L'unico valore valido è `default`.

```

aws kms put-key-policy \
  --policy-name default \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --policy file://key_policy.json

```

Il comando non produce output. Per verificare che il comando sia stato eseguito correttamente, utilizza il comando `get-key-policy`. Il comando di esempio seguente ottiene la policy della chiave per la stessa chiave KMS. Il parametro `output` con il valore `text` restituisce un formato di testo facile da leggere.

```

aws kms get-key-policy \
  --policy-name default \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --output text

```

Output:

```

{
  "Version": "2012-10-17",
  "Id" : "key-default-1",
  "Statement" : [
    {
      "Sid" : "Enable IAM User Permissions",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "arn:aws:iam::111122223333:root"
      },
      "Action" : "kms:*",
      "Resource" : "*"
    }
  ]
}

```

```

    },
    {
      "Sid" : "Allow Use of Key",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "arn:aws:iam::111122223333:user/test-user"
      },
      "Action" : [ "kms:Describe", "kms:List" ],
      "Resource" : "*"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Modifica di una policy delle chiavi](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

- Per i dettagli sull'API, consulta [PutKeyPolicy AWS CLI Command Reference](#).

PHP

SDK per PHP

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

/**
 * @param string $keyId
 * @param string $policy
 * @return void
 */
public function putKeyPolicy(string $keyId, string $policy)
{
    try {
        $this->client->putKeyPolicy([
            'KeyId' => $keyId,
            'Policy' => $policy,
        ]);
    } catch (KmsException $caught){

```

```

        echo "There was a problem replacing the key policy: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}

```

- Per i dettagli sull'API, [PutKeyPolicy](#) consulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

class KeyPolicy:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyPolicy":
        """
        Creates a KeyPolicy instance with a default KMS client.

        :return: An instance of KeyPolicy initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def set_policy(self, key_id: str, policy: dict[str, any]) -> None:
        """
        Sets the policy of a key. Setting a policy entirely overwrites the
        existing
        policy, so care is taken to add a statement to the existing list of
        statements

```

```

rather than simply writing a new policy.

:param key_id: The ARN or ID of the key to set the policy to.
:param policy: The existing policy of the key.
:return: None
"""
principal = input(
    "Enter the ARN of an IAM role to set as the principal on the policy:
"
)
if key_id != "" and principal != "":
    # The updated policy replaces the existing policy. Add a new
statement to
    # the list along with the original policy statements.
    policy["Statement"].append(
        {
            "Sid": "Allow access for ExampleRole",
            "Effect": "Allow",
            "Principal": {"AWS": principal},
            "Action": [
                "kms:Encrypt",
                "kms:GenerateDataKey*",
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:ReEncrypt*",
            ],
            "Resource": "*",
        }
    )
    try:
        self.kms_client.put_key_policy(KeyId=key_id,
Policy=json.dumps(policy))
    except ClientError as err:
        logger.error(
            "Couldn't set policy for key %s. Here's why %s",
            key_id,
            err.response["Error"]["Message"],
        )
        raise
    else:
        print(f"Set policy for key {key_id}.")
else:
    print("Skipping set policy demo.")

```

- Per i dettagli sull'API, consulta [PutKeyPolicy AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **ReEncrypt** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare ReEncrypt.

CLI

AWS CLI

Esempio 1: come ripetere la crittografia di un messaggio crittografato con una chiave KMS simmetrica diversa (Linux e macOS).

L'esempio di `re-encrypt` comando seguente mostra il metodo consigliato per ricrittografare i dati con la CLI. AWS

Fornisci il testo criptato in un file. Nel valore del parametro `--ciphertext-blob`, utilizza il prefisso `fileb://` per indicare alla CLI di leggere i dati da un file binario. Se il file non è presente nella directory corrente, digita il percorso completo del file. Per ulteriori informazioni sulla lettura dei valori dei parametri AWS CLI da un file, consulta Caricamento dei [parametri AWS CLI da un file nella Guida per l'utente dell'interfaccia a riga di AWS comando e Best Practices for Local File Parameters](#) nel Command Line Tool Blog. Specificare la AWS chiave KMS di origine, che decrittografa il CipherText. Il parametro non è richiesto quando si decrittografa con chiavi KMS di crittografia simmetrica. `--source-key-id` AWS KMS può ottenere la chiave KMS utilizzata per crittografare i dati dai metadati nel blob di testo cifrato. Tuttavia è sempre consigliabile specificare la chiave KMS che stai utilizzando. Questa procedura garantisce di utilizzare la chiave KMS desiderata e impedisce di decrittografare inavvertitamente un testo criptato utilizzando una chiave KMS non attendibile. Specifica la chiave KMS di destinazione, che ripete la crittografia dei dati. Il parametro `--destination-key-id` è sempre obbligatorio. Questo esempio utilizza un ARN di chiave, ma puoi utilizzare qualsiasi identificatore di chiave valido. Richiedi l'output in testo normale come valore di testo. Il parametro `--query` indica alla CLI di recuperare solo il valore del campo Plaintext dall'output. Il parametro `--output` restituisce l'output come testo. Procedi a decrittografare

in formato Base64 il testo normale e a salvarlo in un file. Nell'esempio seguente il valore del parametro Plaintext viene inviato (|) all'utilità Base64, che lo decodifica. Quindi, reindirizza (>) l'output decodificato al file ExamplePlaintext.

Prima di eseguire questo comando, sostituisci la chiave di esempio IDs con identificatori di chiave validi del tuo account. AWS

```
aws kms re-encrypt \
  --ciphertext-blob fileb://ExampleEncryptedFile \
  --source-key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --destination-key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
  --query CiphertextBlob \
  --output text | base64 --decode > ExampleReEncryptedFile
```

Questo comando non produce alcun output. L'output del comando re-encrypt viene decrittografato in formato Base64 e salvato in un file.

Per ulteriori informazioni, consulta [ReEncrypt](#) la sezione AWS Key Management Service API Reference.

Esempio 2: come crittografare di nuovo un messaggio crittografato con una chiave KMS simmetrica diversa (prompt dei comandi di Windows).

Il seguente comando re-encrypt di esempio è lo stesso del precedente, tranne per il fatto che utilizza l'utilità certutil per decrittografare in formato Base64 i dati in testo normale. Questa procedura richiede due comandi, come illustrato negli esempi seguenti.

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID chiave valido del tuo AWS account.

```
aws kms re-encrypt ^
  --ciphertext-blob fileb://ExampleEncryptedFile ^
  --source-key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^
  --destination-key-id 0987dcba-09fe-87dc-65ba-ab0987654321 ^
  --query CiphertextBlob ^
  --output text > ExampleReEncryptedFile.base64
```

Utilizza quindi l'utilità certutil

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Output:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

Per ulteriori informazioni, consulta la [ReEncrypt](#) sezione AWS Key Management Service API Reference.

- Per i dettagli sull'API, consulta [ReEncrypt AWS CLI](#) Command Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def re_encrypt(self, source_key_id, cipher_text):
        """
        Takes ciphertext previously encrypted with one key and reencrypt it by
        using
```

```

another key.

:param source_key_id: The ARN or ID of the original key used to encrypt
the
                    ciphertext.
:param cipher_text: The encrypted ciphertext.
:return: The ciphertext encrypted by the second key.
"""
destination_key_id = input(
    f"Your ciphertext is currently encrypted with key {source_key_id}. "
    f"Enter another key ID or ARN to reencrypt it: "
)
if destination_key_id != "":
    try:
        cipher_text = self.kms_client.re_encrypt(
            SourceKeyId=source_key_id,
            DestinationKeyId=destination_key_id,
            CiphertextBlob=cipher_text,
        )["CiphertextBlob"]
    except ClientError as err:
        logger.error(
            "Couldn't reencrypt your ciphertext. Here's why: %s",
            err.response["Error"]["Message"],
        )
    else:
        print(f"Reencrypted your ciphertext as: {cipher_text}")
        return cipher_text
else:
    print("Skipping reencryption demo.")

```

- Per i dettagli sull'API, consulta [ReEncrypt AWS SDK for Python \(Boto3\) API Reference](#).

Ruby

SDK per Ruby

Note

C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).


```
require 'aws-sdk-kms' # v2: require 'aws-sdk'

# Human-readable version of the ciphertext of the data to reencrypt.

blob =
  '01020200785d68faeec386af1057904926253051eb2919d3c16078badf65b808b26dd057c101747cadf3593'
sourceCiphertextBlob = [blob].pack('H*')

# Replace the fictitious key ARN with a valid key ID

destinationKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321'

client = Aws::KMS::Client.new(region: 'us-west-2')

resp = client.re_encrypt({
                        ciphertext_blob: sourceCiphertextBlob,
                        destination_key_id: destinationKeyId
                      })

# Display a readable version of the resulting re-encrypted blob.
puts 'Blob:'
puts resp.ciphertext_blob.unpack('H*')
```

- Per i dettagli sull'API, [ReEncrypt](#) consulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
async fn reencrypt_string(
    verbose: bool,
    client: &Client,
```

```

    input_file: &str,
    output_file: &str,
    first_key: &str,
    new_key: &str,
) -> Result<(), Error> {
    // Get blob from input file
    // Open input text file and get contents as a string
    // input is a base-64 encoded string, so decode it:
    let data = fs::read_to_string(input_file)
        .map(|input_file| base64::decode(input_file).expect("invalid base 64"))
        .map(Blob::new);

    let resp = client
        .re_encrypt()
        .ciphertext_blob(data.unwrap())
        .source_key_id(first_key)
        .destination_key_id(new_key)
        .send()
        .await?;

    // Did we get an encrypted blob?
    let blob = resp.ciphertext_blob.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);
    let o = &output_file;

    let mut ofile = File::create(o).expect("unable to create file");
    ofile.write_all(s.as_bytes()).expect("unable to write");

    if verbose {
        println!("Wrote the following to {}: ", output_file);
        println!("{}", s);
    } else {
        println!("Wrote base64-encoded output to {}", output_file);
    }

    Ok(())
}

```

- Per i dettagli sulle API, consulta la [ReEncrypt](#) guida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un AWS SDK](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **RetireGrant** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare RetireGrant.

CLI

AWS CLI

Come ritirare una concessione associata a una chiave principale del cliente

L'esempio retire-grant seguente elimina una concessione da una chiave KMS.

Il comando di esempio seguente specifica i parametri grant-id e key-id. Il valore del parametro key-id deve essere l'ARN della chiave KMS.

```
aws kms retire-grant \  
  --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \  
  --key-id arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non produce alcun output. Per confermare il ritiro della concessione, utilizza il comando list-grants.

Per ulteriori informazioni, consulta [Ritirare e revocare le concessioni](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

- Per i dettagli sull'API, consulta [RetireGrant AWS CLI](#) Command Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def retire_grant(self, grant):
        """
        Retires a grant so that it can no longer be used.

        :param grant: The grant to retire.
        """
        try:
            self.kms_client.retire_grant(GrantToken=grant["GrantToken"])
        except ClientError as err:
            logger.error(
                "Couldn't retire grant %s. Here's why: %s",
                grant["GrantId"],
                err.response["Error"]["Message"],
            )
        else:
            print(f"Grant {grant['GrantId']} retired.")

```

- Per i dettagli sull'API, consulta [RetireGrant AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un AWS SDK](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **RevokeGrant** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare RevokeGrant.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Informazioni di base](#)

CLI

AWS CLI

Come revocare una concessione associata a una chiave principale del cliente

L'esempio `revoke-grant` seguente elimina una concessione da una chiave KMS. Il comando di esempio seguente specifica i parametri `grant-id` e `key-id`. Il valore del parametro `key-id` può essere l'ID o l'ARN della chiave KMS.

```
aws kms revoke-grant \
  --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non produce alcun output. Per confermare la revoca della concessione, utilizza il comando `list-grants`.

Per ulteriori informazioni, consulta [Ritirare e revocare le concessioni](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

- Per i dettagli sull'API, consulta [RevokeGrant AWS CLI Command Reference](#).

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

/**
 * Revokes a grant for the specified AWS KMS key asynchronously.
 *
 * @param keyId    The ID or key ARN of the AWS KMS key.
 * @param grantId  The identifier of the grant to be revoked.
 * @return A {@link CompletableFuture} representing the asynchronous
operation of revoking the grant.
 *
 *      The {@link CompletableFuture} will complete with a {@link
RevokeGrantResponse} object
 *
 *      if the operation is successful, or with a {@code null} value if an
error occurs.
 */
public CompletableFuture<RevokeGrantResponse> revokeKeyGrantAsync(String
keyId, String grantId) {
    RevokeGrantRequest grantRequest = RevokeGrantRequest.builder()
        .keyId(keyId)
        .grantId(grantId)
        .build();

    CompletableFuture<RevokeGrantResponse> responseFuture =
getAsyncClient().revokeGrant(grantRequest);
    responseFuture.whenComplete((response, exception) -> {
        if (exception == null) {
            logger.info("Grant ID: [" + grantId + "] was successfully
revoked!");
        } else {
            if (exception instanceof KmsException kmsEx) {
                if (kmsEx.getMessage().contains("Grant does not exist")) {
                    logger.info("The grant ID '" + grantId + "' does not
exist. Moving on...");
                } else {
                    throw new RuntimeException("KMS error occurred: " +
kmsEx.getMessage(), kmsEx);
                }
            } else {
                throw new RuntimeException("An unexpected error occurred: " +
exception.getMessage(), exception);
            }
        }
    });

    return responseFuture;
}

```

- Per i dettagli sull'API, [RevokeGrant](#) consulta AWS SDK for Java 2.x API Reference.

PHP

SDK per PHP

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * @param string $grantId
 * @param string $keyId
 * @return void
 */
public function revokeGrant(string $grantId, string $keyId)
{
    try{
        $this->client->revokeGrant([
            'GrantId' => $grantId,
            'KeyId' => $keyId,
        ]);
    }catch(KmsException $caught){
        echo "There was a problem with revoking the grant: {$caught-
>getAwsErrorMessage()}.\\n";
        throw $caught;
    }
}
```

- Per i dettagli sull'API, [RevokeGrant](#) consulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def revoke_grant(self, key_id: str, grant_id: str) -> None:
        """
        Revokes a grant so that it can no longer be used.

        :param key_id: The ARN or ID of the key associated with the grant.
        :param grant_id: The ID of the grant to revoke.
        """
        try:
            self.kms_client.revoke_grant(KeyId=key_id, GrantId=grant_id)
        except ClientError as err:
            logger.error(
                "Couldn't revoke grant %s. Here's why: %s",
                grant_id,
                err.response["Error"]["Message"],
            )
            raise
```


- Per i dettagli sull'API, consulta [RevokeGrant AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un AWS SDK](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **ScheduleKeyDeletion** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `ScheduleKeyDeletion`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Informazioni di base](#)

CLI

AWS CLI

Come pianificare l'eliminazione di una chiave KMS gestita dal cliente.

L'esempio `schedule-key-deletion` seguente pianifica l'eliminazione della chiave KMS gestita dal cliente specificata tra 15 giorni.

Il parametro `--key-id` identifica la chiave KMS. Questo esempio utilizza il valore di un ARN di chiave, ma puoi utilizzare l'ID o l'ARN della chiave KMS. Il parametro `--pending-window-in-days` specifica la durata del periodo di attesa (da 7 a 30 giorni). Per impostazione predefinita, il periodo di attesa è pari a 30 giorni. Questo esempio specifica il valore 15, che indica di AWS eliminare definitivamente la chiave KMS 15 giorni dopo il completamento del comando.

```
aws kms schedule-key-deletion \
    --key-id arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --pending-window-in-days 15
```

La risposta include l'ARN della chiave, lo stato della chiave, il periodo di attesa (PendingWindowInDays) e la data di eliminazione in formato Unix. Per visualizzare la data di eliminazione nell'ora locale, usa la AWS console KMS. Le chiavi KMS con stato PendingDeletion non possono essere utilizzate nelle operazioni di crittografia.

```
{
  "KeyId": "arn:aws:kms:us-west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "DeletionDate": "2022-06-18T23:43:51.272000+00:00",
  "KeyState": "PendingDeletion",
  "PendingWindowInDays": 15
}
```

Per ulteriori informazioni, consulta [Eliminazione delle chiavi](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

- Per i dettagli sull'API, consulta [ScheduleKeyDeletion AWS CLI Command Reference](#).

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * Deletes a KMS key asynchronously.
 *
 * <p><strong>Warning:</strong> Deleting a KMS key is a destructive and
 * potentially dangerous operation.
 * When a KMS key is deleted, all data that was encrypted under the KMS key
 * becomes unrecoverable.
 * This means that any files, databases, or other data that were encrypted
 * using the deleted KMS key
 * will become permanently inaccessible. Exercise extreme caution when
 * deleting KMS keys.</p>
 *
 * @param keyId the ID of the KMS key to delete
```

```

    * @return a {@link CompletableFuture} that completes when the key deletion
    is scheduled
    */
    public CompletableFuture<Void> deleteKeyAsync(String keyId) {
        ScheduleKeyDeletionRequest deletionRequest =
        ScheduleKeyDeletionRequest.builder()
            .keyId(keyId)
            .pendingWindowInDays(7)
            .build();

        return getAsyncClient().scheduleKeyDeletion(deletionRequest)
            .thenRun(() -> {
                logger.info("Key {} will be deleted in 7 days", keyId);
            })
            .exceptionally(throwable -> {
                throw new RuntimeException("Failed to schedule key deletion for
key ID: " + keyId, throwable);
            });
    }

```

- Per i dettagli sull'API, [ScheduleKeyDeletion](#) consulta AWS SDK for Java 2.x API Reference.

PHP

SDK per PHP

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

/**
 * @param string $keyId
 * @param int $pendingWindowInDays
 * @return void
 */
public function scheduleKeyDeletion(string $keyId, int $pendingWindowInDays =
7)
{

```

```

    try {
        $this->client->scheduleKeyDeletion([
            'KeyId' => $keyId,
            'PendingWindowInDays' => $pendingWindowInDays,
        ]);
    } catch (KmsException $caught){
        echo "There was a problem scheduling the key deletion: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}

```

- Per i dettagli sull'API, [ScheduleKeyDeletion](#) consulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
        client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

```

```

def delete_key(self, key_id: str, window: int) -> None:
    """
    Deletes a list of keys.

    Warning:
    Deleting a KMS key is a destructive and potentially dangerous operation.
    When a KMS key is deleted,
    all data that was encrypted under the KMS key is unrecoverable.

    :param key_id: The ARN or ID of the key to delete.
    :param window: The waiting period, in days, before the KMS key is
    deleted.
    """

    try:
        self.kms_client.schedule_key_deletion(
            KeyId=key_id, PendingWindowInDays=window
        )
    except ClientError as err:
        logging.error(
            "Couldn't delete key %s. Here's why: %s",
            key_id,
            err.response["Error"]["Message"],
        )
        raise

```

- Per i dettagli sull'API, consulta [ScheduleKeyDeletion AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **Sign** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare Sign.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Informazioni di base](#)

CLI

AWS CLI

Esempio 1: come generare una firma digitale per un messaggio

L'esempio `sign` seguente genera una firma crittografica per un messaggio breve. L'output del comando include un campo `Signature` crittografato in formato Base64 che è possibile verificare utilizzando il comando `verify`.

È necessario specificare un messaggio da firmare e un algoritmo di firma supportato dalla chiave KMS asimmetrica. Per ottenere gli algoritmi di firma per la chiave KMS, utilizza il comando `describe-key`.

In AWS CLI v2, il valore del `message` parametro deve essere codificato in Base64. In alternativa, puoi salvare il messaggio in un file e utilizzare il `fileb://` prefisso, che indica alla AWS CLI di leggere i dati binari dal file.

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID chiave valido del tuo AWS account. L'ID chiave deve rappresentare una chiave KMS asimmetrica il cui utilizzo è definito da `SIGN_VERIFY`.

```
msg=(echo 'Hello World' | base64)

aws kms sign \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --message fileb://UnsignedMessage \
  --message-type RAW \
  --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256
```

Output:

```
{
  "KeyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Signature": "ABCDEFhpyVYyTxbaFE74ccSvEJLJr3zuoV1Hfymz4qv+/fmxNLA7SE1SiF8lHw80fKZZ3bJ...",
  "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
```

```
}
```

Per ulteriori informazioni sull'utilizzo delle chiavi KMS asimmetriche in KMS, consulta [Chiavi asimmetriche in AWS KMS nella AWS Key Management Service Developer Guide](#).AWS

Esempio 2: come salvare una firma digitale in un file (Linux e macOS)

L'esempio `sign` seguente genera una firma crittografica per un messaggio breve archiviato in un file locale. Il comando ottiene anche la `Signature` proprietà dalla risposta, Base64-la decodifica e la salva nel file. `ExampleSignature` Puoi utilizzare il file di firma in un comando `verify` che verifica la firma.

Il comando `sign` richiede un messaggio crittografato in formato Base64 e un algoritmo di firma supportato dalla chiave KMS asimmetrica. Per ottenere gli algoritmi di firma supportati dalla chiave KMS, utilizza il comando `describe-key`.

Prima di eseguire questo comando, sostituite l'ID della chiave di esempio con un ID chiave valido del vostro account. AWS L'ID chiave deve rappresentare una chiave KMS asimmetrica il cui utilizzo è definito da `SIGN_VERIFY`.

```
echo 'hello world' | base64 > EncodedMessage

aws kms sign \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --message fileb://EncodedMessage \
  --message-type RAW \
  --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256 \
  --output text \
  --query Signature | base64 --decode > ExampleSignature
```

Questo comando non produce alcun output. Questo esempio estrae la proprietà `Signature` dell'output e la salva in un file.

Per ulteriori informazioni sull'utilizzo delle chiavi KMS asimmetriche in KMS, consulta [Chiavi asimmetriche in AWS KMS nella AWS Key Management Service Developer Guide](#).AWS

- Per informazioni dettagliate sull'API, consulta [Sign](#) in AWS CLI Command Reference.

Java

SDK per Java 2.x

 Note

GitHubC'è di più su. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * Asynchronously signs and verifies data using AWS KMS.
 *
 * <p>The method performs the following steps:
 * <ol>
 *     <li>Creates an AWS KMS key with the specified key spec, key usage, and
origin.</li>
 *     <li>Signs the provided message using the created KMS key and the
RSASSA-PSS-SHA-256 algorithm.</li>
 *     <li>Verifies the signature of the message using the created KMS key
and the RSASSA-PSS-SHA-256 algorithm.</li>
 * </ol>
 *
 * @return a {@link CompletableFuture} that completes with the result of the
signature verification,
 *         {@code true} if the signature is valid, {@code false} otherwise.
 * @throws KmsException if any error occurs during the KMS operations.
 * @throws RuntimeException if an unexpected error occurs.
 */
public CompletableFuture<Boolean> signVerifyDataAsync() {
    String signMessage = "Here is the message that will be digitally signed";

    // Create an AWS KMS key used to digitally sign data.
    CreateKeyRequest createKeyRequest = CreateKeyRequest.builder()
        .keySpec(KeySpec.RSA_2048)
        .keyUsage(KeyUsageType.SIGN_VERIFY)
        .origin(OriginType.AWS_KMS)
        .build();

    return getAsyncClient().createKey(createKeyRequest)
        .thenCompose(createKeyResponse -> {
            String keyId = createKeyResponse.keyMetadata().keyId();
```



```

        SdkBytes messageBytes = SdkBytes.fromString(signMessage,
Charset.defaultCharset());
        SignRequest signRequest = SignRequest.builder()
            .keyId(keyId)
            .message(messageBytes)
            .signingAlgorithm(SigningAlgorithmSpec.RSASSA_PSS_SHA_256)
            .build();

        return getAsyncClient().sign(signRequest)
            .thenCompose(signResponse -> {
                byte[] signedBytes =
signResponse.signature().asByteArray();

                VerifyRequest verifyRequest = VerifyRequest.builder()
                    .keyId(keyId)

.message(SdkBytes.fromByteArray(signMessage.getBytes(Charset.defaultCharset()))))

.signature(SdkBytes.fromByteBuffer(ByteBuffer.wrap(signedBytes)))

.signingAlgorithm(SigningAlgorithmSpec.RSASSA_PSS_SHA_256)
                    .build();

                return getAsyncClient().verify(verifyRequest)
                    .thenApply(verifyResponse -> {
                        return (boolean) verifyResponse.signatureValid();
                    });
            });
    })
    .exceptionally(throwable -> {
        throw new RuntimeException("Failed to sign or verify data",
throwable);
    });
}

```

- Per i dettagli sull'API, consulta [Sign](#) in AWS SDK for Java 2.x API Reference.

PHP

SDK per PHP

 Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * @param string $keyId
 * @param string $message
 * @param string $algorithm
 * @return Result
 */
public function sign(string $keyId, string $message, string $algorithm)
{
    try {
        return $this->client->sign([
            'KeyId' => $keyId,
            'Message' => $message,
            'SigningAlgorithm' => $algorithm,
        ]);
    } catch (KmsException $caught) {
        echo "There was a problem signing the data: {$caught-
>getAwsErrorMessage()}\n";
        throw $caught;
    }
}
```

- Per i dettagli sull'API, consulta [Sign](#) in AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def sign(self, key_id: str, message: str) -> str:
        """
        Signs a message with a key.

        :param key_id: The ARN or ID of the key to use for signing.
        :param message: The message to sign.
        :return: The signature of the message.
        """
        try:
            return self.kms_client.sign(
                KeyId=key_id,
                Message=message.encode(),
                SigningAlgorithm="RSASSA_PSS_SHA_256",
            )["Signature"]
        except ClientError as err:
            logger.error(
```

```
        "Couldn't sign your message. Here's why: %s",
        err.response["Error"]["Message"],
    )
    raise
```

- Per i dettagli sull'API, consulta [Sign](#) in AWS SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un AWS SDK](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **TagResource** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare TagResource.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Informazioni di base](#)

CLI

AWS CLI

Come aggiungere un tag a una chiave KMS

L'esempio tag-resource seguente aggiunge i tag "Purpose": "Test" e "Dept": "IT" a una chiave KMS gestita dal cliente. Puoi utilizzare tag simili a quelli riportati di seguito per etichettare le chiavi KMS e creare categorie di chiavi KMS per le autorizzazioni e il controllo.

Per specificare la chiave KMS, utilizza il parametro key-id. Questo esempio utilizza un valore per l'ID chiave, ma in questo comando è possibile utilizzare un ID o un ARN di chiave.

```
aws kms tag-resource \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --tags TagKey='Purpose',TagValue='Test' TagKey='Dept',TagValue='IT'
```

Questo comando non produce alcun output. Per visualizzare i tag su una chiave AWS KMS, usa il comando. list-resource-tags

Per ulteriori informazioni sull'utilizzo dei tag in AWS KMS, consulta [Tagging keys](#) nella AWS Key Management Service Developer Guide.

- Per i dettagli sull'API, consulta AWS CLI Command [TagResource](#) Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * Asynchronously tags a KMS key with a specific tag.
 *
 * @param keyId the ID of the KMS key to be tagged
 * @return a {@link CompletableFuture} that completes when the tagging
 * operation is finished
 */
public CompletableFuture<Void> tagKMSKeyAsync(String keyId) {
    Tag tag = Tag.builder()
        .tagKey("Environment")
        .tagValue("Production")
        .build();

    TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
        .keyId(keyId)
        .tags(tag)
        .build();

    return getAsyncClient().tagResource(tagResourceRequest)
        .thenRun(() -> {
            logger.info("{} key was tagged", keyId);
        })
        .exceptionally(throwable -> {
            throw new RuntimeException("Failed to tag the KMS key",
throwable);
        });
}
```

- Per i dettagli sull'API, [TagResource](#) consulta AWS SDK for Java 2.x API Reference.

PHP

SDK per PHP

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * @param string $keyId
 * @param array $tags
 * @return void
 */
public function tagResource(string $keyId, array $tags)
{
    try {
        $this->client->tagResource([
            'KeyId' => $keyId,
            'Tags' => $tags,
        ]);
    } catch (KmsException $caught) {
        echo "There was a problem applying the tag(s): {"$caught->getAwsErrorMessage()}\n";
        throw $caught;
    }
}
```

- Per i dettagli sull'API, [TagResource](#) consulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def tag_resource(self, key_id: str, tag_key: str, tag_value: str) -> None:
        """
        Add or edit tags on a customer managed key.

        :param key_id: The ARN or ID of the key to enable rotation for.
        :param tag_key: Key for the tag.
        :param tag_value: Value for the tag.
        """
        try:
            self.kms_client.tag_resource(
                KeyId=key_id, Tags=[{"TagKey": tag_key, "TagValue": tag_value}]
            )
        except ClientError as err:
            logging.error(
                "Couldn't add a tag for the key '%s'. Here's why: %s",

```

```
        key_id,  
        err.response["Error"]["Message"],  
    )  
    raise
```

- Per i dettagli sull'API, consulta [TagResource AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **UpdateAlias** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare UpdateAlias.

CLI

AWS CLI

Come associare un alias a una chiave KMS diversa

L'esempio update-alias seguente associa l'alias alias/test-key a una chiave KMS diversa.

Il parametro `--alias-name` specifica l'alias. Il valore del nome dell'alias deve iniziare con `alias/`. Il parametro `--target-key-id` specifica la chiave KMS da associare all'alias. Non è necessario specificare la chiave KMS corrente per l'alias.

```
aws kms update-alias \  
    --alias-name alias/test-key \  
    --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non produce alcun output. Per cercare l'alias, utilizza il comando `list-aliases`.

Per ulteriori informazioni, consulta [Aggiornamento degli alias](#) nella Guida per gli sviluppatori del Servizio AWS di gestione delle chiavi.

- Per i dettagli sull'API, consulta [UpdateAlias AWS CLI Command Reference](#).

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)

    def update_alias(self, alias, current_key_id):
        """
        Updates an alias by assigning it to another key.

        :param alias: The alias to reassign.
        :param current_key_id: The ARN or ID of the key currently associated with
the alias.
        """
        new_key_id = input(
            f"Alias {alias} is currently associated with {current_key_id}. "
            f"Enter another key ID or ARN that you want to associate with
{alias}: "
        )
        if new_key_id != "":
            try:
```

```
        self.kms_client.update_alias(AliasName=alias,
        TargetKeyId=new_key_id)
    except ClientError as err:
        logger.error(
            "Couldn't associate alias %s with key %s. Here's why: %s",
            alias,
            new_key_id,
            err.response["Error"]["Message"],
        )
    else:
        print(f"Alias {alias} is now associated with key {new_key_id}.")
else:
    print("Skipping alias update.")
```

- Per i dettagli sull'API, consulta [UpdateAlias AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **Verify** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare **Verify**.

CLI

AWS CLI

Come verificare una firma digitale

Il seguente comando `verify` verifica una firma crittografica per un breve messaggio crittografato in formato Base64. L'ID chiave, il messaggio, il tipo di messaggio e l'algoritmo di firma devono essere gli stessi utilizzati per firmare il messaggio.

In AWS CLI v2, il valore del `message` parametro deve essere codificato in Base64. In alternativa, puoi salvare il messaggio in un file e utilizzare il `fileb://` prefisso, che indica alla AWS CLI di leggere i dati binari dal file.

La firma specificata non può essere crittografata in formato Base64. Per informazioni sulla decrittografia della firma restituita dal comando `sign`, consulta gli esempi del comando `sign`.

L'output del comando include un campo booleano `SignatureValid` che indica che la firma è stata verificata. Se la convalida della firma ha esito negativo, anche il comando `verify` ha esito negativo.

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID chiave valido del tuo AWS account.

```
aws kms verify \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --message fileb://EncodedMessage \
  --message-type RAW \
  --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256 \
  --signature fileb://ExampleSignature
```

Output:

```
{
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "SignatureValid": true,
  "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

Per ulteriori informazioni sull'utilizzo delle chiavi KMS asimmetriche in AWS KMS, consulta [Using asymmetric keys](#) nella Key Management Service Developer Guide.AWS

- Per informazioni dettagliate sull'API, consulta [Verify](#) in AWS CLI Command Reference.

Python

SDK per Python (Boto3)

Note

GitHubC'è di più su. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client
```

```
@classmethod
def from_client(cls) -> "KeyEncrypt":
    """
    Creates a KeyEncrypt instance with a default KMS client.

    :return: An instance of KeyEncrypt initialized with the default KMS
client.
    """
    kms_client = boto3.client("kms")
    return cls(kms_client)

def verify(self, key_id: str, message: str, signature: str) -> bool:
    """
    Verifies a signature against a message.

    :param key_id: The ARN or ID of the key used to sign the message.
    :param message: The message to verify.
    :param signature: The signature to verify.
    :return: True when the signature matches the message, otherwise False.
    """
    try:
        response = self.kms_client.verify(
            KeyId=key_id,
            Message=message.encode(),
            Signature=signature,
            SigningAlgorithm="RSASSA_PSS_SHA_256",
        )
        valid = response["SignatureValid"]
        print(f"The signature is {'valid' if valid else 'invalid'}.")
        return valid
    except ClientError as err:
        if err.response["Error"]["Code"] == "SignatureDoesNotMatchException":
            print("The signature is not valid.")
        else:
            logger.error(
                "Couldn't verify your signature. Here's why: %s",
                err.response["Error"]["Message"],
            )
        raise
```

- Per i dettagli sull'API, consulta [Verify](#) in AWS SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scenari di utilizzo AWS KMSAWS SDKs

I seguenti esempi di codice mostrano come implementare scenari comuni in AWS KMS with AWS SDKs. Questi scenari mostrano come eseguire attività specifiche richiamando più funzioni all'interno AWS KMS o combinandole con altre Servizi AWS. Ogni scenario include un collegamento al codice sorgente completo, dove è possibile trovare le istruzioni su come configurare ed eseguire il codice.

Gli scenari sono relativi a un livello intermedio di esperienza per aiutarti a comprendere le azioni di servizio nel contesto.

Esempi

- [Lavora con la crittografia delle tabelle DynamoDB utilizzando v2 AWS Command Line Interface](#)

Lavora con la crittografia delle tabelle DynamoDB utilizzando v2 AWS Command Line Interface

L'esempio di codice seguente mostra come gestire le opzioni di crittografia per le tabelle DynamoDB.

- Creazione di una tabella con la crittografia predefinita.
- Creazione di una tabella con una chiave gestita dal cliente (CMK).
- Aggiornamento delle impostazioni di crittografia delle tabelle.
- Descrizione della crittografia delle tabelle.

Bash

AWS CLI con lo script Bash

Creazione di una tabella con la crittografia predefinita.

```
# Create a table with default encryption (AWS owned key)
```

```
aws dynamodb create-table \  
  --table-name CustomerData \  
  --attribute-definitions \  
    AttributeName=CustomerID,AttributeType=S \  
  --key-schema \  
    AttributeName=CustomerID,KeyType=HASH \  
  --billing-mode PAY_PER_REQUEST \  
  --sse-specification Enabled=true,SSEType=KMS
```

Creazione di una tabella con una chiave gestita dal cliente (CMK).

```
# Step 1: Create a customer managed key in KMS  
aws kms create-key \  
  --description "Key for DynamoDB table encryption" \  
  --key-usage ENCRYPT_DECRYPT \  
  --customer-master-key-spec SYMMETRIC_DEFAULT  
  
# Store the key ID for later use  
KEY_ID=$(aws kms list-keys --query "Keys[?contains(KeyArn, 'Key for  
DynamoDB')].KeyId" --output text)  
  
# Step 2: Create a table with the customer managed key  
aws dynamodb create-table \  
  --table-name SensitiveData \  
  --attribute-definitions \  
    AttributeName=RecordID,AttributeType=S \  
  --key-schema \  
    AttributeName=RecordID,KeyType=HASH \  
  --billing-mode PAY_PER_REQUEST \  
  --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

Aggiornamento della crittografia delle tabelle.

```
# Update a table to use a different KMS key  
aws dynamodb update-table \  
  --table-name CustomerData \  
  --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

Descrizione della crittografia delle tabelle.

```
# Describe the table to see encryption settings
aws dynamodb describe-table \
  --table-name CustomerData \
  --query "Table.SSEDescription"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento dei comandi AWS CLI .
 - [CreateKey](#)
 - [CreateTable](#)
 - [DescribeTable](#)
 - [UpdateTable](#)

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta.

[Utilizzo di questo servizio con un AWS SDK](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Supporto per l'attestazione crittografica in AWS KMS

AWS KMS [supporta l'attestazione crittografica per Nitro Enclaves e NitroTPM](#). Le applicazioni che supportano questi metodi di attestazione richiamano le seguenti operazioni AWS KMS crittografiche con un documento di attestazione firmato. AWS KMS verifica che il documento di attestazione provenga da una fonte valida (un'enclave Nitro o un NitroTPM). Quindi, invece di restituire dati in chiaro nella risposta, APIs crittografano il testo in chiaro con la chiave pubblica del documento di attestazione e restituiscono testo cifrato che può essere decrittografato solo dalla chiave privata corrispondente nell'enclave o nell'istanza. EC2

- [Decrypt](#)
- [DeriveSharedSecret](#)
- [GenerateDataKey](#)
- [GenerateDataKeyPair](#)
- [GenerateRandom](#)

La tabella seguente mostra in che modo la risposta alle richieste attestate differisce dalla risposta standard per ogni operazione API.

AWS KMS operazione	Risposta standard	Risposta per richieste attestate
Decrypt	Restituisce dati in testo normale	Restituisce i dati di testo normale crittografati dalla chiave pubblica dal documento di attestazione
DeriveSharedSecret	Restituisce un segreto condiviso non elaborato	Restituisce il segreto condiviso non elaborato crittografato dalla chiave pubblica del documento di attestazione
GenerateDataKey	Restituisce una copia in testo normale della chiave dati	Restituisce una copia della chiave dati crittografata dalla chiave pubblica dal documento di attestazione

AWS KMS operazione	Risposta standard	Risposta per richieste attestate
	(Restituisce anche una copia della chiave dati crittografata da una chiave KMS)	(Restituisce anche una copia della chiave dati crittografata da una chiave KMS)
<code>GenerateDataKeyPair</code>	Restituisce una copia in testo normale della chiave privata (Restituisce anche la chiave pubblica e una copia della chiave privata crittografata da una chiave KMS)	Restituisce una copia della chiave privata crittografata dalla chiave pubblica dal documento di attestazione (Restituisce anche la chiave pubblica e una copia della chiave privata crittografata da una chiave KMS)
<code>GenerateRandom</code>	Restituisce una stringa di byte casuali	Restituisce la stringa di byte casuali crittografata dalla chiave pubblica dal documento di attestazione

AWS KMS supporta [le chiavi relative alle condizioni delle policy](#) che è possibile utilizzare per consentire o negare operazioni attestate con una AWS KMS chiave basata sul contenuto del documento di attestazione. Puoi anche [monitorare le richieste attestate](#) nei tuoi registri. AWS KMS
AWS CloudTrail

Ulteriori informazioni

- [Attestazione crittografica](#)
- [AWS KMS chiavi di condizione per piattaforme attestate](#)
- [Come effettuare chiamate certificate a AWS KMS](#)
- [Monitoraggio delle richieste attestate](#)

Come effettuare chiamate certificate a AWS KMS

Per effettuare una chiamata attestata a AWS KMS, utilizzate il `Recipient` parametro nella richiesta per fornire il documento di attestazione firmato e l'algoritmo di crittografia da utilizzare con la chiave pubblica nel documento di attestazione. Quando una richiesta include il parametro `Recipient` con un documento di attestazione firmato, la risposta include un campo `CiphertextForRecipient` con il testo criptato crittografato dalla chiave pubblica. Il campo di testo normale è nullo o vuoto.

Il `Recipient` parametro deve specificare un documento di attestazione firmato da un Nitro Enclaves o AWS NitroTPM. AWS KMS si affida alla firma digitale del documento di attestazione per dimostrare che la chiave pubblica nella richiesta proviene da una fonte valida. Non è possibile fornire il proprio certificato per firmare digitalmente il documento di attestazione.

L'SDK AWS Nitro Enclaves, supportato solo all'interno di un'enclave Nitro, aggiunge automaticamente il parametro e i relativi valori a ogni richiesta. `Recipient` AWS KMS

Per effettuare richieste attestate in AWS SDKs, devi specificare il parametro e i relativi valori.

`Recipient` [Il documento di attestazione può essere recuperato da NitroTPM utilizzando l'nitro-tpm-attest utilità o dal Nitro Secure Module \(NSM\) utilizzando l'API NSM.](#)

AWS KMS supporta [le chiavi delle condizioni delle policy](#) che è possibile utilizzare per consentire o negare operazioni attestate con una AWS KMS chiave basata sul contenuto del documento di attestazione. Puoi anche [monitorare le richieste attestate](#) nei tuoi registri. AWS KMS AWS CloudTrail

Per informazioni dettagliate sul `Recipient` parametro e sul campo di `CiphertextForRecipient` risposta AWS, [consulta Decrypt](#), [DeriveSharedSecret](#), e [GenerateRandom](#) gli argomenti nell'AWS Key Management Service API Reference [GenerateDataKeyGenerateDataKeyPair](#), nell'SDK [AWS Nitro Enclaves o in qualsiasi SDK](#). AWS [Per informazioni sulla configurazione dei dati e delle chiavi dati per la crittografia, consulta Usare l'attestazione crittografica con. AWS KMS](#)

Monitoraggio delle richieste attestate

Puoi utilizzare i tuoi AWS CloudTrail log per monitorare [Decrypt](#), [DeriveSharedSecretGenerateDataKeyGenerateDataKeyPair](#), e [GenerateRandom](#) le operazioni che utilizzano l'attestazione. In queste voci di registro, il `additionalEventData` campo contiene un `recipient` campo con le informazioni del documento di attestazione contenuto nella richiesta. Questi campi sono inclusi solo quando il `Recipient` parametro nella richiesta specifica un documento di attestazione firmato.

Le informazioni specifiche incluse nel CloudTrail registro dipendono dal metodo di attestazione utilizzato.

Richieste di monitoraggio per enclavi Nitro

Per l'attestazione dell'enclave di Nitro, il CloudTrail registro include l'ID del modulo (attestationDocumentModuleId), l'immagine digest () e i registri di configurazione della piattaforma (attestationDocumentEnclaveImageDigest) contenuti nel documento di attestazione. PCRs

L'ID del modulo è l'[ID enclave](#) dell'enclave Nitro. L'immagine digest è l'hash dell'immagine dell'enclave. SHA384 Puoi possibile utilizzare il digest dell'immagine e i valori PCR in [condizioni per le policy delle chiavi e le policy IAM](#). Per informazioni su PCRs, consulta [Dove reperire le misure di un'enclave nella Guida per l'utente di Nitro Enclaves](#).AWS

Questa sezione mostra un esempio di voce di CloudTrail registro per ciascuna delle richieste di enclave Nitro supportate a. AWS KMS

Decrypt (per un'enclave)

L'esempio seguente mostra una voce di AWS CloudTrail registro di un'operazione [Decrypt](#) per un'enclave Nitro. AWS

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2020-07-27T22:58:24Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

```

},
"responseElements": null,
"additionalEventData": {
  "recipient": {
    "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
    "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
    "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
    "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
    "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
    "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
    "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
  }
},
"requestID": "b4a65126-30d5-4b28-98b9-9153da559963",
"eventID": "e5a2f202-ba1a-467c-b4ba-f729d45ae521",
"readOnly": true,
"resources": [
  {
    "accountId": "111122223333",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  }
],
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}

```

GenerateDataKey (per un'enclave)

L'esempio seguente mostra una voce di AWS CloudTrail registro di un'[GenerateDataKey](#) operazione per un' AWS enclave Nitro.

```

{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },

```

```

    "eventTime": "2014-11-04T00:52:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "numberOfBytes": 32
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
            "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
            "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
            "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
            "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
            "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
            "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
            "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
        }
    },
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}

```

GenerateDataKeyPair (per un'enclave)

L'esempio seguente mostra una voce di AWS CloudTrail registro di un'[GenerateDataKeyPair](#) operazione per un' AWS enclave Nitro.

```

{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",

```

```

        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-27T18:57:57Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyPair",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyPairSpec": "RSA_3072",
        "encryptionContext": {
            "Project": "Alpha"
        }
    },
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
},
"responseElements": null,
"additionalEventData": {
    "recipient": {
        "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
        "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
        "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
        "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
        "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
        "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
        "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
    }
},
"requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
"eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
"readOnly": true,
"resources": [
    {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
],
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"

```

```
}
```

GenerateRandom (per un'enclave)

L'esempio seguente mostra una voce di AWS CloudTrail registro di un'[GenerateRandom](#) operazione per un'enclave AWS Nitro.

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:52:37Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateRandom",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData": {
    "recipient": {
      "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
      "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
      "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
      "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
      "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
      "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
      "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
    }
  },
  "requestID": "df1e3de6-63bc-11e4-bc2b-4198b6150d5c",
  "eventID": "239cb9f7-ae05-4c94-9221-6ea30eef0442",
  "readOnly": true,
  "resources": [],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

Richieste di monitoraggio per NitroTPM

Per l'attestazione NitroTPM, il CloudTrail registro include l'ID del modulo (`attestationDocumentModuleId`) e i registri di configurazione della piattaforma () contenuti nel documento di attestazione. PCRs

L'ID del modulo è l'ID dell' EC2 istanza con NitroTPM con un identificatore TPM. È possibile utilizzare i valori PCR in [condizioni per le politiche chiave](#) e le politiche IAM.

Questa sezione mostra un esempio di voce di CloudTrail registro per ciascuna delle richieste NitroTPM supportate a. AWS KMS

Decrittografia (per un NitroTPM)

L'esempio seguente mostra una voce di AWS CloudTrail registro di un'operazione [Decrypt](#) per un NitroTPM.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2020-07-27T22:58:24Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": null,
  "additionalEventData": {
    "recipient": {
      "attestationDocumentModuleId": "i-123456789abcde123-tpm000000000000000000",
      "attestationDocumentNitroTPMPCR4": "<AttestationDocument.PCR4>",

```



```

        "attestationDocumentNitroTPMPCR7": "<AttestationDocument.PCR7>",
        "attestationDocumentNitroTPMPCR8": "<AttestationDocument.PCR8>",
        "attestationDocumentNitroTPMPCR9": "<AttestationDocument.PCR9>",
        "attestationDocumentNitroTPMPCR16": "<AttestationDocument.PCR16>",
        "attestationDocumentNitroTPMPCR23": "<AttestationDocument.PCR23>"
    }
},
"requestID": "b4a65126-30d5-4b28-98b9-9153da559963",
"eventID": "e5a2f202-ba1a-467c-b4ba-f729d45ae521",
"readOnly": true,
"resources": [
    {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
],
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}

```

GenerateDataKey (per un NitroTPM)

L'esempio seguente mostra una voce di AWS CloudTrail registro di un'[GenerateDataKey](#) operazione per un NitroTPM.

```

{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",

```

```

    "requestParameters": {
      "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "numberOfBytes": 32
    },
    "responseElements": null,
    "additionalEventData": {
      "recipient": {
        "attestationDocumentModuleId": "i-123456789abcde123-tpm000000000000000000",
        "attestationDocumentNitroTPMPCR4": "<AttestationDocument.PCR4>",
        "attestationDocumentNitroTPMPCR7": "<AttestationDocument.PCR7>",
        "attestationDocumentNitroTPMPCR8": "<AttestationDocument.PCR8>",
        "attestationDocumentNitroTPMPCR9": "<AttestationDocument.PCR9>",
        "attestationDocumentNitroTPMPCR16": "<AttestationDocument.PCR16>",
        "attestationDocumentNitroTPMPCR23": "<AttestationDocument.PCR23>"
      }
    },
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [{
      "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }
}

```

GenerateDataKeyPair (per un NitroTPM)

L'esempio seguente mostra una voce di AWS CloudTrail registro di un'[GenerateDataKeyPair](#) operazione per un NitroTPM.

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },

```

```

    "eventTime": "2020-07-27T18:57:57Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyPair",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
      "keyPairSpec": "RSA_3072",
      "encryptionContext": {
        "Project": "Alpha"
      },
      "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
      "recipient": {
        "attestationDocumentModuleId": "i-123456789abcde123-tpm000000000000000000",
        "attestationDocumentNitroTPMPCR4": "<AttestationDocument.PCR4>",
        "attestationDocumentNitroTPMPCR7": "<AttestationDocument.PCR7>",
        "attestationDocumentNitroTPMPCR8": "<AttestationDocument.PCR8>",
        "attestationDocumentNitroTPMPCR9": "<AttestationDocument.PCR9>",
        "attestationDocumentNitroTPMPCR16": "<AttestationDocument.PCR16>",
        "attestationDocumentNitroTPMPCR23": "<AttestationDocument.PCR23>"
      }
    },
    "requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
    "readOnly": true,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }

```

GenerateRandom (per un NitroTPM)

L'esempio seguente mostra una voce di AWS CloudTrail registro di un'[GenerateRandom](#) operazione per un NitroTPM.

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:52:37Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateRandom",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": null,
  "additionalEventData": {
    "recipient": {
      "attestationDocumentModuleId": "i-123456789abcde123-tpm000000000000000000",
      "attestationDocumentNitroTPMPCR4": "<AttestationDocument.PCR4>",
      "attestationDocumentNitroTPMPCR7": "<AttestationDocument.PCR7>",
      "attestationDocumentNitroTPMPCR8": "<AttestationDocument.PCR8>",
      "attestationDocumentNitroTPMPCR9": "<AttestationDocument.PCR9>",
      "attestationDocumentNitroTPMPCR16": "<AttestationDocument.PCR16>",
      "attestationDocumentNitroTPMPCR23": "<AttestationDocument.PCR23>"
    }
  },
  "requestID": "df1e3de6-63bc-11e4-bc2b-4198b6150d5c",
  "eventID": "239cb9f7-ae05-4c94-9221-6ea30eef0442",
  "readOnly": true,
  "resources": [],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

Utilizzo della AWS KMS crittografia con AWS i servizi

Con AWS Key Management Service, puoi fornire chiavi di crittografia per proteggere i dati in altri AWS servizi. L'utilizzo della AWS KMS crittografia con AWS i servizi si riferisce al processo di integrazione AWS KMS con altri AWS servizi per crittografare e decrittografare i dati inattivi o in transito. Sviluppatori, amministratori di sistema e professionisti della sicurezza potrebbero essere interessati a questo argomento per proteggere i dati sensibili archiviati o trasmessi tramite AWS servizi, soddisfare i requisiti di conformità normativa o implementare le migliori pratiche di crittografia. I casi d'uso più comuni includono la crittografia di volumi Amazon EBS, bucket Amazon S3 e database Amazon RDS. [Le sezioni seguenti illustreranno i passaggi per configurare e gestire le chiavi di AWS KMS crittografia per AWS servizi specifici, garantendo la riservatezza e l'integrità dei dati in tutto l' AWS ambiente. Per l'elenco completo dei AWS servizi integrati con, consulta \[Service Integration. AWS KMSAWS\]\(#\)](#)

I seguenti argomenti illustrano in dettaglio l'utilizzo di determinati servizi AWS KMS, comprese le chiavi KMS che supportano, come gestiscono le chiavi di dati, le autorizzazioni richieste e come tenere traccia dell'utilizzo delle chiavi KMS nell'account da parte di ciascun servizio.

Important

[AWS i servizi integrati con AWS KMS](#) utilizzano solo chiavi KMS di crittografia simmetrica per crittografare i dati. Questi servizi non supportano la crittografia con chiavi KMS asimmetriche. Per informazioni su come determinare se una è simmetrica o asimmetrica, consulta [Identifica diversi tipi di chiave](#).

Argomenti

- [Come utilizza Amazon Elastic Block Store \(Amazon EBS\) AWS KMS](#)
- [Come utilizza Amazon EMR AWS KMS](#)
- [Come utilizza Amazon Redshift AWS KMS](#)

Come utilizza Amazon Elastic Block Store (Amazon EBS) AWS KMS

Questo argomento illustra in dettaglio come [Amazon Elastic Block Store \(Amazon EBS\) utilizza AWS KMS per crittografare volumi](#) e snapshot. Per istruzioni di base su come crittografare i volumi Amazon EBS, consulta [Crittografia Amazon EBS](#).

Argomenti

- [Crittografia Amazon EBS](#)
- [Utilizzo di chiavi KMS e chiavi dati](#)
- [Contesto di crittografia di Amazon EBS](#)
- [Rilevamento degli errori Amazon EBS](#)
- [Utilizzo AWS CloudFormation per creare volumi Amazon EBS crittografati](#)

Crittografia Amazon EBS

Quando colleghi un volume Amazon EBS crittografato a un [tipo di istanza Amazon Elastic Compute Cloud EC2 \(Amazon\) supportato](#), i dati archiviati inattivi sul volume, l'I/O del disco e le istantanee create dal volume sono tutti crittografati. La crittografia avviene sui server che ospitano EC2 le istanze Amazon.

Questa caratteristica è supportata su tutti i [tipi di volume Amazon EBS](#). Accedi ai volumi crittografati nello stesso modo in cui accedi ad altri volumi; la crittografia e la decrittografia vengono gestite in modo trasparente e non richiedono alcuna azione aggiuntiva da parte tua, della tua EC2 istanza o della tua applicazione. Gli snapshot di volumi crittografati vengono automaticamente crittografati e i volumi creati da snapshot crittografati vengono anch'essi automaticamente crittografati.

Lo stato di crittografia di un volume EBS viene determinato quando si crea il volume. Non è possibile modificare lo stato di crittografia di un volume esistente. Tuttavia, è possibile [eseguire la migrazione dei dati](#) tra i volumi crittografati e non crittografati e applicare un nuovo stato di crittografia durante la copia di uno snapshot.

Amazon EBS supporta la crittografia opzionale per impostazione predefinita. Puoi abilitare automaticamente la crittografia su tutti i nuovi volumi EBS e le copie istantanee nella tua regione. Account AWS Questa impostazione di configurazione non influisce sui volumi o sugli snapshot

esistenti. Per informazioni dettagliate, consulta la [crittografia di Amazon EBS](#) nella Amazon EBS User Guide.

Utilizzo di chiavi KMS e chiavi dati

Quando [crei un volume Amazon EBS crittografato](#), specifichi una AWS KMS key. Per impostazione predefinita, Amazon EBS utilizza la [Chiave gestita da AWS](#) per Amazon EBS nel tuo account (aws/ebs). Tuttavia puoi specificare una [chiave gestita dal cliente](#) creata e gestita da te.

Per utilizzare una chiave gestita dal cliente, è necessario concedere ad Amazon EBS l'autorizzazione a utilizzare una chiave KMS per conto dell'utente. Per ulteriori informazioni, consulta [Come funziona la crittografia di Amazon EBS nella Guida](#) per l'utente di Amazon EBS.

Important

Amazon EBS supporta solo [chiavi KMS simmetriche](#). Non è possibile utilizzare una [chiave KMS asimmetrica](#) per crittografare il volume Amazon EBS. Per informazioni su come determinare se una chiave KMS è simmetrica o asimmetrica, consulta [Identifica diversi tipi di chiave](#).

Per ogni volume, Amazon EBS chiede di AWS KMS generare una chiave dati univoca crittografata con la chiave KMS specificata. Amazon EBS archivia la chiave di dati crittografata con il volume. Quindi, quando colleghi il volume a un' EC2 istanza Amazon, Amazon EBS chiama AWS KMS per decrittografare la chiave dati. Amazon EBS utilizza la chiave dati in chiaro nella memoria dell'hypervisor per crittografare tutto il disco nel volume. I/O Per i dettagli, consulta [Come funziona la crittografia EBS nella Amazon EC2 User Guide](#) o [Amazon EC2 User Guide](#).

Contesto di crittografia di Amazon EBS

Nelle sue richieste [GenerateDataKeyWithoutPlaintext](#) e [Decrypt](#), AWS KMS Amazon EBS utilizza un contesto di crittografia con una coppia nome-valore che identifica il volume o lo snapshot nella richiesta. Il nome nel contesto di crittografia non varia.

Un [contesto di crittografia](#) è un set di coppie chiave-valore che contiene dati arbitrari non segreti. Quando includi un contesto di crittografia in una richiesta di crittografia dei dati, associa AWS KMS crittograficamente il contesto di crittografia ai dati crittografati. lo stesso contesto di crittografia sia necessario per decrittografare i dati.

Per tutti i volumi e per gli snapshot crittografati creati con l'[CreateSnapshot](#) operazione Amazon EBS, Amazon EBS utilizza l'ID del volume come valore del contesto di crittografia. Nel campo `requestParameters` di una voce di log di CloudTrail, il contesto di crittografia è simile a quanto segue:

```
"encryptionContext": {  
  "aws:ebs:id": "vol-0cfb133e847d28be9"  
}
```

Per le istantanee crittografate create con l' EC2 [CopySnapshot](#) operazione Amazon, Amazon EBS utilizza l'ID snapshot come valore del contesto di crittografia. Nel campo `requestParameters` di una voce di log di CloudTrail, il contesto di crittografia è simile a quanto segue:

```
"encryptionContext": {  
  "aws:ebs:id": "snap-069a655b568de654f"  
}
```

Rilevamento degli errori Amazon EBS

Per creare un volume EBS crittografato o collegare il volume a un' EC2 istanza, Amazon EBS e l' EC2 infrastruttura Amazon devono essere in grado di utilizzare la chiave KMS specificata per la crittografia dei volumi EBS. Quando la chiave KMS non è utilizzabile, ad esempio quando lo [stato della chiave](#) non è `Enabled`, la creazione del volume o l'allegato del volume ha esito negativo.

In questo caso, Amazon EBS invia un evento ad Amazon EventBridge (precedentemente CloudWatch Events) per informarti dell'errore. In EventBridge, puoi stabilire regole che attivano azioni automatiche in risposta a questi eventi. Per ulteriori informazioni, consulta [Amazon CloudWatch Events for Amazon EBS](#) nella Guida per l'utente di Amazon EBS, in particolare le seguenti sezioni:

- [Chiave di crittografia non valida per il collegamento il ricollegamento del volume](#)
- [Chiave di crittografia non valida su Crea volume](#)

Per risolvere questi problemi, assicurati che la chiave KMS specificata per la crittografia del volume EBS sia abilitata. A tale scopo, [visualizza innanzitutto la chiave KMS](#) per determinare lo stato corrente della chiave (la colonna Status in). Console di gestione AWS Quindi, consulta le informazioni contenute in uno dei seguenti collegamenti:

- Se lo stato di chiave KMS è disabilitato, [abilitalo](#).

- Se lo stato di chiave KMS è in importazione in sospeso, [importa il materiale chiave](#).
- Se lo stato di chiave KMS è in eliminazione in sospeso, [annulla l'eliminazione della chiave](#).

Utilizzo AWS CloudFormation per creare volumi Amazon EBS crittografati

È possibile utilizzare [AWS CloudFormation](#) per creare volumi di Amazon EBS crittografati. Per ulteriori informazioni, consulta [AWS::EC2::Volume](#) nella Guida per l'utente di AWS CloudFormation .

Come utilizza Amazon EMR AWS KMS

Quando si utilizza un cluster [Amazon EMR](#), è possibile configurare il cluster per crittografare i dati inattivi prima di salvarli in una posizione di storage persistente. È possibile crittografare i dati inattivi su EMR File System (EMRFS), sui volumi di storage dei nodi del cluster o su entrambi. Per crittografare i dati a riposo è possibile utilizzare una AWS KMS key. Gli argomenti seguenti descrivono come un cluster Amazon EMR utilizza una chiave KMS per crittografare i dati inattivi.

Important

Amazon EMR supporta solo [chiavi KMS simmetriche](#). Non è possibile utilizzare una [chiave KMS asimmetrica](#) per crittografare i dati inattivi in un cluster Amazon EMR. Per informazioni su come determinare se una è simmetrica o asimmetrica, consulta [Identifica diversi tipi di chiave](#).

I cluster Amazon EMR crittografano anche i dati in transito, il che significa che il cluster crittografa i dati prima dell'invio in rete. Non è possibile utilizzare una chiave KMS per crittografare i dati in transito. Per ulteriori informazioni, consulta [Crittografia dei dati in transito](#) nella Guida alla gestione di Amazon EMR.

Per ulteriori informazioni su tutte le opzioni di crittografia disponibili in Amazon EMR, consulta [Opzioni di crittografia](#) nella Guida alla gestione di Amazon EMR.

Argomenti

- [Crittografia dei dati su EMR File System \(EMRFS\)](#)
- [Crittografia dei dati su volumi di storage di nodi cluster](#)
- [Contesto di crittografia](#)

Crittografia dei dati su EMR File System (EMRFS)

I cluster Amazon EMR usano due file system distribuiti:

- Hadoop Distributed File System (HDFS). La crittografia HDFS non utilizza una chiave KMS in AWS KMS.
- File system EMR (EMRFS). EMRFS è un'implementazione di HDFS che consente ai cluster Amazon EMR di archiviare dati in Amazon Simple Storage Service (Amazon S3). EMRFS supporta quattro opzioni di crittografia, due delle quali utilizzano una chiave KMS in AWS KMS. Per ulteriori informazioni su tutte le opzioni di crittografia disponibili in Amazon EMRFS, consulta [Opzioni di crittografia](#) nella Guida alla gestione di Amazon EMR.

Le due opzioni di crittografia EMRFS che usano una chiave KMS utilizzano le seguenti caratteristiche di crittografia offerte da Amazon S3:

- [Protezione dei dati utilizzando la crittografia lato server con AWS Key Management Service \(SSE-KMS\)](#). Il cluster Amazon EMR invia i dati ad Amazon S3. Amazon S3 utilizza una chiave KMS per crittografare i dati prima di salvarli in un bucket S3. Per ulteriori informazioni su come effettuare tale operazione, consulta [Processo di crittografia dei dati su EMRFS con SSE-KMS](#).
- [Protezione dei dati con la crittografia lato client](#) (CSE-KMS). I dati in un Amazon EMR sono crittografati con una AWS KMS key prima di essere inviati ad Amazon S3 per l'archiviazione. Per ulteriori informazioni su come effettuare tale operazione, consulta [Processo di crittografia dei dati su EMRFS con CSE-KMS](#).

Quando si configura un cluster Amazon EMR per crittografare i dati su EMRFS con una chiave KMS, si sceglie la chiave KMS che si desidera sia utilizzata da Amazon S3 o dal cluster Amazon EMR. Con SSE-KMS, puoi scegliere la Chiave gestita da AWS per Amazon S3 con l'alias `aws/s3` o una chiave simmetrica gestita dal cliente che crei. Con la crittografia lato client, è necessario scegliere una chiave gestita dal cliente simmetrica creata dall'utente. Quando scegli una chiave gestita dal cliente, devi assicurarti che il cluster Amazon EMR abbia le autorizzazioni per utilizzare la chiave KMS. Per ulteriori informazioni, consulta [Using AWS KMS keys for encryption](#) nella Amazon EMR Management Guide.

Per la crittografia lato server e lato client, la chiave KMS scelta è la chiave root in un flusso di lavoro di [crittografia envelope](#). I dati vengono crittografati con una [chiave dati](#) univoca crittografata con la chiave KMS. AWS KMS I dati crittografati e una copia crittografata della chiave di dati vengono

archiviati insieme come un singolo oggetto crittografato in un bucket S3. Per ulteriori informazioni sul funzionamento, consulta gli argomenti indicati di seguito.

Argomenti

- [Processo di crittografia dei dati su EMRFS con SSE-KMS](#)
- [Processo di crittografia dei dati su EMRFS con CSE-KMS](#)

Processo di crittografia dei dati su EMRFS con SSE-KMS

Quando configuri un cluster Amazon EMR per utilizzare SSE-KMS, il processo di crittografia lavora come segue:

1. Il cluster invia i dati ad Amazon S3 per lo storage in un bucket S3.
2. Amazon S3 invia una [GenerateDataKey](#) richiesta a AWS KMS, specificando l'ID della chiave KMS che hai scelto quando hai configurato il cluster per l'utilizzo di SSE-KMS. La richiesta include il contesto di crittografia, per ulteriori informazioni consulta [Contesto di crittografia](#).
3. AWS KMS genera una chiave di crittografia dei dati univoca (chiave dati) e quindi invia due copie di questa chiave dati ad Amazon S3. Una copia non è crittografata (testo normale) e l'altra copia viene crittografata con la chiave KMS.
4. Amazon S3 utilizza la chiave di dati in testo in chiaro per crittografare i dati ricevuti nella fase 1, quindi rimuove il prima possibile la chiave di dati in testo normale dalla memoria dopo l'utilizzo.
5. Amazon S3 archivia i dati crittografati e una copia crittografata della chiave di dati insieme come un singolo oggetto crittografato in un bucket S3.

Il processo di decrittografia avviene in questo modo:

1. Il cluster richiede un oggetto dati crittografati da un bucket S3.
2. [Amazon S3 estrae la chiave dati crittografata dall'oggetto S3, quindi invia la chiave dati crittografata a AWS KMS con una richiesta Decrypt](#). La richiesta include un [contesto di crittografia](#).
3. AWS KMS decrittografa la chiave dati crittografata utilizzando la stessa chiave KMS utilizzata per crittografarla, quindi invia la chiave dati decrittografata (testo semplice) ad Amazon S3.
4. Amazon S3 utilizza la chiave di dati in testo normale per decrittografare i dati crittografati, quindi rimuove il prima possibile la chiave di dati in testo in chiaro dalla memoria dopo l'utilizzo.
5. Amazon S3 invia i dati decrittografati al cluster.

Processo di crittografia dei dati su EMRFS con CSE-KMS

Quando configuri un cluster Amazon EMR per utilizzare CSE-KMS, il processo di crittografia lavora come segue:

1. Quando è pronto per archiviare i dati in Amazon S3, il cluster invia una [GenerateDataKey](#) richiesta a AWS KMS, specificando l'ID chiave della chiave KMS che hai scelto quando hai configurato il cluster per l'utilizzo di CSE-KMS. La richiesta include il contesto di crittografia, per ulteriori informazioni consulta [Contesto di crittografia](#).
2. AWS KMS genera una chiave di crittografia dei dati unica (chiave dati) e quindi invia due copie di questa chiave di dati al cluster. Una copia non è crittografata (testo normale) e l'altra copia viene crittografata con la chiave KMS.
3. Il cluster utilizza la chiave di dati in testo normale per crittografare i dati, quindi rimuove il prima possibile la chiave di dati in testo normale dalla memoria dopo l'utilizzo.
4. Il cluster abbina i dati crittografati e una copia crittografata della chiave di dati insieme in un singolo oggetto crittografato.
5. Il cluster invia l'oggetto crittografato ad Amazon S3 per lo storage.

Il processo di decrittografia avviene in questo modo:

1. Il cluster richiede un oggetto dati crittografati a un bucket S3.
2. Amazon S3 invia l'oggetto crittografato al cluster.
3. Il cluster estrae la chiave dati crittografata dall'oggetto crittografato e quindi invia la chiave dati crittografata a AWS KMS con una richiesta [Decrypt](#). La richiesta include il [contesto di crittografia](#).
4. AWS KMS decrittografa la chiave dati crittografata utilizzando la stessa chiave KMS utilizzata per crittografarla, quindi invia la chiave di dati decrittografata (testo semplice) al cluster.
5. Il cluster utilizza la chiave di dati in testo normale per decrittografare i dati crittografati, quindi rimuove il prima possibile la chiave di dati in testo normale dalla memoria dopo l'utilizzo.

Crittografia dei dati su volumi di storage di nodi cluster

Un cluster Amazon EMR è una raccolta di istanze Amazon Elastic Compute Cloud (Amazon EC2). Ogni istanza nel cluster viene chiamata nodo cluster o nodo. Ogni nodo può avere due tipi di volumi di archiviazione: volumi di archiviazione di istanze e volumi Amazon Elastic Block Store (Amazon EBS). È possibile configurare il cluster per l'utilizzo di [Linux Unified Key Setup \(LUKS\)](#) per

crittografare entrambi i tipi di volumi di storage sui nodi (ma non il volume di avvio di ogni nodo). Questa si chiama la crittografia dei dati su disco locale.

Quando si abilita la crittografia dei dati su disco locale per un cluster, è possibile scegliere di crittografare la chiave LUKS con una chiave KMS in AWS KMS. Devi scegliere una [chiave gestita dal cliente](#) che hai creato; non è possibile utilizzare una [Chiave gestita da AWS](#). Se scegli una gestita dal cliente, devi assicurarti che il cluster Amazon EMR abbia le autorizzazioni per utilizzare la chiave KMS. Per ulteriori informazioni, consulta [Using AWS KMS keys for encryption](#) nella Amazon EMR Management Guide.

Quando abiliti la crittografia dei dati su disco locale utilizzando una chiave KMS, il processo di crittografia funziona in questo modo:

1. All'avvio di ogni nodo del cluster, invia una [GenerateDataKey](#) richiesta a AWS KMS, specificando l'ID della chiave KMS scelta quando hai abilitato la crittografia del disco locale per il cluster.
2. AWS KMS genera una chiave di crittografia dei dati univoca (chiave dati) e quindi invia due copie di questa chiave di dati al nodo. Una copia non è crittografata (testo normale) e l'altra copia viene crittografata con la chiave KMS.
3. Il nodo utilizza una versione con codifica base64 della chiave di dati di testo normale come password che protegge la chiave LUKS. Il nodo salva la copia crittografata della chiave di dati per il volume di avvio.
4. [Se il nodo si riavvia, il nodo riavviato invia la chiave dati crittografata a AWS KMS con una richiesta Decrypt.](#)
5. AWS KMS decrittografa la chiave dati crittografata utilizzando la stessa chiave KMS utilizzata per crittografarla, quindi invia la chiave di dati decrittografata (testo semplice) al nodo.
6. Il nodo utilizza una versione con codifica base64 della chiave di dati di testo normale come password per sbloccare la chiave LUKS.

Contesto di crittografia

Ogni AWS servizio integrato con AWS KMS può specificare un [contesto di crittografia](#) quando il servizio utilizza AWS KMS per generare chiavi di dati o crittografare o decrittografare i dati. Il contesto di crittografia è costituito da informazioni autenticate aggiuntive AWS KMS utilizzate per verificare l'integrità dei dati. Quando un servizio specifica un contesto di crittografia per un'operazione di crittografia, il servizio deve specificare lo stesso contesto di crittografia anche per l'operazione di decrittografia corrispondente o la decrittografia non riuscirà. Il contesto di crittografia viene inoltre

scritto nei file di AWS CloudTrail registro, il che può aiutarti a capire perché è stata utilizzata una chiave KMS specifica.

La sezione seguente spiega il contesto di crittografia utilizzato in ogni scenario di crittografia Amazon EMR che utilizza una chiave KMS.

Contesto di crittografia per la crittografia EMRFS con SSE-KMS

Con SSE-KMS, il cluster Amazon EMR invia i dati ad Amazon S3, quindi Amazon S3 utilizza una chiave KMS per crittografare i dati prima di salvarli in un bucket S3. In questo caso, Amazon S3 utilizza l'Amazon Resource Name (ARN) dell'oggetto S3 come contesto di crittografia per ogni richiesta [GenerateDataKey](#) and [Decrypt](#) a cui viene inviata. AWS KMS L'esempio seguente mostra una rappresentazione JSON del contesto di crittografia che Amazon S3 utilizza.

```
{ "aws:s3:arn" : "arn:aws:s3:::S3_bucket_name/S3_object_key" }
```

Contesto di crittografia per la crittografia EMRFS con CSE-KMS

Con CSE-KMS, il cluster Amazon EMR utilizza una chiave KMS per crittografare i dati prima di inviarli a Amazon S3 per l'archiviazione. In questo caso, il cluster utilizza l'Amazon Resource Name (ARN) della chiave KMS come contesto di crittografia per ogni richiesta [GenerateDataKey](#) and [Decrypt](#) a cui invia. AWS KMS L'esempio seguente mostra una rappresentazione JSON del contesto di crittografia che il cluster utilizza.

```
{ "kms_cmek_id" : "arn:aws:kms:us-east-2:111122223333:key/0987ab65-43cd-21ef-09ab-87654321cdef" }
```

Contesto di crittografia per la crittografia su disco locale con LUKS

Quando un cluster Amazon EMR utilizza la crittografia del disco locale con LUKS, i nodi del cluster non specificano il contesto di crittografia con le richieste [GenerateDataKey](#) [Decrypt](#) a cui inviano. AWS KMS

Come utilizza Amazon Redshift AWS KMS

Questo argomento illustra come Amazon Redshift AWS KMS utilizza per crittografare i dati.

Argomenti

- [Crittografia di Amazon Redshift](#)
- [Contesto di crittografia](#)

Crittografia di Amazon Redshift

Un data warehouse Amazon Redshift è costituito da un insieme di risorse di calcolo denominate nodi, strutturate in un gruppo denominato cluster. Ciascun cluster esegue un motore Amazon Redshift e contiene uno o più database.

Per la crittografia Amazon Redshift usa un'architettura a quattro livelli basata su chiavi. L'architettura consiste in chiavi di crittografia dei dati, una chiave di database, una chiave del cluster e una chiave root. Puoi usare un AWS KMS key come chiave principale.

Le chiavi di crittografia dei dati crittografano i blocchi di dati nel cluster. Ogni blocco di dati viene assegnato una chiave AES-256 generata in modo casuale. Queste chiavi sono crittografate utilizzando la chiave di database per il cluster.

La chiave di database crittografa le chiavi di crittografia dei dati nel cluster. La chiave del database è una chiave AES-256 generata in modo casuale. È archiviata su disco in una rete separata dal cluster Amazon Redshift e passata al cluster attraverso un canale sicuro.

La chiave del cluster crittografa la chiave di database per il cluster Amazon Redshift. È possibile utilizzare AWS KMS, AWS CloudHSM, o un modulo di sicurezza hardware esterno (HSM) per gestire la chiave del cluster. Consulta la documentazione di [Amazon Redshift Database Encryption](#) per ulteriori dettagli.

È possibile richiedere la crittografia selezionando la casella appropriata nella console Amazon Redshift. Puoi specificare una [chiave gestita dal cliente](#) da utilizzare scegliendone una dall'elenco che appare sotto la casella di crittografia. Se non specifichi una chiave gestita dal cliente, Amazon Redshift utilizza la [Chiave gestita da AWS](#) per Amazon Redshift sotto l'account.

Important

Amazon Redshift supporta solo chiavi KMS di crittografia simmetrica. Non è possibile utilizzare una chiave KMS asimmetrica come chiave master in un flusso di lavoro di crittografia Amazon Redshift. Per informazioni su come determinare se una chiave KMS è simmetrica o asimmetrica, consulta [Identifica diversi tipi di chiave](#).

Contesto di crittografia

Ogni servizio integrato con AWS KMS specifica un [contesto di crittografia](#) per la richiesta di chiavi di dati, la crittografia e la decrittografia. Il contesto di crittografia è costituito da dati autenticati aggiuntivi (AAD) utilizzati per verificare l'integrità dei dati. AWS KMS Questo significa che, quando viene specificato un contesto di crittografia per un'operazione di crittografia, il servizio specifica lo stesso contesto di crittografia anche per l'operazione di decrittografia o la decrittografia non riuscirà. Amazon Redshift utilizza l'ID cluster e il tempo di creazione per il contesto di crittografia. Nel `requestParameters` campo di un file di CloudTrail registro, il contesto di crittografia sarà simile a questo.

```
"encryptionContext": {  
  "aws:redshift:arn": "arn:aws:redshift:region:account_ID:cluster:cluster_name",  
  "aws:redshift:createtime": "20150206T1832Z"  
},
```

Puoi cercare il nome del cluster nei tuoi CloudTrail log per capire quali operazioni sono state eseguite utilizzando una AWS KMS key (chiave KMS). Le operazioni includono la crittografia e la decrittografia dei cluster e la generazione di chiavi di dati.

AWS KMS Riferimento

Il seguente materiale di riferimento fornisce informazioni utili sull'uso e la gestione delle chiavi KMS.

- [Documentazione di riferimento dei tipi di chiave](#). Elenca il tipo di chiave KMS che supporta ciascuna operazione API AWS KMS .

Per scoprire: è possibile abilitare e disabilitare una chiave KMS di firma RSA?

- [Tabella dello stato delle chiavi](#). Mostra come lo stato chiave di una chiave KMS influisce sul suo utilizzo nelle operazioni API AWS KMS .

Per scoprire: è possibile modificare l'alias di una chiave KMS in attesa di eliminazione?

- AWS KMS Riferimento alle [autorizzazioni API](#). Fornisce informazioni sulle autorizzazioni richieste per ogni operazione AWS KMS API.

Per trovare: Posso eseguire [GetKeyPolicy](#) l'esecuzione con una chiave in un altro AWS account? È possibile concedere l'autorizzazione `kms:Decrypt` in una policy IAM?

- [ViaService riferimento](#). Elenca i servizi AWS che supportano la chiave di condizione `kms:ViaService`.

Per trovare: posso utilizzare la chiave di `kms:ViaService` condizione per consentire un'autorizzazione solo quando proviene da Amazon ElastiCache? E per Amazon Neptune?

- [AWS KMS prezzi](#). Elenca e spiega il prezzo delle chiavi KMS.

Per scoprire: quanto costa usare le chiavi asimmetriche?

- [AWS KMS richiedere quote](#). Elenca le quote al secondo per le richieste AWS KMS API in ogni account e regione.

Per scoprire: quante richieste [Decrypt](#) possono essere eseguite al secondo? Quante richieste [Decrypt](#) richieste possono essere eseguite sulle chiavi KMS in un archivio chiavi personalizzate?

- [AWS KMS quote di risorse](#). Elenca le quote delle risorse AWS KMS .

Per scoprire: quante chiavi KMS è possibile avere in ogni Regione dell'account? Quanti alias è possibile avere su ogni chiave KMS?

- [AWS servizi integrati con AWS KMS](#). Elenca i AWS servizi che utilizzano le chiavi KMS per proteggere le risorse che creano, archiviano e gestiscono.

Per scoprire: Amazon Connect utilizza le chiavi KMS per proteggere le risorse Connect?

Stati chiave delle AWS KMS chiavi

Una chiave AWS KMS ha sempre uno stato chiave. Le operazioni sulla chiave KMS e sul relativo ambiente possono modificare lo stato della chiave. Lo stato della chiave può cambiare in modo transitorio o finché un'altra operazione non cambia lo stato della chiave. Queste operazioni vengono eseguite in modo asincrono o tramite una chiamata API.

La tabella in questa sezione mostra come gli stati chiave influiscono sulle chiamate alle AWS KMS operazioni API. Come risultato dello stato della chiave, si prevede che un'operazione su una chiave KMS abbia esito positivo (#), esito negativo (X), o esito positivo solo in determinate condizioni (?). Il risultato spesso differisce per le chiavi KMS con il materiale della chiave importato.

Questa tabella include solo le operazioni API che utilizzano una chiave KMS. Altre operazioni, come [CreateKey](#) e [ListKeys](#), vengono omesse.

Argomenti

- [Stato chiave e tipi di chiave KMS](#)
- [Tabella dello stato delle chiavi](#)

Stato chiave e tipi di chiave KMS

Il tipo di chiave KMS determina gli stati della chiave che può avere.

- Tutte le chiavi KMS possono essere incluse negli stati Enabled, Disabled e PendingDeletion.
- La maggior parte delle chiavi KMS viene creata nello stato Enabled. Le chiavi KMS con il materiale chiave importato vengono create nello stato PendingImport.
- Lo stato PendingImport si applica unicamente alle chiavi KMS con [materiale chiave importato](#). Quando un materiale chiave per una chiave importata viene eliminato o questa scade, lo stato cambia da aEnabled. PendingImport
- Lo stato Unavailable si applica unicamente a una chiave KMS in un [archivio delle chiavi personalizzate](#). Una chiave KMS in un [archivio AWS CloudHSM chiavi si verifica Unavailable quando l'archivio](#) chiavi personalizzato viene intenzionalmente disconnesso dal relativo cluster.

AWS CloudHSM Una chiave KMS in un [archivio delle chiavi esterne](#) è Unavailable quando l'archivio delle chiavi personalizzate viene disconnesso intenzionalmente dal relativo [proxy dell'archivio delle chiavi esterne](#). Puoi visualizzare e gestire chiavi KMS non disponibili, ma non puoi utilizzarle nelle operazioni di crittografia.

Lo stato di una chiave KMS in un archivio delle chiavi personalizzate non è influenzato dalle modifiche apportate al relativo materiale della chiave. Una chiave KMS in un archivio AWS CloudHSM chiavi non è influenzata dalle modifiche al [materiale chiave associato](#) nel cluster. AWS CloudHSM Una chiave KMS in un archivio delle chiavi esterne non è influenzata dalle modifiche apportate alla relativa [chiave esterna](#) in un gestore delle chiavi esterne. Se il materiale della chiave è disattivato o eliminato, lo stato della chiave KMS non cambia, ma le operazioni di crittografia che utilizzano la chiave KMS avranno esito negativo.

- Gli stati della chiave Creating, Updating e PendingReplicaDeletion si applicano solo alle [chiavi multiregione](#).
 - Una chiave di replica multiregione si trova nello stato della chiave Creating transitorio mentre è in fase di creazione. Questo processo potrebbe essere ancora in corso al termine dell'[ReplicateKey](#) operazione. Una volta completato il processo di replica, la chiave di replica si trova nello stato Enabled o PendingImport.
 - Le chiavi multi-regione si trovano nello stato della chiave Updating transitorio durante l'aggiornamento della Regione primaria. Questo processo potrebbe essere ancora in corso al termine dell'[UpdatePrimaryRegion](#) operazione. Al termine del processo di aggiornamento, le chiavi primarie e di replica riprendono lo stato della chiave Enabled.
 - Quando si pianificherà l'eliminazione di una chiave primaria multiregione che dispone di chiavi di replica, la chiave primaria si trova nello stato PendingReplicaDeletion finché non vengono eliminate tutte le chiavi di replica. Lo stato della chiave diventa PendingDeletion. Per informazioni dettagliate, consultare [Deleting multi-Region keys](#).

Tabella dello stato delle chiavi

Nella tabella seguente viene illustrato l'effetto dello stato chiave di una chiave KMS sulle operazioni AWS KMS .

Le descrizioni delle note a piè di pagina numerate ([n]) si trovano alla fine di questo argomento.

Note

Potrebbe essere necessario scorrere orizzontalmente o verticalmente per visualizzare tutti i dati di questa tabella.

API	Abilitato	Disabilitato	In attesa di eliminazione In attesa di eliminazione della replica	In attesa di importazione	Non disponibile	Creazione	Aggiornamento in corso
CancelKey Deletion	 [4]	 [4]		 [4]	 [4], [13]	 [4]	 [4]
CreateAlias			 [3]				
CreateGrant		 [1]	 [2] o [3]	 [5]		 [14]	
Decrypt		 [1]	 [2] o [3]	 [5]	 [11]	 [14]	
DeleteAlias							

API	Abilitato	Disabilitato	In attesa di eliminazione In attesa di eliminazione della replica	In attesa di importazione	Non disponibile	Creazione	Aggiornamento in corso
DeleteImportedKeyMaterial	✓ [9]	✓ [9]	✓ [9]	✓	N/D	✗ [14]	✗ [15]
DeriveSharedSecret	✓	✗ [1]	✗ [2] o [3]	✗ [5]	N/D	✗ [14]	✓
DescribeKey	✓	✓	✓	✓	✓	✓	✓
DisableKey	✓	✓	✗ [3]	✗ [5]	✓ [12]	✗ [14]	✗ [15]
DisableKeyRotation	❓ [7]	✗ [1] o [7]	✗ [3] o [7]	✗ [6]	✗ [7]	✗ [14]	❓ [7]
EnableKey	✓	✓	✗ [3]	✗ [5]	✓ [12]	✗ [14]	✗ [15]

API	Abilitato	Disabilitato	In attesa di eliminazione In attesa di eliminazione della replica	In attesa di importazione	Non disponibile	Creazione	Aggiornamento in corso
EnableKeyRotation	 [7]	 [1] o [7]	 [3] o [7]	 [6]	 [7]	 [14]	 [7]
Crittografia		 [1]	 [2] o [3]	 [5]	 [11]	 [14]	
GenerateDataKey		 [1]	 [2] o [3]	 [5]	 [11]	 [14]	
GenerateDataKeyPair		 [1]	 [2] o [3]	 [5]	 [7]	 [14]	
GenerateDataKeyPairWithoutPlaintext		 [1]	 [2] o [3]	 [5]	 [7]	 [14]	
GenerateDataKeyWithoutPlaintext		 [1]	 [2] o [3]	 [5]	 [11]	 [14]	

API	Abilitato	Disabilitato	In attesa di eliminazione In attesa di eliminazione della replica	In attesa di importazione	Non disponibile	Creazione	Aggiornamento in corso
GenerateMac	✓	✗ [1]	✗ [2] o [3]	✗ [5]	N/D	✗ [14]	✓
GetKeyPolicy	✓	✓	✓	✓	✓	✓	✓
GetKeyRotationStatus	?	?	?	✗ [6]	✗ [7]	?	?
GetParametersForImport	?	?	✗ [8] o [9]	✓	✗ [9]	✗ [14]	✗ [15]
GetPublicKey	✓	✓	✗ [2] o [3]	✓	N/D	✗ [14]	✓
ImportKeyMaterial	?	?	✗ [9]	✓	✗ [9]	✗ [14]	✓
ListAliases	✓	✓	✓	✓	✓	✓	✓

API	Abilitato	Disabilitato	In attesa di eliminazione In attesa di eliminazione della replica	In attesa di importazione	Non disponibile	Creazione	Aggiornamento in corso
ListGrants	✓	✓	✓	✓	✓	✓	✓
ListKeyPolicies	✓	✓	✓	✓	✓	✓	✓
ListKeyRotations	⓪ [7]	⓪ [7]	⓪ [7]	ⓧ [6]	ⓧ [7]	⓪ [7]	⓪ [7]
ListResourceTags	✓	✓	✓	✓	✓	✓	✓
PutKeyPolicy	✓	✓	✓	✓	✓	✓	✓
ReEncrypt	✓	ⓧ [1]	ⓧ [2] o [3]	ⓧ [5]	ⓧ [11]	ⓧ [14]	✓
Replicate Key	✓	ⓧ [1]	ⓧ [2] o [3]	ⓧ [5]	N/D	ⓧ [14]	ⓧ [15]
RetireGrant	✓	✓	✓	✓	✓	✓	✓

API	Abilitato	Disabilitato	In attesa di eliminazione In attesa di eliminazione della replica	In attesa di importazione	Non disponibile	Creazione	Aggiornamento in corso
RevokeGrant	✓	✓	✓	✓	✓	✓	✓
RotateKeyOnDemand	 [7]	 [1] o [7]	 [3] o [7]	 [5]	 [7]	 [14]	 [7]
ScheduleKeyDeletion	✓	✓	 [3]	✓	✓	✓	 [15]
Sign	✓	 [1]	 [2] o [3]	 [5]	N/D	 [14]	✓
TagResource	✓	✓	 [3]	✓	✓	✓	✓
UntagResource	✓	✓	 [3]	✓	✓	✓	✓

API	Abilitato	Disabilitato	In attesa di eliminazione In attesa di eliminazione della replica	In attesa di importazione	Non disponibile	Creazione	Aggiornamento in corso
UpdateAliases	✓	✓	❓ [10]	✓	✓	✓	✓
UpdateKeyDescription	✓	✓	🚫 [3]	✓	✓	✓	✓
UpdatePrimaryRegion	✓	🚫 [1]	🚫 [2] o [3]	🚫 [5]	N/D	🚫 [14]	✓
Verifica	✓	🚫 [1]	🚫 [2] o [3]	🚫 [5]	N/D	🚫 [14]	✓
VerifyMac	✓	🚫 [1]	🚫 [2] o [3]	🚫 [5]	N/D	🚫 [14]	✓

Dettagli tabella

- [1] DisabledException: *<key ARN>* is disabled.

- [2] DisabledException: `<key ARN>` is pending deletion (or pending replica deletion).
- [3] KMSInvalidStateException: `<key ARN>` is pending deletion (or pending replica deletion).
- [4] KMSInvalidStateException: `<key ARN>` is not pending deletion (or pending replica deletion).
- [5] KMSInvalidStateException: `<key ARN>` is pending import because no key material has ever been imported or one of the imported key materials is deleted or expired.
- [6] UnsupportedOperationException: `<key ARN>` origin is EXTERNAL which is not valid for this operation.
- [7] Se la chiave KMS si trova in un archivio di chiavi personalizzato:
UnsupportedOperationException
- [8] Se la chiave KMS ha importato il materiale della chiave: KMSInvalidStateException
- [9] Se la chiave KMS non può avere importato materiale chiave:
UnsupportedOperationException
- [10] Se la chiave KMS di origine è in attesa di eliminazione, il comando viene completato. Se la chiave KMS di destinazione è in attesa di eliminazione, il comando ha esito negativo con l'errore: KMSInvalidStateException : `<key ARN>` is pending deletion.
- [11] KMSInvalidStateException: `<key ARN>` is unavailable. Non puoi eseguire questa operazione su una chiave KMS non disponibile.
- [12] L'operazione riesce, ma lo stato di chiave della chiave KMS non cambia finché questa non diventa disponibile.
- [13] Quando una chiave KMS in un archivio delle chiavi personalizzate è in attesa di eliminazione, il relativo stato rimane PendingDeletion anche se la chiave KMS diventa non disponibile. Ciò ti consente di annullare l'eliminazione della chiave KMS in qualsiasi momento durante il periodo di attesa.
- [14] KMSInvalidStateException: `<key ARN>` is creating. AWS KMS genera questa eccezione mentre replica una chiave multiregionale (). ReplicateKey
- [15] KMSInvalidStateException: `<key ARN>` is updating. AWS KMS genera questa eccezione mentre aggiorna la regione principale di una chiave multiregionale (). UpdatePrimaryRegion

Documentazione di riferimento dei tipi di chiave

AWS KMS supporta diverse funzionalità per diversi tipi di chiavi KMS. Ad esempio, puoi utilizzare solo [chiavi KMS di crittografia simmetrica](#) per [generare chiavi di dati simmetriche](#) e [coppie di chiavi di dati asimmetriche](#). Inoltre, [l'importazione del materiale della chiave](#) e la [rotazione automatica delle chiavi](#) sono supportate solo per le chiavi KMS di crittografia simmetrica e in un [archivio delle chiavi personalizzate](#) è possibile creare solo chiavi KMS di crittografia simmetrica.

Questo riferimento include due tabelle.

- La [tabella dei tipi di chiave](#) elenca le AWS KMS operazioni valide per le chiavi KMS con crittografia simmetrica, le chiavi KMS asimmetriche e le chiavi KMS HMAC.
- La [tabella delle caratteristiche speciali](#) elenca le operazioni AWS KMS valide per le chiavi KMS multi-regione, le chiavi KMS con materiale della chiave importato e le chiavi KMS negli archivi delle chiavi personalizzate.

Tabella dei tipi di chiave

Potrebbe essere necessario scorrere orizzontalmente o verticalmente per visualizzare tutti i dati di questa tabella.

AWS KMS Funzionamento dell'API	Chiavi KMS di crittografia simmetrica	Chiavi KMS HMAC	Chiavi KMS asimmetriche (ENCRYPT DECRYPT)	Chiavi KMS asimmetriche (SIGN_VERIFY)	Chiavi KMS asimmetriche (KEY_AGREEMENT)
CancelKeyDeletion	Sì	Sì	Sì	Sì	Sì
CreateAlias	Sì	Sì	Sì	Sì	Sì
CreateGrant	Sì	Sì	Sì	Sì	Sì
CreateKey	Sì	Sì	Sì	Sì	Sì
Decrypt	Sì	No	Sì	No	No

AWS KMS Funzionamento dell'API	Chiavi KMS di crittografia simmetrica	Chiavi KMS HMAC	Chiavi KMS asimmetriche (ENCRYPT DECRYPT)	Chiavi KMS asimmetriche (SIGN_VERIFY)	Chiavi KMS asimmetriche (KEY_AGREEMENT)
DeleteAlias	Sì	Sì	Sì	Sì	Sì
DeleteImportedKeyMaterial Valido solo nelle chiavi KMS con materiale della chiave importato (Origin è EXTERNAL).	Sì	Sì	Sì	Sì	Sì
DeriveSharedSecret	No	No	No	No	Sì
DescribeKey	Sì	Sì	Sì	Sì	Sì
DisableKey	Sì	Sì	Sì	Sì	Sì

AWS KMS Funzionamento dell'API	Chiavi KMS di crittografia simmetrica	Chiavi KMS HMAC	Chiavi KMS asimmetriche (ENCRYPT DECRYPT)	Chiavi KMS asimmetriche (SIGN_VERIFY)	Chiavi KMS asimmetriche (KEY_AGREEMENT)
DisableKeyRotation	Sì Valido solo sulle chiavi KMS con (i) materiale chiave. AWS KMS Origin AWS_KMS	No	No	No	No
EnableKey	Sì	Sì	Sì	Sì	Sì

AWS KMS Funzionamento dell'API	Chiavi KMS di crittografia simmetrica	Chiavi KMS HMAC	Chiavi KMS asimmetriche (ENCRYPT DECRYPT)	Chiavi KMS asimmetriche (SIGN_VERIFY)	Chiavi KMS asimmetriche (KEY_AGREEMENT)
EnableKeyRotation	Sì	No	No	No	No
	Valido solo sulle chiavi KMS con (i) Origin materiale AWS KMS chiave. AWS_KMS				
Encrypt	Sì	No	Sì	No	No
GenerateDataKey	Sì	No	No	No	No

AWS KMS Funzionamento dell'API	Chiavi KMS di crittografia simmetrica	Chiavi KMS HMAC	Chiavi KMS asimmetriche (ENCRYPT DECRYPT)	Chiavi KMS asimmetriche (SIGN_VERIFY)	Chiavi KMS asimmetriche (KEY_AGREEMENT)
GenerateDataKeyPair Genera coppie di chiavi di dati asimmetriche protette da una chiave KMS di crittografia simmetrica.	Sì Non è valido per le chiavi KMS negli archivi delle chiavi personalizzate.	No	No	No	No
GenerateDataKeyPairWithoutPlaintext Genera coppie di chiavi di dati asimmetriche protette da una chiave KMS di crittografia simmetrica.	Sì Non è valido per le chiavi KMS negli archivi delle chiavi personalizzate.	No	No	No	No
GenerateDataKeyWithoutPlaintext	Sì	No	No	No	No

AWS KMS Funzionamento dell'API	Chiavi KMS di crittografia simmetrica	Chiavi KMS HMAC	Chiavi KMS asimmetriche (ENCRYPT DECRYPT)	Chiavi KMS asimmetriche (SIGN_VERIFY)	Chiavi KMS asimmetriche (KEY_AGREEMENT)
GenerateMac	No	Sì	No	No	No
GetKeyPolicy	Sì	Sì	Sì	Sì	Sì
GetKeyRotationStatus	Sì	(KeyRotationEnabled sarà sempre false).			(KeyRotationEnabled sarà sempre false).
GetParametersForImport Valido solo nelle chiavi KMS con materiale della chiave importato (Origin è EXTERNAL).	Sì	Sì	Sì	Sì	Sì
GetPublicKey	No	No	Sì	Sì	Sì
ImportKeyMaterial Valido solo nelle chiavi KMS con materiale della chiave importato (Origin è EXTERNAL).	Sì	Sì	Sì	Sì	Sì

AWS KMS Funzionamento dell'API	Chiavi KMS di crittografia simmetrica	Chiavi KMS HMAC	Chiavi KMS asimmetriche (ENCRYPT DECRYPT)	Chiavi KMS asimmetriche (SIGN_VERIFY)	Chiavi KMS asimmetriche (KEY_AGREEMENT)
ListAliases	Sì	Sì	Sì	Sì	Sì
ListGrants	Sì	Sì	Sì	Sì	Sì
ListKeyPolicies	Sì	Sì	Sì	Sì	Sì
ListKeyRotations	Sì	Sì (Il Rotation campo sarà sempre nullo o vuoto.)	Sì (Il Rotation campo sarà sempre nullo o vuoto.)	Sì (Il Rotations campo sarà sempre nullo o vuoto.)	Sì (Il Rotations campo sarà sempre nullo o vuoto.)
ListResourceTags	Sì	Sì	Sì	Sì	Sì
ListRetirableGrants	Sì	Sì	Sì	Sì	Sì
PutKeyPolicy	Sì	Sì	Sì	Sì	Sì
ReEncrypt	Sì	No	Sì	No	No
ReplicateKey	Sì	Sì	Sì	Sì	Sì
- Valido solo su chiavi multi-Regione					
RetireGrant	Sì	Sì	Sì	Sì	Sì

AWS KMS Funzionamento dell'API	Chiavi KMS di crittografia simmetrica	Chiavi KMS HMAC	Chiavi KMS asimmetriche (ENCRYPT DECRYPT)	Chiavi KMS asimmetriche (SIGN_VERIFY)	Chiavi KMS asimmetriche (KEY_AGREEMENT)
RevokeGrant	Sì	Sì	Sì	Sì	Sì
RotateKeyOnDemand	Sì Valido solo su chiavi KMS di crittografia simmetrica gestite dal cliente con origine. AWS_KMS_EXTERNAL	No	No	No	No
ScheduleKeyDeletion	Sì	Sì	Sì	Sì	Sì
Sign	No	No	No	Sì	No
TagResource	Sì	Sì	Sì	Sì	Sì
UntagResource	Sì	Sì	Sì	Sì	Sì

AWS KMS Funzionamento dell'API	Chiavi KMS di crittografia simmetrica	Chiavi KMS HMAC	Chiavi KMS asimmetriche (ENCRYPT DECRYPT)	Chiavi KMS asimmetriche (SIGN_VERIFY)	Chiavi KMS asimmetriche (KEY_AGREEMENT)
UpdateAlias	Sì	Sì	Sì	Sì	Sì
La chiave KMS corrente e la nuova chiave KMS devono essere dello stesso tipo (entrambe simmetriche, asimmetriche o HMAC) e devono avere lo stesso utilizzo della chiave .					
UpdateKey Description	Sì	Sì	Sì	Sì	Sì
UpdateReplicaRegion	Sì	Sì	Sì	Sì	Sì
- Valido solo su chiavi multi-Regione					
Verify	No	No	No	Sì	No
VerifyMac	No	Sì	No	No	No

Tabella delle caratteristiche speciali

Questa tabella mostra le operazioni AWS KMS API supportate su ogni tipo di chiave speciale.

Durante la lettura di questa tabella, tieni a mente le interazioni seguenti:

- [Chiavi multi-regione](#):
 - Le chiavi KMS multi-regione possono essere costituite da chiavi KMS di crittografia simmetrica, chiavi KMS asimmetriche, chiavi KMS HMAC o chiavi KMS con materiale della chiave importato.
 - Non è possibile creare chiavi multi-regione in un archivio delle chiavi personalizzate.
- [Materiale della chiave importato](#)
 - Puoi importare il materiale della chiave per chiavi KMS di crittografia simmetrica, KMS asimmetriche e chiavi KMS HMAC.
 - Puoi creare [chiavi multi-regione con materiale della chiave importato](#).
 - Non puoi creare chiavi con materiale della chiave importato in un archivio delle chiavi personalizzate.
 - La rotazione automatica delle chiavi (`EnableKeyRotation`, `DisableKeyRotation`) non è supportata per le chiavi KMS con materiale della chiave importato.
 - La rotazione delle chiavi su richiesta (`RotateKeyOnDemand`) è supportata per le chiavi KMS di crittografia simmetrica con materiale chiave importato.
- [store delle chiavi personalizzate](#)
 - Gli archivi delle chiavi personalizzate supportano solo chiavi KMS di crittografia simmetrica.
 - Le operazioni di simmetria su coppie di chiavi asimmetriche (`GenerateDataKeyPair`, `GenerateDataKeyPairWithoutPlaintext`) non sono supportate sulle chiavi KMS negli archivi delle chiavi personalizzate.
 - La rotazione automatica delle chiavi (`EnableKeyRotation`, `DisableKeyRotation`) non è supportata sulle chiavi KMS negli archivi di chiavi personalizzate.
 - Non puoi creare chiavi multi-regione negli archivi delle chiavi personalizzate.

Potrebbe essere necessario scorrere orizzontalmente o verticalmente per visualizzare tutti i dati di questa tabella.

AWS KMS Funzionamento tramite API	Chiavi multi-regione	Materiale della chiave importato	Chiavi KMS in un archivio delle chiavi personalizzate
CancelKeyDeletion	✓	✓	✓
CreateAlias	✓	✓	✓
CreateGrant	✓	✓	✓
CreateKey Puoi utilizzare <code>CreateKey</code> per creare una chiave primaria multi-regione, una chiave KMS con materiale della chiave importato o una chiave KMS in un archivio delle chiavi personalizzate. Per creare una chiave di replica multi-regione, utilizza <code>ReplicateKey</code> .	✓	✓	✓
Decrypt	✓ Valido solo quando <code>KeyUsage</code> è <code>ENCRYPT_D</code> <code>ECRYPT</code>	✓	✓
DeleteAlias	✓	✓	✓
DeleteImportedKeyMaterial	✓	✓	✗

AWS KMS Funzionamento tramite API	Chiavi multi-regione	Materiale della chiave importato	Chiavi KMS in un archivio delle chiavi personalizzate
	Valido solo per le chiavi con materiale della chiave importato (Origin è EXTERNAL)		
DeriveSharedSecret	 Valido solo quando KeyUsage è KEY_AGREEMENT)	 Valido solo quando KeyUsage è KEY_AGREEMENT)	
DescribeKey			
DisableKey			
DisableKeyRotation	 Valido solo su chiavi di crittografia simmetriche con (i) Origin materiale AWS KMS chiave. AWS_KMS		

AWS KMS Funzionamento tramite API	Chiavi multi-regione	Materiale della chiave importato	Chiavi KMS in un archivio delle chiavi personalizzate
EnableKey	 Valido solo su chiavi KMS di crittografia simmetrica		
EnableKeyRotation	 Valido solo su chiavi di crittografia simmetriche con materiale AWS KMS chiave (i). Origin AWS_KMS		
Encrypt	 Valido solo quando KeyUsage è ENCRYPT_DECRYPT		

AWS KMS Funzionamento tramite API	Chiavi multi-regione	Materiale della chiave importato	Chiavi KMS in un archivio delle chiavi personalizzate
GenerateDataKey	✓ Valido solo su chiavi KMS di crittografia simmetrica	✓	✓
GenerateDataKeyPair	✓ Valido solo su chiavi KMS di crittografia simmetrica	✓	✗
GenerateDataKeyPairWithoutPlaintext	✓ Valido solo su chiavi KMS di crittografia simmetrica	✓	✗
GenerateDataKeyWithoutPlaintext	✓ Valido solo su chiavi KMS di crittografia simmetrica	✓	✓
GenerateMac Valido solo per chiavi KMS HMAC	✓	✓	✗

AWS KMS Funzionamento tramite API	Chiavi multi-regione	Materiale della chiave importato	Chiavi KMS in un archivio delle chiavi personalizzate
GetKeyPolicy	✓	✓	✓
GetKeyRotationStatus	✓	✓ (KeyRotationEnabled sarà sempre false).	✗
GetParametersForImport	✓ Valido solo per le chiavi con materiale della chiave importato (Origin è EXTERNAL).	✓	✗
GetPublicKey Valido solo per chiavi KMS asimmetriche .	✓	✓	✗
ImportKeyMaterial	✓ Valido solo per le chiavi con materiale della chiave importato (Origin è EXTERNAL).	✓	✗

AWS KMS Funzionamento tramite API	Chiavi multi-regione	Materiale della chiave importato	Chiavi KMS in un archivio delle chiavi personalizzate
ListAliases	✓	✓	✓
ListGrants	✓	✓	✓
ListKeyPolicies	✓	✓	✓
ListKeyRotations	✓ Valido solo su chiavi di crittografia simmetriche con o origine. AWS_KMS EXTERNAL	✓ Valido solo su chiavi di crittografia simmetriche.	✓
ListResourceTags	✓	✓	✓
ListRetirableGrants	✓	✓	✓
PutKeyPolicy	✓	✓	✓
ReEncrypt	✓ Valido solo quando KeyUsage è ENCRYPT_DECRYPT	✓	✓

AWS KMS Funzionamento tramite API	Chiavi multi-regione	Materiale della chiave importato	Chiavi KMS in un archivio delle chiavi personalizzate
ReplicateKey	✓ Valido solo su chiavi primarie multi-regione.	✓ Valido solo su chiavi primarie multi-regione.	✗
RetireGrant	✓	✓	✓
RevokeGrant	✓	✓	✓
RotateKeyOnDemand	✓ Valido solo su chiavi di crittografia simmetriche con o origine. AWS_KMS EXTERNAL	✓ Valido solo su chiavi di crittografia simmetriche.	✗
ScheduleKeyDeletion	✓	✓	✓
Sign Valido solo quando KeyUsage è SIGN_VERIFY .	✓	✓	✗
TagResource	✓	✓	✓
UntagResource	✓	✓	✓

AWS KMS Funzionamento tramite API	Chiavi multi-regione	Materiale della chiave importato	Chiavi KMS in un archivio delle chiavi personalizzate
UpdateAlias - La chiave KMS corrente e la nuova chiave KMS devono essere dello stesso tipo (entrambe simmetriche, asimmetriche o HMAC) e devono avere lo stesso utilizzo della chiave .	✓	✓	✓
UpdateKeyDescription	✓	✓	✓
UpdateReplicaRegion	✓	✓	✗
		Valido solo su chiavi multi-regione.	
Verify Valido solo quando KeyUsage è SIGN_VERIFY .	✓	✓	✗
VerifyMac Valido solo per chiavi KMS HMAC	✓	✓	✗

Riferimento alle specifiche chiave

Quando crei una chiave KMS asimmetrica o una chiave KMS HMAC, selezioni la relativa [specifica della chiave](#). La specifica della chiave, che è una proprietà di every AWS KMS key, rappresenta la configurazione crittografica della chiave KMS. La specifica della chiave viene scelta in fase di creazione della chiave KMS e non può essere modificata in seguito. Se hai selezionato la specifica della chiave errata, [elimina la chiave KMS](#) e creane una nuova.

Note

La specifica della chiave per una chiave del servizio di gestione delle chiavi era nota come "specifica chiave master cliente". Il `CustomerMasterKeySpec` parametro dell'[CreateKey](#) operazione è obsoleto. Utilizza invece il parametro `KeySpec`. La risposta delle [DescribeKey](#) operazioni `CreateKey` and include un `CustomerMasterKeySpec` membro `KeySpec` and con lo stesso valore.

Le specifiche della chiave determinano se la chiave KMS è simmetrica o asimmetrica, il tipo di materiale chiave nella chiave KMS e gli algoritmi di crittografia, gli algoritmi di firma o gli algoritmi MAC (Message Authentication Code) supportati dalla chiave KMS. AWS KMS La specifica della chiave scelta è in genere determinata dal caso d'uso e dai requisiti normativi. Tuttavia, le operazioni crittografiche sulle chiavi KMS con specifiche diverse della chiave ECC hanno prezzi differenti e sono soggette a varie quote. Per i dettagli sui prezzi, vedere [Prezzi di AWS Key Management Service](#). Per informazioni sulle quote di richieste, consulta [Quote di richieste](#).

[Per limitare le specifiche chiave che i principali possono utilizzare durante la creazione di chiavi KMS, usa la chiave kms: condition. KeySpec](#) Puoi anche usare la chiave `kms:KeySpec` condition per consentire ai principali di richiamare AWS KMS le operazioni solo sulle chiavi KMS con una specifica chiave particolare. Ad esempio, puoi negare l'autorizzazione per pianificare l'eliminazione di tutte le chiavi KMS con una specifica delle chiavi `RSA_4096`.

AWS KMS supporta le seguenti specifiche chiave per le chiavi KMS:

[Specifica della chiave crittografica simmetrica](#) (impostazione predefinita)

- `SYMMETRIC_DEFAULT`

[Specifiche della chiave RSA](#) (crittografia e decrittografia o firma e verifica)

- `RSA_2048`
- `RSA_3072`
- `RSA_4096`

[Specifiche della chiave basata su curva ellittica](#)

- [Coppie di chiavi asimmetriche a curva ellittica standard NIST \(firma e verifica, o derivazione di segreti condivisi\)](#)
 - `ECC_NIST_P256` (secp256r1)
 - `ECC_NIST_P384` (secp384r1)

- ECC_NIST_P521 (secp521r1)
- EDWARDS25519 ECC_NIST_ (ed25519) - solo firma e verifica
 - Nota: per le chiavi EDWARDS25519 KMS ECC_NIST_, è necessario l'algoritmo di firma _SHA_512, mentre _PH_SHA_512 richiede. ED25519 [MessageType:RAW](#) [ED25519MessageType:DIGEST](#) Questi tipi di messaggi non possono essere usati in modo intercambiabile.
- Altre coppie di chiavi asimmetriche basate su curva ellittica (firma e verifica)
 - ECC_SECG_P256K1 ([secp256k1](#)), comunemente usate per la criptovaluta.

[SM2 specifica chiave](#) (crittografia e decrittografia -oppure- firma e verifica -o- derivazione di segreti condivisi)

- SM2 (Solo regioni della Cina)

[Specifiche della chiave HMAC](#)

- HMAC_224
- HMAC_256
- HMAC_384
- HMAC_512

[Specifiche principali di ML-DSA](#)

- ML_DSA_44
- ML_DSA_65
- ML_DSA_87

Specifica della chiave SYMMETRIC_DEFAULT

La specifica della chiave di default, SYMMETRIC_DEFAULT, è la specifica della chiave per le chiavi KMS di crittografia simmetrica. Quando si seleziona il tipo di chiave simmetrica e l'utilizzo della chiave di crittografia e decrittografia nella console, viene selezionata la specifica della chiave. AWS KMS SYMMETRIC_DEFAULT Nell'[CreateKey](#) operazione, se non si specifica un valore, viene selezionato SYMMETRIC_DEFAULT. KeySpec Se non hai motivo di utilizzare una specifica della chiave diversa, SYMMETRIC_DEFAULT è una scelta valida.

SYMMETRIC_DEFAULT rappresenta AES-256-GCM, un algoritmo simmetrico basato su [Advanced Encryption Standard \(AES\) in Galois Counter Mode \(GCM\) con chiavi a 256 bit, uno standard di settore per la crittografia](#) sicura. Il testo cifrato generato da questo algoritmo supporta ulteriori dati

autenticati (AAD), ad esempio un [contesto di crittografia](#), e GCM fornisce un controllo di integrità aggiuntivo sul testo cifrato.

AES-256-GCM garantisce la protezione attuale e futura dei dati crittografati. I crittografi considerano questo algoritmo resistente alla quantistica. Attacchi teorici futuri di calcolo quantistico su larga scala su testi cifrati creati con chiavi AES-GCM a 256 bit [riducono l'effettiva sicurezza della chiave a 128 bit](#). Tuttavia, questo livello di sicurezza è sufficiente a rendere impossibili gli attacchi di forza bruta su testi cifrati. AWS KMS

L'unica eccezione nelle regioni della Cina, dove SYMMETRIC_DEFAULT rappresenta una chiave simmetrica a 128 bit che utilizza la crittografia. SM4 Puoi creare una SM4 chiave a 128 bit solo all'interno delle regioni della Cina. Non puoi creare una chiave KMS AES-GCM a 256 bit in Cina.

Puoi utilizzare una chiave KMS con crittografia simmetrica per crittografare, AWS KMS decrittografare e ricrittografare i dati e proteggere le chiavi e le coppie di chiavi di dati generate. AWS i servizi integrati con AWS KMS utilizzano chiavi KMS di crittografia simmetrica per crittografare i dati inattivi. Puoi [importare il materiale della chiave di tua proprietà](#) in una chiave KMS di crittografia simmetrica e creare chiavi KMS di crittografia simmetrica negli [archivi delle chiavi personalizzate](#). Per una tabella di confronto delle azioni eseguibili su chiavi KMS simmetriche e asimmetriche, consulta [Confronto tra chiavi KMS simmetriche e asimmetriche](#).

È possibile utilizzare una chiave KMS con crittografia simmetrica AWS KMS per crittografare, decrittografare e ricrittografare i dati e generare chiavi di dati e coppie di chiavi di dati. Puoi creare chiavi KMS di crittografia simmetrica [multi-regione](#), [importare il materiale della chiave](#) in una chiave KMS di crittografia simmetrica e creare chiavi KMS di crittografia simmetrica negli [archivi delle chiavi personalizzate](#). Per una tabella di confronto delle operazioni eseguibili sulle diverse tipologie di chiavi KMS, consultare [Documentazione di riferimento dei tipi di chiave](#).

Specifiche della chiave RSA

Quando utilizzi una specifica chiave RSA, AWS KMS crea una chiave KMS asimmetrica con una coppia di chiavi RSA. La chiave privata non esce mai non crittografata. AWS KMS È possibile utilizzare la chiave pubblica all'interno AWS KMS o scaricare la chiave pubblica per utilizzarla all'esterno. AWS KMS

Warning

Quando crittografate i dati all'esterno AWS KMS, assicuratevi di poter decriptare il testo cifrato. Se si utilizza la chiave pubblica da una chiave KMS che è stata eliminata da AWS

KMS, la chiave pubblica da una chiave KMS configurata per la firma e la verifica o un algoritmo di crittografia che non è supportato dalla chiave KMS, i dati non sono recuperabili.

In AWS KMS, puoi utilizzare chiavi KMS asimmetriche con coppie di chiavi RSA per la crittografia e la decrittografia, oppure la firma e la verifica, ma non entrambe. Questa proprietà, nota come [Key usage](#), viene determinata separatamente dalla specifica chiave, ma è necessario prendere questa decisione prima di selezionare una specifica chiave.

AWS KMS supporta le seguenti specifiche chiave RSA per la crittografia e la decrittografia o la firma e la verifica:

- RSA_2048
- RSA_3072
- RSA_4096

Le specifiche della chiave RSA differiscono in base alla lunghezza di bit della chiave RSA. La specifica della chiave RSA scelta potrebbe essere determinata dai tuoi standard di sicurezza o dai requisiti della tua attività. In generale, usa la chiave più grande che ritieni pratica e conveniente per la tua attività. Le operazioni crittografiche sulle chiavi KMS con specifiche della chiave RSA diverse hanno prezzi differenti. Per informazioni sui AWS KMS prezzi, consulta la sezione Prezzi del servizio di [gestione delle AWS chiavi](#). Per informazioni sulle quote di richieste, consulta [Quote di richieste](#).

Specifiche della chiave RSA per la crittografia e la decrittografia

Quando una chiave KMS asimmetrica RSA viene utilizzata per la crittografia e la decrittazione, esegui la crittografia con la chiave pubblica e la decrittazione con la chiave privata. Quando richiami l'Encryptoperazione AWS KMS per una chiave RSA KMS, AWS KMS utilizza la chiave pubblica nella coppia di chiavi RSA e l'algoritmo di crittografia specificato per crittografare i dati. Per decrittografare il testo cifrato, richiama l'Decryptoperazione e specifica la stessa chiave KMS e lo stesso algoritmo di crittografia. AWS KMS utilizza quindi la chiave privata nella coppia di chiavi RSA per decrittografare i dati.

Puoi anche scaricare la chiave pubblica e utilizzarla per crittografare i dati all'esterno di AWS KMS. Assicurati di utilizzare un algoritmo di crittografia che AWS KMS supporti le chiavi RSA KMS. Per decrittare il testo cifrato, chiama la funzione Decrypt con la stessa chiave KMS e lo stesso algoritmo di crittografia.

AWS KMS supporta due algoritmi di crittografia per chiavi KMS con specifiche chiave RSA. Questi algoritmi, definiti in [PKCS #1 v2.2](#), differiscono per la funzione hash che usano internamente. [In AWS KMS, gli algoritmi RSAES_OAEP utilizzano sempre la stessa funzione hash sia per scopi di hashing che per la funzione di generazione di maschere \(\)](#). MGF1 Occorre specificare un algoritmo di crittografia quando chiami le azioni [Encrypt](#) e [Decrypt](#). Puoi scegliere un algoritmo diverso per ogni richiesta.

Algoritmi di crittografia supportati per le specifiche della chiave RSA

Algoritmo di crittografia	Descrizione dell'algoritmo
RSAES_OAEP_SHA_1	PKCS #1 v2.2, sezione 7.1. Crittografia RSA con imbottitura OAEP che utilizza SHA-1 sia per la funzione di generazione dell'hash che per quella della maschera insieme a un'etichetta vuota. MGF1
RSAES_OAEP_SHA_256	PKCS #1, sezione 7.1. Crittografia RSA con imbottitura OAEP che utilizza SHA-256 sia per la funzione di generazione dell'hash che per quella della maschera insieme a un'etichetta vuota. MGF1

Non puoi configurare una chiave KMS per utilizzare un particolare algoritmo di crittografia. Tuttavia, puoi utilizzare la condizione [kms: EncryptionAlgorithm](#) policy per specificare gli algoritmi di crittografia che i principali possono utilizzare con la chiave KMS.

Per ottenere gli algoritmi di crittografia per una chiave KMS, [visualizza la configurazione crittografica della](#) chiave KMS nella console o utilizza l'operazione. AWS KMS [DescribeKey](#) AWS KMS fornisce inoltre le specifiche della chiave e gli algoritmi di crittografia quando scarichi la chiave pubblica, nella AWS KMS console o utilizzando l'operazione. [GetPublicKey](#)

Potresti scegliere una specifica della chiave RSA in base alla lunghezza dei dati in chiaro che puoi crittografare in ogni richiesta. Nella tabella che segue vengono illustrate le dimensioni massime, in byte, del testo in chiaro che puoi crittografare in una singola chiamata all'azione [Encrypt](#). I valori differiscono con la specifica della chiave e l'algoritmo di crittografia. Per eseguire un confronto, puoi utilizzare una chiave KMS di crittografia simmetrica per crittografare fino a 4.096 byte contemporaneamente.

Per calcolare la lunghezza massima del testo in chiaro in byte per questi algoritmi, utilizzate la formula seguente: $(key\_size\_in\_bits/8) - (2 * hash\_length\_in\_bits/8) - 2$. Ad esempio, per RSA_2048 con SHA-256, la dimensione massima del testo in chiaro in byte è $(2048/8) - (2 * 256/8) - 2 = 190$.

Dimensione massima del testo in chiaro (in byte) in un'azione Encrypt

Specifica della chiave	Algoritmo di crittografia	
	RSAES_OAEP_SHA_1	RSAES_OAEP_SHA_256
RSA_2048	214	190
RSA_3072	342	318
RSA_4096	470	446

Specifiche della chiave RSA per la firma e la verifica

Quando una chiave KMS asimmetrica RSA viene utilizzata per la firma e la verifica, generi la firma per un messaggio con la chiave privata e verifichi la firma con la chiave pubblica.

Quando richiami l'operazione AWS KMS per una chiave KMS asimmetrica, AWS KMS utilizza la chiave privata nella coppia di chiavi RSA, il messaggio e l'algoritmo di firma specificato per generare una firma. Per verificare la firma, chiamare l'azione [Verify](#). Specificate la firma, più la stessa chiave KMS, lo stesso messaggio e lo stesso algoritmo di firma. AWS KMS utilizza quindi la chiave pubblica nella coppia di chiavi RSA per verificare la firma. Puoi anche scaricare la chiave pubblica e utilizzarla per verificare la firma all'esterno di AWS KMS.

AWS KMS supporta i seguenti algoritmi di firma per tutte le chiavi KMS con una specifica chiave RSA. Devi specificare un algoritmo di firma quando chiami le operazioni [Sign](#) (firma) e [Verify](#) (verifica). Puoi scegliere un algoritmo diverso per ogni richiesta. Quando si firma con coppie di chiavi RSA, sono preferiti gli algoritmi RSASSA-PSS. Includiamo algoritmi RSASSA-PKCS1-v1_5 per la compatibilità con le applicazioni esistenti.

Algoritmi di firma supportati per le specifiche della chiave RSA

Algoritmo di firma	Descrizione dell'algoritmo
RSASSA_PSS_SHA_256	PKCS #1 v2.2, Sezione 8.1, firma RSA con imbottitura PSS che utilizza SHA-256 sia per il digest dei messaggi che per la funzione di generazione della maschera insieme a un sale a 256 bit MGF1
RSASSA_PSS_SHA_384	PKCS #1 v2.2, Sezione 8.1, firma RSA con imbottitura PSS che utilizza SHA-384 sia per il message digest che per la funzione di generazione di maschere insieme a un sale a 384 bit MGF1
RSASSA_PSS_SHA_512	PKCS #1 v2.2, Sezione 8.1, firma RSA con imbottitura PSS che utilizza SHA-512 sia per il message digest che per la funzione di generazione di maschere insieme a un sale a 512 bit MGF1
RSASSA_ _V1_5_SHA_256 PKCS1	PKCS #1 v2.2, sezione 8.2, firma RSA con riempimento PKCS #1v1.5 e SHA-256
RSASSA_PKCS1_ _V1_5_SHA_384	PKCS #1 v2.2, sezione 8.2, firma RSA con riempimento PKCS #1v1.5 e SHA-384
RSASSA_PKCS1_ _V1_5_SHA_512	PKCS #1 v2.2, sezione 8.2, firma RSA con riempimento PKCS #1v1.5 e SHA-512

Non puoi configurare una chiave KMS per utilizzare algoritmi di firma particolari. Tuttavia, puoi utilizzare la condizione [kms: SigningAlgorithm policy per specificare gli algoritmi](#) di firma che i principali possono utilizzare con la chiave KMS.

Per ottenere gli algoritmi di firma per una chiave KMS, [visualizza la configurazione crittografica](#) della chiave KMS nella console o utilizzando l'operazione. AWS KMS [DescribeKey](#) AWS KMS fornisce

anche le specifiche della chiave e gli algoritmi di firma quando scarichi la chiave pubblica, nella AWS KMS console o utilizzando l'operazione. [GetPublicKey](#)

Specifiche della chiave basata su curva ellittica

Quando si utilizza una specifica chiave a curva ellittica (ECC), AWS KMS crea una chiave KMS asimmetrica con una coppia di chiavi ECC per la firma e la verifica o la derivazione di segreti condivisi (ma non entrambi). La chiave privata che genera firme o ricava segreti condivisi non esce mai non crittografata. AWS KMS È possibile utilizzare la chiave pubblica per [verificare le firme](#) all'interno AWS KMS o [scaricare la chiave pubblica](#) per utilizzarla all'esterno. AWS KMS

Quando si utilizza una specifica chiave Edwards Curve, AWS KMS crea una chiave KMS asimmetrica con una coppia di chiavi Ed25519 per la firma e la verifica. La chiave privata che genera le firme non esce mai non crittografata. AWS KMS È possibile utilizzare la chiave pubblica per [verificare le firme](#) all'interno AWS KMS o [scaricare la chiave pubblica](#) per utilizzarla all'esterno. AWS KMS

AWS KMS supporta le seguenti specifiche chiave ECC per le chiavi KMS asimmetriche.

- Coppie di chiavi asimmetriche a curva ellittica standard NIST (firma e verifica, o derivazione di segreti condivisi)
 - ECC_NIST_P256 (secp256r1)
 - ECC_NIST_P384 (secp384r1)
 - ECC_NIST_P521 (secp521r1)
 - EDWARDS25519 ECC_NIST_ (ed25519) - solo firma e verifica
 - Nota: per le chiavi EDWARDS25519 KMS ECC_NIST_, è necessario l'algoritmo di firma _SHA_512, mentre _PH_SHA_512 richiede. ED25519 [MessageType:RAW](#) [ED25519MessageType:DIGEST](#) Questi tipi di messaggi non possono essere usati in modo intercambiabile.
- Altre coppie di chiavi asimmetriche basate su curva ellittica (firma e verifica)
 - ECC_SECG_P256K1 ([secp256k1](#)), comunemente usate per le criptovalute.

La specifica della chiave ECC scelta potrebbe essere determinata dai tuoi standard di sicurezza o dai requisiti della tua attività. In generale, utilizza la curva con il maggior numero di punti che ritieni pratica e conveniente per la tua attività.

Se state creando una chiave KMS asimmetrica per [derivare segreti condivisi](#), utilizzate una delle specifiche chiave a curva ellittica standard del NIST (ecc_SECG_P256K1 e ECC_NIST_).

EDWARDS25519 [L'unico algoritmo di accordo delle chiavi supportato per derivare segreti condivisi è l'Elliptic Curve Cryptography Cofactor Diffie-Hellman Primitive \(ECDH\)](#). Per un esempio di come derivare segreti condivisi offline, vedi. [the section called “Ricevere segreti condivisi offline”](#)

Se stai creando una chiave KMS asimmetrica da utilizzare con le criptovalute, usa la specifica della chiave ECC_SECG_P256K1. Puoi utilizzare anche questa specifica della chiave per altri scopi, ma è necessaria per il Bitcoin e le altre criptovalute.

La tabella seguente mostra gli algoritmi di firma AWS KMS supportati per ciascuna delle specifiche chiave ECC. Non puoi configurare una chiave KMS per utilizzare algoritmi di firma particolari. Tuttavia, puoi utilizzare la condizione [kms: SigningAlgorithm](#) policy per specificare gli algoritmi di firma che i principali possono utilizzare con la chiave KMS.

Algoritmi di firma supportati per le specifiche della chiave ECC

Specifica della chiave	Algoritmo di firma	Descrizione dell'algoritmo
ECC_NIST_P256	ECDSA_SHA_256	NIST FIPS 186-4, Section 6.4, ECDSA signature using the curve specified by the key and SHA-256 per il digest del messaggio.
ECC_NIST_P384	ECDSA_SHA_384	NIST FIPS 186-4, Section 6.4, ECDSA signature using the curve specified by the key and SHA-384 per il message digest.
ECC_NIST_P521	ECDSA_SHA_512	NIST FIPS 186-4, Section 6.4, ECDSA signature using the curve specified by the key and SHA-512 per il message digest.
ECC_SECG_P256K1	ECDSA_SHA_256	NIST FIPS 186-4, Section 6.4, ECDSA signature using the curve specified by the key

Specifica della chiave	Algoritmo di firma	Descrizione dell'algoritmo
		and SHA-256 per il message digest.
ECC_NIST_EDWARDS25519	ED25519_SHA_512	NIST FIPS 186-5, Section 7, EdDSA signature using the curve specified by the key and SHA-512 per il digest del messaggio. KMS richiede questo MessageType:RAW algoritmo.
ECC_NIST_EDWARDS25519	ED25519_PH_SHA_512	NIST FIPS 186-5, Section 7, EdDSA signature using the curve specified by the key and SHA-512 per il digest del messaggio. KMS richiede questo MessageType:DIGEST algoritmo.

Specifiche della chiave per le chiavi KMS HMAC

AWS KMS supporta chiavi HMAC simmetriche di diverse lunghezze. La scelta della specifica della chiave può dipendere da requisiti normativi, di sicurezza o aziendali. La lunghezza della chiave determina l'algoritmo MAC utilizzato nelle operazioni e nelle operazioni. [GenerateMacVerifyMac](#)

In generale, le chiavi più lunghe sono più sicure. Usa la chiave più lunga funzionale per il tuo caso d'uso.

Specifica della chiave HMAC	Algoritmo MAC
HMAC_224	HMAC_SHA_224
HMAC_256	HMAC_SHA_256
HMAC_384	HMAC_SHA_384
HMAC_512	HMAC_SHA_512

Specifiche principali di ML-DSA

Una chiave ML-DSA è una chiave crittografica utilizzata nel Module-Lattice-Based Digital Signature Algorithm (ML-DSA), progettato per la crittografia post-quantistica. [Questo algoritmo fa parte dello sforzo di standardizzazione del NIST \(National Institute of Standards and Technology\), specificamente delineato nei Federal Information Processing Standards \(FIPS\) 204.](#)

Le chiavi ML-DSA vengono utilizzate in un sistema di key pair pubblico-privato. La chiave privata viene utilizzata per firmare i dati, mentre la chiave pubblica viene utilizzata per verificare la firma. Questo sistema garantisce l'autenticità, l'integrità e il non ripudio di messaggi o documenti digitali, anche di fronte a potenziali minacce informatiche quantistiche.

Quando si crea una chiave con le specifiche della chiave ML-DSA, AWS KMS crea una chiave KMS asimmetrica con una coppia di chiavi ML-DSA per la firma e la verifica. La chiave privata che genera le firme non esce mai non crittografata. AWS KMS È possibile utilizzare la chiave pubblica per [verificare le firme](#) all'interno AWS KMS o [scaricare la chiave pubblica](#) per utilizzarla all'esterno. AWS KMS

AWS KMS supporta le seguenti specifiche chiave ML-DSA per le chiavi KMS asimmetriche:

- ML_DSA_44
- ML_DSA_65
- ML_DSA_87

AWS KMS supporta l'algoritmo di firma ML_DSA_SHAKE_256 per tutte le specifiche chiave di ML-DSA.

SM2 specifiche chiave (solo regioni della Cina)

La specifica SM2 chiave è una specifica chiave a curva ellittica definita all'interno della GM/T serie di specifiche pubblicate dall'[Office of State Commercial Cryptography Administration \(OSCCA\) cinese](#). Le specifiche SM2 chiave sono disponibili solo nelle regioni della Cina. Quando si utilizza la specifica SM2 chiave, AWS KMS crea una chiave KMS asimmetrica con una coppia di chiavi. SM2 Puoi usare la tua coppia di SM2 chiavi all'interno AWS KMS o scaricare la chiave pubblica per utilizzarla all'esterno AWS KMS. Per ulteriori informazioni, consulta [the section called “Verifica offline con coppie di SM2 chiavi \(solo regioni della Cina\)”](#).

Ogni chiave KMS può averne solo una [Key usage](#). Puoi utilizzare una chiave SM2 KMS per la firma e la verifica, la crittografia e la decrittografia o la derivazione di segreti condivisi. È necessario

specificare il parametro sull'utilizzo della chiave in fase di creazione della chiave KMS e non può essere modificata in seguito.

Se stai creando una chiave KMS asimmetrica per [derivare](#) segreti condivisi, usa le specifiche della chiave. SM2 L'unico algoritmo di accordo chiave supportato per derivare segreti condivisi è l'[Elliptic Curve Cryptography Cofactor Diffie-Hellman Primitive](#) (ECDH).

AWS KMS supporta SM2 i seguenti algoritmi di crittografia e firma:

- SM2algoritmo di crittografia PKE

SM2PKE è un algoritmo di crittografia basato su curve ellittiche definito dall'OSCCA nel 0003.4-2012. GM/T

- SM2Algoritmo di firma DSA

SM2DSA è un algoritmo di firma basato su curve ellittiche definito dall'OSCCA nel 0003.2-2012. GM/T SM2DSA richiede un ID distintivo sottoposto a hash con l'algoritmo di SM3 hashing e quindi combinato con il messaggio, o digest del messaggio, a cui è stato passato. AWS KMS Questo valore concatenato viene quindi sottoposto a hash e firmato da. AWS KMS

AWS KMS autorizzazioni

Questa tabella è progettata per aiutarti a comprendere AWS KMS le autorizzazioni in modo da poter controllare l'accesso alle tue risorse. AWS KMS Le definizioni delle intestazioni di colonna vengono visualizzate sotto la tabella.

Per ulteriori informazioni sulle AWS KMS autorizzazioni, consulta la sezione [Azioni, risorse e chiavi di condizione relativa](#) all' AWS Key Management Service argomento del Service Authorization Reference. Tuttavia, questo argomento non riporta tutte le chiavi di condizione che possono essere utilizzate per rifinire ogni autorizzazione.

Per ulteriori informazioni su quali AWS KMS operazioni sono valide per le chiavi KMS con crittografia simmetrica, le chiavi KMS asimmetriche e le chiavi KMS HMAC, consulta la. [Documentazione di riferimento dei tipi di chiave](#)

Note

Potrebbe essere necessario scorrere orizzontalmente o verticalmente per visualizzare tutti i dati della tabella.

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
CancelKeyDeletion kms:CancelKeyDeletion	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService
ConnectCustomKeyStore kms:ConnectCustomKeyStore	Policy IAM	No	*	km: CallerAccount
CreateAlias kms:CreateAlias Per utilizzare questa operazione, il chiamante necessita dell'auto	Policy IAM (per l'alias) Policy chiave (per	No No	Alias Chiave KMS	Nessuna (in caso di controllo dell'accesso all'alias) Condizioni per le operazioni delle chiavi KMS:

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
autorizzazione kms:CreateAlias su due risorse: - L'alias (in una policy IAM) - La chiave KMS (in una policy chiave) Per informazioni dettagliate, consultare Controllo dell'accesso agli alias.	la chiave KMS)			km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService
CreateCustomKeyStore kms:CreateCustomKeyStore	Policy IAM	No	*	km: CallerAccount

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
CreateGrant kms:CreateGrant	Policy della chiave	Sì	Chiave KMS	Condizioni per il contesto di crittografia: kms:: chiave contestuale EncryptionContext km: EncryptionContextKeys Condizioni di concessione: km: GrantConstraintType km: GranteePrincipal km: GrantsForAWSResource km: GrantOperations km: RetiringPrincipal Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
				km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService
CreateKey kms:CreateKey	Policy IAM	No	*	km: BypassPolicyLockoutSafetyCheck km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ViaService aws:RequestTag/tag-key (chiave di condizione AWS globale) aws:ResourceTag/tag-key (chiave di condizione globale)AWS aws: TagKeys (chiave di condizione AWS globale)

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
Decrypt kms:Decrypt	Policy della chiave	Sì	Chiave KMS	Condizioni per le operazioni di crittografia km: EncryptionAlgorithm km: RequestAlias Condizioni per il contesto di crittografia: kms:: chiave contestuale EncryptionContext km: EncryptionContextKeys Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale)

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
				km: ViaService
DeleteAlias <code>kms:DeleteAlias</code>	Policy IAM (per l'alias)	No	Alias	Nessuna (in caso di controllo dell'accesso all'alias)
Per utilizzare questa operazione, il chiamante necessita dell'autorizzazione <code>kms:DeleteAlias</code> su due risorse: - L'alias (in una policy IAM) - La chiave KMS (in una policy chiave) Per informazioni dettagliate, consultare Controllo dell'accesso agli alias.	Policy chiave (per la chiave KMS)	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService
DeleteCustomKeyStore <code>kms:DeleteCustomKeyStore</code>	Policy IAM	No	*	km: CallerAccount

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
DeleteImportedKeyMaterial kms:DeleteImportedKeyMaterial	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
DeriveSharedSecret kms:DeriveSharedSecret	Policy della chiave	Sì	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService Condizioni per le operazioni di crittografia: km: KeyAgreementAlgorithm
DescribeCustomKeyStores kms:DescribeCustomKeyStores	Policy IAM	No	*	km: CallerAccount

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
DescribeKey kms:DescribeKey	Policy della chiave	Sì	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService Altre condizioni: km: RequestAlias

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
DisableKey kms:DisableKey	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
DisableKeyRotation <code>kms:DisableKeyRotation</code>	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService
DisconnectCustomKeyStore <code>kms:DisconnectCustomKeyStore</code>	Policy IAM	No	*	km: CallerAccount

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
EnableKey kms:EnableKey	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
EnableKeyRotation kms:EnableKeyRotation	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService Condizioni di rotazione automatica dei tasti: km: RotationPeriodInDays

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
Encrypt kms:Encrypt	Policy della chiave	Sì	Chiave KMS	Condizioni per le operazioni di crittografia km: EncryptionAlgorithm km: RequestAlias Condizioni per il contesto di crittografia: kms:: chiave contestuale EncryptionContext km: EncryptionContextKeys Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale)

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione

km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
GenerateDataKey kms:GenerateDataKey	Policy della chiave	Sì	Chiave KMS	Condizioni per le operazioni di crittografia km: EncryptionAlgorithm km: RequestAlias Condizioni per il contesto di crittografia: kms:: chiave contestuale EncryptionContext km: EncryptionContextKeys Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale)

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
				km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
GenerateDataKeyPair <code>kms:GenerateDataKeyPair</code>	Policy della chiave	Sì	Chiave KMS Genera coppie di chiavi di dati asimmetri che protette da una chiave KMS di crittografia simmetrica.	Condizioni per coppie di chiavi di dati: km: DataKeyPairSpec Condizioni per le operazioni di crittografia km: EncryptionAlgorithm km: RequestAlias Condizioni per il contesto di crittografia: kms:: chiave contestuale EncryptionContext km: EncryptionContextKeys Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
				km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
GenerateDataKeyPairWithoutPlaintext <code>kms:GenerateDataKeyPairWithoutPlaintext</code>	Policy della chiave	Sì	Chiave KMS Genera coppie di chiavi di dati asimmetriche protette da una chiave KMS di crittografia simmetrica.	Condizioni per coppie di chiavi di dati: km: DataKeyPairSpec Condizioni per le operazioni di crittografia km: EncryptionAlgorithm km: RequestAlias Condizioni per il contesto di crittografia: kms:: chiave contestuale EncryptionContext km: EncryptionContextKeys Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
				km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
GenerateDataKeyWithoutPlaintext <code>kms:GenerateDataKeyWithoutPlaintext</code>	Policy della chiave	Sì	Chiave KMS	Condizioni per le operazioni di crittografia km: EncryptionAlgorithm km: RequestAlias Condizioni per il contesto di crittografia: kms:: chiave contestuale EncryptionContext km: EncryptionContextKeys Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale)

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
				km: ViaService
GenerateMac kms:GenerateMac	Policy della chiave	Sì	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService Condizioni per le operazioni di crittografia: km: MacAlgorithm km: RequestAlias
GenerateRandom kms:GenerateRandom	Policy IAM	N/D	*	Nessuno

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
GetKeyPolicy kms:GetKeyPolicy	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
GetKeyRotationStatus kms:GetKeyRotationStatus	Policy della chiave	Sì	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
GetParametersForImport <code>kms:GetParametersForImport</code>	Policy della chiave	No	Chiave KMS	km: WrappingAlgorithm km: WrappingKeySpec Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
GetPublicKey kms:GetPublicKey	Policy della chiave	Sì	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService Altre condizioni: km: RequestAlias

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
ImportKeyMaterial <code>kms:ImportKeyMaterial</code>	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService Altre condizioni: km: ExpirationModel km: ValidTo
ListAliases <code>kms:ListAliases</code>	Policy IAM	No	*	Nessuno

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
ListGrants kms:ListGrants	Policy della chiave	Sì	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService Altre condizioni: km: GrantsForAWSResource

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
ListKeyPolicies kms:ListKeyPolicies	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
ListKeyRotations kms:ListKeyRotations	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService
ListKeys kms:ListKeys	Policy IAM	No	*	Nessuno

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
ListResourceTags <code>kms:ListResourceTags</code>	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
ListRetirableGrants <code>kms:ListRetirableGrants</code>	Policy IAM	L'entità principale specifica che deve trovarsi nell'account locale, ma l'operazione restituisce concessioni in tutti gli account.	*	Nessuno

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
PutKeyPolicy kms:PutKeyPolicy	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService Altre condizioni: km: BypassPolicyLockoutSafetyCheck

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
ReEncrypt <code>kms:ReEncryptFrom</code> <code>kms:ReEncryptTo</code> Per utilizzare questa operazione, il chiamante necessita dell'autorizzazione per due chiavi KMS: <code>kms:ReEncryptFrom</code> sulla chiave KMS usata per la decrittografia <code>kms:ReEncryptTo</code> sulla chiave KMS utilizzata per la crittografia	Policy della chiave	Sì	Chiave KMS	Condizioni per le operazioni di crittografia km: EncryptionAlgorithm km: RequestAlias Condizioni per il contesto di crittografia: kms:: chiave contestuale EncryptionContext km: EncryptionContextKeys Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale)

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
				km: ViaService Altre condizioni: km: ReEncryptOnSameKey
ReplicateKey kms:ReplicateKey Per utilizzare questa operazione, il chiamante necessita delle seguenti autorizzazioni: • kms:ReplicateKey sulla chiave primaria multiregione • kms:CreateKey in una policy IAM nella Regione di replica	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService Altre condizioni: km: ReplicaRegion

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
RetireGrant <code>kms:RetireGrant</code> L'autorizzazione per ritirare una concessione è determinata principalmente dalla concessione. Una policy da sola non può consentire l'accesso a questa operazione. Per ulteriori informazioni, consulta Ritirare e revocare le concessioni.	Policy IAM Questa autorizzazione non è valida in una policy chiave.	Sì	Chiave KMS	Condizioni per il contesto di crittografia: kms:: chiave contestuale EncryptionContext km: EncryptionContextKeys Condizioni di concessione: km: GrantConstraintType Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
RevokeGrant kms:RevokeGrant	Policy della chiave	Sì	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService Altre condizioni: km: GrantIsForAWSResource

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
RotateKeyOnDemand kms:RotateKeyOnDemand	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
ScheduleKeyDeletion kms:ScheduleKeyDeletion	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
Sign kms:Sign	Policy della chiave	Sì	Chiave KMS	Condizioni per la firma e la verifica: km: MessageType km: RequestAlias km: SigningAlgorithm Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
TagResource kms:TagResource	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService Condizioni per l'assegnazione di tag: aws:RequestTag/tag-key (chiave di condizione AWS globale) aws: TagKeys (chiave di condizione AWS globale)

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
UntagResource kms:UntagResource	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService Condizioni per l'assegnazione di tag: aws:RequestTag/tag-key (chiave di condizione AWS globale) aws: TagKeys (chiave di condizione AWS globale)

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
UpdateAlias kms:UpdateAlias	Policy IAM (per l'alias)	No	Alias	Nessuna (in caso di controllo dell'accesso all'alias)
Per utilizzare questa operazione, il chiamante necessita dell'autorizzazione kms:UpdateAlias su tre risorse: • L'alias • La chiave KMS attualmente associata • La chiave KMS appena associata Per informazioni dettagliate, consultare Controllo dell'accesso agli alias.	Policy chiave (per le chiavi KMS)	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService
UpdateCustomKeyStore kms:UpdateCustomKeyStore	Policy IAM	No	*	km: CallerAccount

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
UpdateKeyDescription kms:UpdateKeyDescription	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
UpdatePrimaryRegion <code>kms:UpdatePrimaryRegion</code> Per utilizzare questa operazione, il chiamante necessita dell'autorizzazione <code>kms:UpdatePrimaryRegion</code> sulla chiave primaria multiregione, che diventerà una chiave di replica, e sulla chiave di replica multiregione che diventerà la chiave primaria.	Policy della chiave	No	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService Altre condizioni: km: PrimaryRegion

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
Verify <code>kms:Verify</code>	Policy della chiave	Sì	Chiave KMS	Condizioni per la firma e la verifica: km: MessageType km: RequestAlias km: SigningAlgorithm Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService

Azioni e autorizzazioni	Tipo di policy	Utilizzo per più account	Risorse (per policy IAM)	AWS KMS chiavi di condizione
VerifyMac kms:VerifyMac	Policy della chiave	Sì	Chiave KMS	Condizioni per le operazioni delle chiavi KMS: km: CallerAccount km: KeySpec km: KeyUsage km: KeyOrigin km: MultiRegion km: MultiRegionKeyType km: ResourceAliases aws:ResourceTag/tag-key (chiave di condizione AWS globale) km: ViaService Condizioni per le operazioni di crittografia: km: MacAlgorithm km: RequestAlias

Descrizioni delle colonne

Le colonne nella tabella forniscono le seguenti informazioni:

- Azioni e autorizzazioni elenca ogni operazione AWS KMS API e l'autorizzazione che consente l'operazione. È possibile specificare l'operazione nell'elemento `Action` di un'istruzione di policy.

- Tipo di policy indica se l'autorizzazione può essere utilizzata in una policy chiave o in una policy IAM.

Policy chiave significa che puoi specificare l'autorizzazione nella policy chiave. Quando la policy chiave contiene l'[istruzione di policy che consente policy IAM](#), puoi specificare l'autorizzazione in una policy IAM.

Policy IAM significa che puoi specificare l'autorizzazione solo in una policy IAM.

- Utilizzo tra account mostra le operazioni che gli utenti autorizzati possono eseguire sulle risorse in un Account AWS diverso.

Un valore di Sì significa che le entità principali possono eseguire l'operazione sulle risorse in un Account AWS diverso.

Un valore di No significa che le entità principali possono eseguire l'operazione solo sulle risorse nel proprio Account AWS.

Se si concede a un'entità in un account diverso un'autorizzazione che non può essere utilizzata su una risorsa tra account, l'autorizzazione non è valida. Ad esempio, se concedi a un responsabile di un altro account [kms](#): l'TagResourceautorizzazione a utilizzare una chiave KMS nel tuo account, i suoi tentativi di taggare la chiave KMS nel tuo account falliranno.

- Resources elenca le AWS KMS risorse a cui si applicano le autorizzazioni. AWS KMS supporta due tipi di risorse: una chiave KMS e un alias. In una policy delle chiavi, il valore dell'elemento Resource è sempre *, che indica la chiave KMS collegata alla policy delle chiavi.

Utilizza i seguenti valori per rappresentare una AWS KMS risorsa in una policy IAM.

Chiave KMS

Quando la risorsa è una chiave KMS utilizza l'[ARN chiave](#). Per assistenza, consulta [the section called “Trova l'ID della chiave e l'ARN della chiave”](#).

`arn:AWS_partition_name:kms:AWS_Region:AWS_account_ID:key/key_ID`

Per esempio:

`arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`

Alias

Quando la risorsa è un alias, utilizza l'[ARN dell'alias](#). Per assistenza, consulta [the section called "Trova il nome dell'alias e l'alias ARN"](#).

`arn:AWS_partition_name:kms:AWS_region:AWS_account_ID:alias/alias_name`

Per esempio:

`arn:aws:kms:us-west-2:111122223333:alias/ExampleAlias`

* (asterisco)

Quando l'autorizzazione non si applica a una determinata risorsa (chiave KMS o alias) utilizza un asterisco (*).

In una policy IAM per un' AWS KMS autorizzazione, un asterisco nell'elemento indica tutte le risorse (chiavi e alias KMS). Resource AWS KMS Puoi anche utilizzare un asterisco nell'Resourceelemento quando l' AWS KMS autorizzazione non si applica a particolari chiavi o alias KMS. Ad esempio, quando si consente o si `kms:CreateKey` nega l'`kms:ListKeys` autorizzazione, è necessario impostare l'elemento su. Resource *

- AWS KMS condition keys elenca le chiavi di AWS KMS condizione che è possibile utilizzare per controllare l'accesso all'operazione. Puoi specificare condizioni nell'elemento Condition di una policy. Per ulteriori informazioni, consulta [AWS KMS tasti di condizione](#). Questa colonna include anche [le chiavi di condizione AWS globali](#) supportate da AWS KMS, ma non da tutti i AWS servizi.

AWS KMS operazioni interne

AWS Key Management Service (AWS KMS) fornisce chiavi e operazioni crittografiche protette da [moduli di sicurezza hardware \(HSM\) convalidati FIPS 140-3 di livello di sicurezza 3 e scalati](#) per il cloud. AWS KMS le chiavi e le funzionalità vengono utilizzate da più servizi AWS cloud e puoi utilizzarle per proteggere i dati nelle tue applicazioni. Questa guida tecnica fornisce dettagli sulle operazioni crittografiche che vengono eseguite AWS durante l'utilizzo AWS KMS.

AWS KMS i componenti interni sono necessari per garantire la scalabilità e la sicurezza di un HSMs servizio di gestione delle chiavi distribuito a livello globale.

Argomenti

- [Domini e stato del dominio](#)

- [Sicurezza delle comunicazioni interne](#)
- [Processo di replica per chiavi multi-regione](#)
- [Protezione della durabilità](#)

Domini e stato del dominio

Una raccolta cooperativa di AWS KMS entità interne affidabili all'interno di un Regione AWS viene definita dominio. Un dominio include un set di entità attendibili, un insieme di regole e un set di chiavi segrete, chiamate chiavi di dominio. Le chiavi di dominio sono condivise tra HSMs i membri del dominio. Uno stato di dominio è costituito dai seguenti campi:

Nome

Un nome di dominio per identificare questo dominio.

Membri

Un elenco di HSMs questi sono membri del dominio, inclusa la chiave di firma pubblica e le chiavi di accordo pubblico.

Operatori

Un elenco di entità, chiavi di firma pubbliche e un ruolo (AWS KMS operatore o host del servizio) che rappresenta gli operatori di questo servizio.

Regolamento

Un elenco di regole quorum per ogni comando che deve essere soddisfatto per eseguire un comando sull'HSM.

Chiavi di dominio

Un elenco di chiavi di dominio (chiavi simmetriche) attualmente in uso all'interno del dominio.

Lo stato completo del dominio è disponibile solo sull'HSM. Lo stato del dominio viene sincronizzato tra i membri del dominio dell'HSM come token di dominio esportato.

Chiavi di dominio

Tutti gli utenti HSMs di un dominio condividono un set di chiavi di dominio, $\{DK_r\}$. Queste chiavi vengono condivise tramite una routine di esportazione dello stato del dominio. Lo stato del dominio esportato può essere importato in qualsiasi HSM membro del dominio.

L'insieme di chiavi di dominio, $\{DK_r\}$, include sempre una chiave di dominio attiva e diverse chiavi di dominio disattivate. Le chiavi di dominio vengono ruotate giornalmente per garantire che siano AWS conformi alla [Raccomandazione per la gestione delle chiavi - Parte 1](#). Durante la rotazione della chiave di dominio, tutte le chiavi KMS crittografate nella chiave di dominio in uscita vengono nuovamente crittografate con la nuova chiave di dominio attiva. La chiave di dominio attiva viene utilizzata per crittografare qualsiasi nuova chiave. EKTs Le chiavi di dominio scadute possono essere utilizzate solo per decrittografare le chiavi di dominio precedentemente crittografate EKTs per un numero di giorni equivalente al numero di chiavi di dominio ruotate di recente.

Token di dominio esportati

Esiste una normale necessità di sincronizzare lo stato tra i partecipanti al dominio. Ciò avviene esportando lo stato del dominio ogni volta che viene apportata una modifica al dominio. Lo stato del dominio viene esportato come token di dominio esportato.

Nome

Un nome di dominio per identificare questo dominio.

Membri

Un elenco di quelli HSMs che sono membri del dominio, comprese le relative chiavi pubbliche per la firma e l'accordo.

Operatori

Un elenco di entità, chiavi di firma pubbliche e un ruolo che rappresenta gli operatori di questo servizio.

Regolamento

Un elenco di regole quorum per ogni comando che deve essere soddisfatto per eseguire un comando su un membro del dominio dell'HSM.

Chiavi di dominio crittografate

Chiavi di dominio crittografate con envelope. Le chiavi di dominio vengono crittografate dal membro firmatario per ciascuno dei membri elencati sopra, con envelope nella chiave di accordo pubblico.

Firma

Una firma sullo stato del dominio prodotto da un HSM, necessariamente un membro del dominio che ha esportato lo stato del dominio.

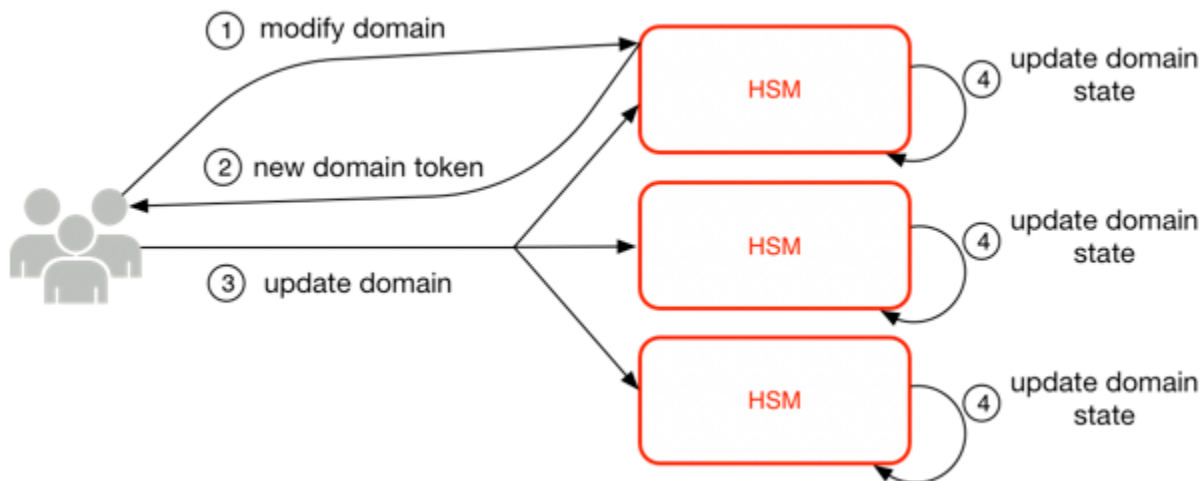
Il token di dominio esportato costituisce la base dell'attendibilità per le entità che operano all'interno del dominio.

Gestione degli stati del dominio

Lo stato del dominio viene gestito tramite comandi autenticati con quorum. Queste modifiche includono la modifica dell'elenco dei partecipanti attendibili nel dominio, la modifica delle regole del quorum per l'esecuzione dei comandi HSM e la rotazione periodica delle chiavi di dominio. Questi comandi vengono autenticati in base al comando anziché alle operazioni di sessione autenticate, come illustrato nell'immagine seguente.

Nel suo stato inizializzato e operativo, un HSM contiene un set di chiavi di identità asimmetriche auto-generate, una coppia di chiavi di firma e una coppia di chiavi per la creazione delle chiavi. Tramite un processo manuale, un AWS KMS operatore può stabilire un dominio iniziale da creare su un primo HSM in una regione. Questo dominio iniziale è costituito da uno stato di dominio completo, come definito in precedenza in questo argomento. Viene installato tramite un comando join su ciascuno dei membri HSM definiti nel dominio.

Dopo che un HSM ha aderito a un dominio iniziale, è legato alle regole definite in quel dominio. Queste regole definiscono i comandi che utilizzano le chiavi crittografiche del cliente o apportano modifiche allo stato dell'host o del dominio. Le operazioni dell'API di sessione autenticate che utilizzano le chiavi crittografiche sono state definite in precedenza.



L'immagine precedente mostra come viene modificato uno stato di dominio. Il processo è costituito da quattro fasi:

1. Un comando basato su quorum viene inviato a un HSM per modificare il dominio.

2. Un nuovo stato di dominio viene generato ed esportato come nuovo token di dominio esportato. Lo stato sull'HSM non viene modificato, il che significa che la modifica non viene promulgata sull'HSM.
3. Un secondo comando viene inviato a ciascuno dei membri del HSMs token di dominio appena esportato per aggiornare lo stato del dominio con il nuovo token di dominio.
4. Gli HSMs elementi elencati nel nuovo token di dominio esportato possono autenticare il comando e il token di dominio. Possono anche decomprimere le chiavi del dominio per aggiornare lo stato del dominio su tutti gli HSMs elementi del dominio.

HSMs non comunicano direttamente tra loro. Invece, un quorum di operatori richiede una modifica allo stato del dominio che si traduce in un nuovo token di dominio esportato. Un membro host del servizio del dominio viene utilizzato per distribuire il nuovo stato del dominio a tutti gli HSM del dominio.

L'abbandono e l'unione di un dominio vengono eseguiti tramite le funzioni di gestione HSM. La modifica dello stato del dominio avviene tramite le funzioni di gestione del dominio.

Abbandona dominio

Fa sì che un HSM lasci un dominio, eliminando dalla memoria tutti i residui e le chiavi di quel dominio.

Unisci dominio

Fa sì che un HSM si unisca a un nuovo dominio o aggiorni lo stato corrente del dominio al nuovo stato del dominio. Il dominio esistente viene utilizzato come origine del set iniziale di regole per autenticare questo messaggio.

Crea dominio

Provoca la creazione di un nuovo dominio su un HSM. Restituisce un primo token di dominio che può essere distribuito ai membri HSMs del dominio.

Modifica operatori

Aggiunge o rimuove gli operatori dall'elenco degli operatori autorizzati e i relativi ruoli nel dominio.

Modifica membri

Aggiunge o rimuove un HSM dall'elenco degli autorizzati HSMs nel dominio.

Modifica regole

Modifica il set di regole quorum necessarie per eseguire i comandi su un HSM.

Ruota chiavi di dominio

Fa sì che una nuova chiave di dominio venga creata e contrassegnata come chiave di dominio attiva. Questo sposta la chiave attiva esistente su una chiave disattivata e rimuove la chiave disattivata più vecchia dallo stato del dominio.

Sicurezza delle comunicazioni interne

I comandi tra gli host o AWS KMS gli operatori del servizio e il HSMs sono protetti tramite due meccanismi illustrati in [Sessioni autenticate](#): un metodo di richiesta firmato dal quorum e una sessione autenticata che utilizza un protocollo host del servizio HSM.

I comandi firmati dal quorum sono progettati in modo che nessun singolo operatore possa modificare le protezioni di sicurezza critiche che forniscono. HSMs I comandi eseguiti sulle sessioni autenticate garantiscono che solo gli operatori del servizio autorizzati possano eseguire operazioni relative alle chiavi KMS. Tutte le informazioni segrete relative al cliente sono protette in tutta l'infrastruttura. AWS

Creazione delle chiavi

Per proteggere le comunicazioni interne, AWS KMS utilizza due diversi metodi di definizione delle chiavi. Il primo è definito come C (1, 2, ECC DH) nella [Raccomandazione per schemi di definizione di chiavi a coppie che utilizzano la crittografia a logaritmi discreti](#). Questo schema ha un iniziatore con una chiave di firma statica. L'iniziatore genera e firma una chiave sulla curva ellittica Diffie-Hellman (ECDH) effimera, per un destinatario con una chiave di accordo ECDH statica. Questo metodo utilizza una chiave effimera e due chiavi statiche con ECDH. Questa è la derivazione dell'etichetta C(1, 2, ECC DH). Il metodo è talvolta chiamato ECDH a un passaggio.

Il secondo metodo per la creazione di una chiave è [C\(2, 2, ECC, DH\)](#). In questo schema, entrambe le parti hanno una chiave di firma statica e generano, firmano e scambiano una chiave ECDH effimera. Questo metodo utilizza due chiavi statiche e due chiavi effimere, ognuna con ECDH. Questa è la derivazione dell'etichetta C(2, 2, ECC DH). Questo metodo è talvolta chiamato ECDH effimero o ECDHE. Tutte le chiavi ECDH vengono generate sulla curva secp384r1 (NIST-P384).

Limite di sicurezza HSM

Il limite di sicurezza interno di è l'HSM. AWS KMS L'HSM ha un'interfaccia proprietaria e nessun'altra interfaccia fisica attiva nel suo stato operativo. Durante l'inizializzazione viene eseguito il provisioning di un HSM operativo con le chiavi di crittografia necessarie per stabilire il proprio ruolo nel dominio. I materiali crittografici sensibili dell'HSM vengono archiviati nella memoria volatile e sono cancellati solo quando il modulo HSM non è in stato operativo, inclusi arresti o ripristini previsti o non intenzionali.

Le operazioni API HSM vengono autenticate da singoli comandi o tramite una sessione riservata autenticata reciprocamente stabilita da un host di servizio.

Comandi firmati con quorum

I comandi firmati dal quorum vengono emessi dagli operatori a. HSMs In questa sezione viene descritto come i comandi basati su quorum vengono creati, firmati e autenticati. Queste regole sono abbastanza semplici. Ad esempio, per essere autenticato il comando Foo richiede due membri dal ruolo Bar. Per la creazione e la verifica di un comando basato su quorum sono necessari tre passaggi. Il primo passo è la creazione iniziale del comando, il secondo è l'invio ad operatori aggiuntivi per la firma e il terzo è la verifica e l'esecuzione.

Per introdurre i concetti, supponiamo che esista un set autentico di chiavi e ruoli pubblici dell'operatore $\{QOS_s\}$ e un insieme di regole di quorum $QR = \{Command_i, Rule_{\{i, t\}}\}$ in cui ogni regola è un insieme di ruoli e un numero minimo $N \{Role, N\}$. t Affinché un comando soddisfi la regola del quorum, il set di dati dei comandi deve essere firmato da un set di operatori elencati in $\{QOS_s\}$ in modo che soddisfino una delle regole elencate per quel comando. Come accennato in precedenza, l'insieme di regole del quorum e degli operatori viene memorizzato nello stato del dominio e nel token di dominio esportato.

In pratica, un firmatario iniziale firma il comando $Sig_1 = Sign(dO_{p1}, Command)$. Anche un secondo operatore firma il comando $Sig_2 = Sign(dO_{p2}, Command)$. Il messaggio doppiamente firmato viene inviato a un HSM per l'esecuzione. L'HSM completa le seguenti attività:

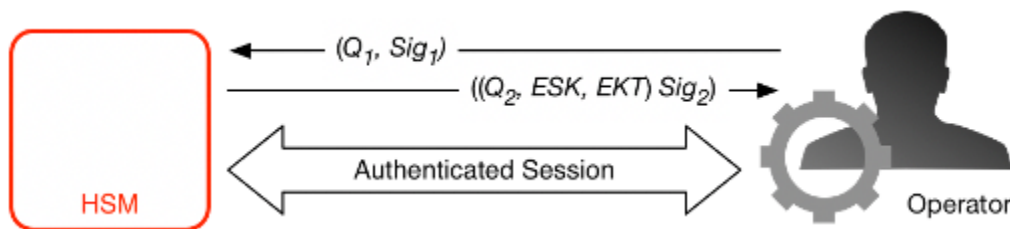
1. Per ogni firma, estrae la chiave pubblica del firmatario dallo stato del dominio e verifica la firma sul comando.
2. Verifica che il set di firmatari soddisfi una regola per il comando.

Sessioni autenticate

Le tue operazioni chiave vengono eseguite tra gli host rivolti verso l'esterno e il. AWS KMS HSMs. Questi comandi riguardano la creazione e l'uso di chiavi di crittografia e la generazione di numeri casuali sicuri. I comandi vengono eseguiti su un canale autenticato dalla sessione tra gli host del servizio e il. HSMs. Oltre alla necessità di autenticità, queste sessioni richiedono la riservatezza. I comandi in esecuzione su queste sessioni includono la restituzione di chiavi di dati in chiaro e messaggi decrittografati destinati all'utente. Per garantire che queste sessioni non possano essere sovvertite tramite man-in-the-middle attacchi, le sessioni vengono autenticate.

Questo protocollo esegue un accordo chiave ECDHE reciprocamente autenticato tra HSM e l'host del servizio. Lo scambio viene avviato dall'host del servizio e completato dall'HSM. L'HSM restituisce anche una chiave di sessione (SK) crittografata dalla chiave negoziata e un token chiave esportato che contiene la chiave di sessione. Il token chiave esportato contiene un periodo di validità, dopo il quale l'host del servizio deve rinegoziare una chiave di sessione.

Un service host è membro del dominio e dispone di una coppia di chiavi per la firma dell'identità ($DhOS_i$, $QHOS_i$) e di una copia autentica delle HSMs chiavi pubbliche di identità. Utilizza il set di chiavi di firma dell'identità per negoziare in modo sicuro una chiave di sessione che può essere utilizzata tra l'host del servizio e qualsiasi HSM nel dominio. Ai token chiave esportati è associato un periodo di validità, dopodiché è necessario negoziare una nuova chiave.



Il processo inizia con il riconoscimento host del servizio che richiede una chiave di sessione per inviare e ricevere flussi di comunicazione sensibili tra sé stesso e un membro HSM del dominio.

1. Un host di servizio genera una coppia di chiavi ECDH effimere d_1 , Q_1) e la firma con la sua chiave di identità $Sig_1 = \text{Sign}(dOS, Q_1)$.
2. HSM verifica la firma sulla chiave pubblica ricevuta utilizzando il token dominio corrente e crea una coppia di chiavi ECDH effimere d_2 , Q_2). Quindi completa la ECDH-key-exchange procedura prevista dalla [Raccomandation for Pair-Wise Key Establishment Schemes utilizzando la crittografia a logaritmi discreti per formare una chiave](#) AES-GCM negoziata a 256 bit. L'HSM genera una nuova chiave di sessione AES-GCM a 256 bit. Crittografa la chiave di sessione con la chiave

negoziata per formare la chiave di sessione crittografata (ESK). Crittografa anche la chiave di sessione con la chiave di dominio come token chiave esportato EKT. Infine, firma un valore di ritorno con la sua coppia di chiavi di identità $\text{Sig}_2 = \text{Sign}(\text{dHsk}, (Q_2, \text{ESK}, \text{EKT}))$.

3. L'host del servizio verifica la firma sulle chiavi ricevute utilizzando il token di dominio corrente. [L'host del servizio completa quindi lo scambio di chiavi ECDH secondo la raccomandazione per gli schemi di definizione delle chiavi a coppie che utilizzano la crittografia a logaritmi discreti.](#) Successivamente decrittografa l'ESK per ottenere la chiave di sessione SK.

Durante il periodo di validità nell'EKT, l'host del servizio può utilizzare la chiave di sessione negoziata SK per inviare comandi crittografati con envelope all'HSM. Ogni comando service-host-initiated di questa sessione autenticata include l'EKT. L'HSM risponde utilizzando la stessa chiave di sessione negoziata SK.

Processo di replica per chiavi multi-regione

AWS KMS utilizza un meccanismo di replica tra regioni per copiare il materiale chiave in una chiave KMS da un HSM in una Regione AWS a un HSM in un altro. Regione AWS Perché questo meccanismo funzioni, la chiave KMS replicata deve essere una chiave multi-regione. Quando si replica una chiave KMS da una regione all'altra, le regioni non possono comunicare direttamente, perché si trovano HSMs in reti isolate. I messaggi scambiati durante la replica tra regioni vengono invece recapitati da un servizio proxy.

Durante la replica tra regioni, ogni messaggio generato da un AWS KMS HSM viene firmato crittograficamente utilizzando una chiave di firma di replica. Le chiavi di firma della replica (RSKs) sono chiavi ECDSA sulla curva NIST P-384. Ogni regione possiede almeno un RSK e il componente pubblico di ogni RSK è condiviso con tutte le altre regioni nella stessa partizione. AWS

Il processo di replica tra regioni per copiare il materiale chiave dalla regione A alla regione B funziona come segue:

1. L'HSM nella regione B genera una chiave ECDH effimera sulla curva NIST P-384, la chiave B dell'accordo di replica (RAKB). La componente pubblica della chiave RAKB viene inviata a un HSM nella regione A dal servizio proxy.
2. L'HSM nella regione A riceve la componente pubblica di RAKB e genera quindi un'altra chiave ECDH effimera sulla curva NIST P-384, la chiave A dell'accordo di replica (RAKA). L'HSM gestisce lo schema di istituzione della chiave ECDH su RAKA e la componente pubblica di RAKB e deriva una chiave simmetrica dall'output, la chiave di replica di wrapping (RWK). La chiave RWK viene

utilizzata per crittografare il materiale delle chiavi della chiave KMS multi-regione che viene replicata.

3. La componente pubblica di RAKA e il materiale chiave crittografato con la RWK vengono inviati all'HSM nella regione B tramite il servizio proxy.
4. L'HSM nella regione B riceve la componente pubblica di RAKA e il materiale chiave crittografato tramite la RWK. L'HSM deriva da RWK eseguendo lo schema di istituzione della chiave ECDH su RAKB e la componente pubblica di RAKA.
5. L'HSM nella regione B utilizza la RWK per decrittare il materiale chiave dalla regione A.

Protezione della durabilità

L'ulteriore durabilità del servizio per le chiavi generate dal servizio è garantita dall'uso dello storage offline HSMs e non volatile multiplo dei token di dominio esportati e dall'archiviazione ridondante delle chiavi KMS crittografate. Gli offline sono membri dei domini esistenti HSMs . Ad eccezione del fatto che non sono online e partecipano alle normali operazioni di dominio, gli offline HSMs appaiono nello stato del dominio in modo identico ai membri HSM esistenti.

Il design di durabilità ha lo scopo di proteggere tutte le chiavi KMS di una regione in caso AWS di perdita su larga scala delle chiavi online HSMs o del set di chiavi KMS archiviate nel nostro sistema di storage principale. AWS KMS keys con materiale per chiavi importato non sono incluse nelle protezioni di durabilità offerte dalle altre chiavi KMS. In caso di errore di immissione a livello regionale AWS KMS, potrebbe essere necessario reimportare il materiale chiave importato in una chiave KMS.

Le informazioni offline e HSMs le credenziali per accedervi sono archiviate in casseforti all'interno di camere sicure monitorate in più località geografiche indipendenti. Ogni cassaforte richiede almeno un addetto alla AWS sicurezza e un AWS KMS operatore, provenienti da due team indipendenti AWS, per ottenere questi materiali. L'uso di questi materiali è regolato da una politica interna che richiede la presenza di un quorum di AWS KMS operatori.

Cronologia dei documenti

Questo argomento descrive gli aggiornamenti importanti alla Guida per gli sviluppatori di AWS Key Management Service .

Argomenti

- [Aggiornamenti recenti](#)
- [Aggiornamenti precedenti](#)

Aggiornamenti recenti

La tabella seguente descrive le modifiche importanti apportate a questa documentazione a partire da gennaio 2018. Oltre alle modifiche maggiori elencate qui, aggiorniamo la documentazione di frequente per migliorare le descrizioni e gli esempi e per dar spazio al feedback inviatoci. Per ricevere una notifica sulle modifiche rilevanti, iscriversi al feed RSS.

Potrebbe essere necessario scorrere orizzontalmente o verticalmente per visualizzare tutti i dati di questa tabella.

Modifica	Descrizione	Data
Aggiornamento funzionalità	AWS KMS supporta la rotazione su richiesta di chiavi multiregionali con crittografia simmetrica con materiale chiave importato.	26 novembre 2025
Aggiornamento funzionalità	AWS KMS supporta un nuovo <code>keyExchange</code> campo nella sezione degli eventi. <code>tlsDetails</code> CloudTrail	24 novembre 2025
Aggiornamento funzionalità	AWS KMS supporta il nuovo <code>KeySpec ECC_NIST_EDWARDS25519</code> .	7 novembre 2025

Supporto dual-stack	AWS KMS supporta il dual-stack.	18 giugno 2025
Aggiornamento funzionalità	Aggiunge il supporto per le firme crittografiche post-quantistiche del Module-Lattice Digital Signature Algorithm (ML-DSA).	13 giugno 2025
Rotazione dei tasti importata	È possibile eseguire la rotazione su richiesta delle chiavi KMS con crittografia simmetrica con materiale chiave importato (origine). EXTERNAL	5 giugno 2025
Aggiornamento funzionalità	È stato aggiunto il supporto per le chiavi KMS multiregionali nelle regioni della Cina.	21 novembre 2024
AWS aggiornamento gestito delle politiche	È stato aggiornato il ruolo AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy collegato al servizio aggiungendo uno statement ID (Sid) alla policy gestita con la versione v2 della policy.	21 novembre 2024
Modifica della quota	È stata aumentata la frequenza di richieste predefinita per le ImportKeyMaterial richieste e. DeleteImportedKeyMaterial	23 luglio 2024

Modifica della quota	È stata aumentata la frequenza di richieste di operazioni crittografiche predefinite per le chiavi KMS di crittografia simmetrica, le chiavi RSA KMS e le chiavi ECC e KMS. SM2	8 luglio 2024
Nuova caratteristica	È stato aggiunto un nuovo KeyUsage tipo KEY_AGREEMENT per le chiavi KMS a curva ellittica (ECC) e (solo per le regioni della SM2 Cina) consigliate dal NIST e aggiunto il supporto per derivare segreti condivisi.	13 giugno 2024
Aggiornamenti alla rotazione dei tasti	È stato aggiunto il supporto per periodi di rotazione personalizzati per le rotazioni automatiche dei tasti, le rotazioni dei tasti su richiesta e la visibilità delle principali rotazioni dei materiali.	12 aprile 2024
Aggiornamenti alla policy gestita	Sono state aggiunte nuove autorizzazioni AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy che consentono di AWS KMS monitorare le modifiche nel VPC che contiene AWS CloudHSM il cluster in modo da AWS KMS fornire messaggi di errore chiari in caso di guasti.	10 novembre 2023

Aggiornamento funzionalità	È stato aggiunto il supporto per il parametro dell'API DryRun.	5 luglio 2023
Aggiornamento funzionalità	È stato aggiunto il supporto per l'importazione di materiale chiave per tutti i tipi di AWS KMS chiavi, ad eccezione degli archivi di chiavi personalizzati.	5 giugno 2023
Aggiornamento funzionalità	Aggiornamenti a AWS KMS APIs for Nitro Enclaves	10 marzo 2023
Aggiornamento funzionalità	L'algoritmo di RSAES_PKCS1_V1_5 wrapping è obsoleto. AWS KMS interromperà tutto il supporto RSAES_PKCS1_V1_5 entro il 1° ottobre 2023 in conformità alle linee guida sulla gestione delle chiavi crittografiche del National Institute of Standards and Technology (NIST). Ti consigliamo di iniziare immediatamente a utilizzare un algoritmo di wrapping diverso.	28 febbraio 2023
Aggiornamento funzionalità	È stato aggiunto il supporto per gli archivi di chiavi esterne, una funzionalità che consente di proteggere le AWS risorse utilizzando chiavi crittografiche esterne a. AWS	29 novembre 2022

Modifica della quota	La quota di AWS KMS keys risorse è stata aumentata a 100.000 chiavi KMS in ogni account e regione.	8 luglio 2022
Aggiornamento funzionalità	È stato aggiunto il supporto per le chiavi HMAC KMS in altre versioni Regioni AWS	8 luglio 2022
Nuovo argomento	È stato aggiunto l' AWS Key Management Service argomento Resilienza al capitolo Sicurezza della Guida per gli AWS KMS sviluppatori.	14 giugno 2022
Nuova caratteristica	È stato aggiunto il supporto per AWS KMS le chiavi e le operazioni API che generano e verificano i codici HMAC.	19 aprile 2022
Modifica della documentazione	Sostituzione del termine chiave master cliente (CMK) con AWS KMS key e chiave KMS.	30 agosto 2021
Nuova caratteristica	Aggiunto il supporto per le chiavi multi-regione , un insieme di chiavi KMS interoperabili in diverse Regioni con lo stesso ID chiave e materiale chiave. È possibile utilizzare le chiavi multi-regione per crittografare i dati in una Regione e decrittografare i dati in una Regione diversa.	08 giugno 2021

Nuova caratteristica	È stato aggiunto il supporto per il controllo degli accessi basato sugli attributi (ABAC). Puoi utilizzare tag e alias per controllare l'accesso ai tuoi AWS KMS keys	17 dicembre 2020
Nuova caratteristica	Aggiunto il supporto per le policy di endpoint VPC.	9 luglio 2020
Nuovo contenuto	Spiega le proprietà di sicurezza di AWS KMS	18 giugno 2020
Nuova caratteristica	È stato aggiunto il supporto per chiavi dati asimmetriche AWS KMS keys e asimmetriche.	25 novembre 2019
Funzionalità aggiornata	È possibile visualizzare la politica chiave di nella console. Chiavi gestite da AWS AWS KMS Questa funzione era limitata alle chiavi gestite dal cliente.	15 novembre 2019
Nuova caratteristica	Illustra utilizzare gli algoritmi di scambio di chiavi post-quantistiche ibride in TLS per le chiamate a AWS KMS.	4 novembre 2019
Modifica della quota	Sono state aumentate le quote di risorse per alcuni APIs che gestiscono le chiavi KMS.	18 settembre 2019
Modifica della quota	Modificate le quote di risorse per le chiavi KMS, gli alias e le concessioni per ogni chiave KMS.	27 marzo 2019

Modifica della quota	Modificata la quota di richieste al secondo condivisa per le operazioni di crittografia che utilizzano le AWS KMS keys in un archivio delle chiavi personalizzate.	7 marzo 2019
Nuova caratteristica	Spiega come creare e gestire archivi di chiavi AWS KMS personalizzati . Ogni archivio di chiavi è supportato da un AWS CloudHSM cluster di proprietà e controllo dell'utente.	26 novembre 2018
Nuova console	Spiega come utilizzare la nuova AWS KMS console, che è indipendente dalla console IAM. La console originaria e le relative istruzioni per l'uso come utilizzarla saranno disponibili per un breve periodo di tempo per consentirti di acquisire familiarità con la nuova console.	7 novembre 2018
Modifica della quota	È stata modificata la quota di richiesta condivisa per l'uso di AWS KMS keys.	21 agosto 2018
Nuovo contenuto	Spiega come Gestione dei segreti AWS utilizza AWS KMS le chiavi per crittografare il valore segreto in un segreto.	13 luglio 2018

[Nuovo contenuto](#)

Spiega come DynamoDB supporta l' AWS KMS AWS KMS keys opzione di crittografia lato server.

23 maggio 2018

[Nuova caratteristica](#)

Spiega come [utilizzare un endpoint privato nel tuo VPC](#) a cui connetterti direttamente AWS KMS, anziché tramite Internet.

22 gennaio 2018

Aggiornamenti precedenti

La tabella seguente descrive le modifiche importanti alla AWS Key Management Service Developer Guide prima del 2018.

Potrebbe essere necessario scorrere orizzontalmente o verticalmente per visualizzare tutti i dati di questa tabella.

Modifica	Descrizione	Data
Nuovo contenuto	È stata aggiunta la documentazione relativa a Tag in AWS KMS .	15 febbraio 2017
Nuovo contenuto	È stata aggiunta la documentazione relativa a Monitor AWS KMS keys e a Monitora le chiavi KMS con Amazon CloudWatch .	31 agosto 2016
Nuovo contenuto	È stata aggiunta la documentazione relativa a Materiale della chiave importato .	11 agosto 2016
Nuovo contenuto	È stata aggiunta la documentazione Policy IAM , Riferimento	5 luglio 2016

Modifica	Descrizione	Data
	per le autorizzazioni e Chiavi di condizione .	
Update	Sono state aggiornate porzioni della documentazione nel capitolo Accesso e autorizzazioni alle chiavi KMS .	5 luglio 2016
Update	È stata aggiornata la pagina Quote per riflettere le nuove quote predefinite.	31 maggio 2016
Update	Aggiornata la pagina Quote per riflettere le nuove quote predefinite e aggiornata la documentazione token di concessione per migliorare la chiarezza e la precisione.	11 aprile 2016
Nuovo contenuto	È stata aggiunta la documentazione relativa a Autorizzazione per più principali IAM di accedere a una chiave KMS e a Utilizzo della condizione con indirizzo IP .	17 febbraio 2016
Update	Sono state aggiornate le pagine Politiche chiave in AWS KMS e Modificare una policy della chiave per migliorarne la chiarezza e la precisione.	17 febbraio 2016

Modifica	Descrizione	Data
Aggiorna	Sono state aggiornate le pagine degli argomenti sulla gestione delle chiavi KMS per una maggiore chiarezza.	5 gennaio 2016
Nuovo contenuto	È stata aggiunta documenta zione su CloudTrail.	18 novembre 2015
Nuovo contenuto	Sono state aggiunte istruzioni per Modificare una policy della chiave .	18 novembre 2015
Aggiorna	È stata aggiornata la documentazione sull'utilizzo di Amazon Relational Database Service AWS KMS.	18 novembre 2015
Nuovo contenuto	È stata aggiunta documenta zione su Amazon WorkSpace s.	6 novembre 2015
Update	È stata aggiornata la pagina Politiche chiave in AWS KMS per migliorarne la chiarezza.	22 ottobre 2015
Nuovo contenuto	È stata aggiunta la documenta zione su Eliminare un AWS KMS key , inclusa la documentazione di supporto relativa a Creazione di un allarme e Determina l'utilizzo passato di una chiave KMS .	15 ottobre 2015
Nuovo contenuto	È stata aggiunta la documenta zione relativa a Determinare l'accesso a AWS KMS keys .	15 ottobre 2015

Modifica	Descrizione	Data
Nuovo contenuto	È stata aggiunta la documentazione relativa a Stati chiave delle AWS KMS chiavi .	15 ottobre 2015
Nuovo contenuto	È stata aggiunta documentazione su Amazon Simple Email Service.	1° ottobre 2015
Update	Aggiornata la pagina Quote per spiegare le nuove quote di richieste.	31 agosto 2015
Nuovo contenuto	Sono state aggiunte informazioni sui costi di utilizzo AWS KMS. Consulta Prezzi di AWS KMS .	14 agosto 2015
Nuovo contenuto	Sono state aggiunte le quote di richiesta a. AWS KMS Quote	11 giugno 2015
Nuovo contenuto	È stato aggiunto un nuovo codice Java di esempio che mostra l'utilizzo dell'operazione UpdateAlias .	1° giugno 2015
Aggiorna	È stata spostata la tabella delle regioni AWS Key Management Service in Riferimenti generali di AWS.	29 maggio 2015
Nuovo contenuto	È stata aggiunta la documentazione relativa a Come utilizza Amazon EMR AWS KMS .	28 gennaio 2015
Nuovo contenuto	È stata aggiunta documentazione su Amazon WorkMail.	28 gennaio 2015

Modifica	Descrizione	Data
Nuovo contenuto	È stata aggiunta documentazione sull'utilizzo di Amazon Relational Database Service AWS KMS.	6 gennaio 2015
Nuovo contenuto	È stata aggiunta documentazione su Amazon Elastic Transcoder.	24 novembre 2014
Nuova guida	Introduzione della Guida per gli sviluppatori di AWS Key Management Service .	12 novembre 2014

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.